Statistics Surveys
Vol. 17 (2023) 216–310
ISSN: 1935-7516
https://doi.org/10.1214/23-SS145

Some models are useful, but how do we know which ones? Towards a unified Bayesian model taxonomy

Paul-Christian Bürkner

Department of Statistics, TU Dortmund University, Germany;
Cluster of Excellence SimTech, University of Stuttgart, Germany
e-mail: paul.buerkner@gmail.com

Maximilian Scholz

Cluster of Excellence SimTech, University of Stuttgart, Germany
e-mail: maximilian.scholz@simtech.uni-stuttgart.de

Stefan T. Radev

Cluster of Excellence STRUCTURES, University of Heidelberg, Germany;
Cognitive Science Department, Rensselaer Polytechnic Institute, NY, USA
e-mail: stefan.radev93@gmail.com

Abstract: Probabilistic (Bayesian) modeling has experienced a surge of applications in almost all quantitative sciences and industrial areas. This development is driven by a combination of several factors, including better probabilistic estimation algorithms, flexible software, increased computing power, and a growing awareness of the benefits of probabilistic learning. However, a principled Bayesian model building workflow is far from complete and many challenges remain. To aid future research and applications of a principled Bayesian workflow, we ask and provide answers for what we perceive as two fundamental questions of Bayesian modeling, namely (a) "What actually *is* a Bayesian model?" and (b) "What makes a *good* Bayesian model?". As an answer to the first question, we propose the PAD model taxonomy that defines four basic kinds of Bayesian models, each representing some combination of the assumed joint distribution of all observable and unobservable variables (P), a posterior approximator (A), and training data (D). As an answer to the second question, we propose and discuss ten utility dimensions according to which we can evaluate Bayesian models holistically, namely, (1) causal consistency, (2) parameter recoverability, (3) predictive performance, (4) fairness, (5) structural faithfulness, (6) parsimony, (7) interpretability, (8) convergence, (9) estimation speed, and (10) robustness. Finally, we propose two example utility decision trees that describe hierarchies and trade-offs between utilities depending on the inferential goals that drive model building and testing.

MSC2020 subject classifications: Primary 62–02; secondary 62C10.

Keywords and phrases: Probabilistic modeling, statistical learning, Bayesian statistics, machine learning, model comparison.

Received October 2022.

Contents

1. Introduction

Probabilistic (Bayesian) modeling has seen a surge of applications in almost all quantitative sciences and industrial areas [108, 202, 113, 62, 173, 148]. This development is driven by a combination of several factors, including powerful probabilistic estimation algorithms [139, 24, 126, 229, 249], efficient post-processing [296, 129], flexible open-source software [286, 78, 40], and increased information processing capacity. Furthermore, these factors are coupled with a growing awareness of the benefits of probabilistic modeling, such as inclusion of prior knowledge [220, 205], regularization [110, 41, 27, 240], or uncertainty quantification and propagation [142, 202, 113].

Despite these advances, creating and improving Bayesian models in the context of a principled Bayesian workflow [267, 113] remains a complicated endeavor that requires expertise in various domains; these include subject matter knowledge about the system and the data it generates, statistical learning expertise, programming and understanding of software development, as well as knowledge of numerical approximation and simulation methods [173, 113]. Thus, to aid future research on and applications of a principled Bayesian workflow, we ask and provide answers to what we hold to be two fundamental questions:

1. What actually *is* a Bayesian model?
2. What makes a *good* Bayesian model?

Table 1
Table of important symbols and their corresponding description.

Notation (Symbol)	Meaning (Description)
P, A, D	Joint distribution, approximator, training data
θ, y, $\tilde{y}$	Latent parameters, unrealized observables, realized observables
z, ξ	Random state, random noise (nuisance or exogenous variables)
φ, ψ	Quantity of interest, its model-based estimator (function of θ)
$p(\theta)$	Prior distribution of parameters
$p(y \mid \theta)$	Likelihood function (explicit or implicit/simulation-based)
$p(\theta, y)$	Joint distribution of parameters and observables
$p(\theta \mid y)$	Posterior distribution of parameters given observables
$p_{\mathrm{A}}(\theta \mid y)$	Approximate representation of posterior by approximator A
$\mathbb{G}(\cdot)$, $p^*(y)$	True data generator, true data-generating distribution
$\mathbb{E}_p[\cdot]$	Expected value of a quantity with respect to density p
T, H	Summary statistics of posterior, summary statistics of data

In current practice, the term *Bayesian model* is highly overloaded and used to describe a wide range of objects with potentially very different properties. Moreover, modern Bayesian models are more than just a likelihood and a prior – rather, they resemble complex simulation programs coupled with black-box approximators, interacting with various data structures and context variables, embedded within iterative workflows with multiple feedback loops [199, 79, 173, 113, 267]. Thus, we aim to disambiguate and structure the different meanings of a Bayesian model by proposing the PAD model taxonomy (see Section 2). Our taxonomy aims to accommodate modern uses of Bayesian models and provides an answer to Question 1. With a clear definition of Bayesian models in hand, we describe a collection of ten *utility dimensions* that can be used to quantify the goodness of Bayesian models holistically (see Section 3), thus providing an answer for Question 2. We then continue with a discussion of importance hierarchies and common trade-offs between utilities in Section 4 and end with a conclusion in Section 5.

This paper started as an attempt to organize our thoughts and provide a unifying and consistent language of Bayesian model building. To a certain extent, it is inevitably opinionated. Nevertheless, we aim to be comprehensive in the utility dimensions we discuss, such that all the goals we can sensibly ask from a Bayesian model to achieve have their place in this paper. In contrast, due to the large number of different topics we touch on in the process, the amount of details and cited literature per topic are necessarily non-exhaustive. The cited literature is only meant as a starting point for the interested reader to dive in deeper if they wish. In terms of the target audience, we hope that this paper will be helpful to both methodological researchers developing Bayesian models as well as users applying Bayesian models in practice.

2. What is a Bayesian model?

As the term *Bayesian model* (or just *model* for that matter) can sustain multiple meanings depending on context, it can prove incredibly difficult to talk about

TABLE 2
List of important abbreviations and their corresponding definitions.

Abbreviation	Definition	Section
BNN	Bayesian neural network	2.1.1
MCMC	Markov chain Monte Carlo	2.3.2
HMC	Hamiltonian Monte Carlo	2.3.2
VI	variational inference	2.3.2
KL (divergence)	Kullback-Leibler (divergence)	2.3.2
ABC	approximate Bayesian computation	2.3.3
SMC	sequential Monte Carlo	2.3.3
KDE	kernel density estimation	2.3.3
NDE	neural density estimation	2.3.3
NPE	neural posterior estimation	2.3.3
SNPE	sequential neural posterior estimation	2.3.4
SCM	structural causal model	3.1.1
DAG	directed acyclic graph	3.1.1
HDI	highest density interval	3.2.2
SBC	simulation-based calibration	3.2.3
ECDF	empirical cumulative distribution function	3.2.3
ELPD	expected log predictive density	3.3.2
ENP	effective number of parameters	3.6.1
LOO-CV	leave-one-out cross-validation	3.6.1
GLS	global-local shrinkage	3.6.1
ENC	effective number of coefficients	3.6.1
ESS	effective sample size	3.8.1
MCSE	Monte Carlo standard error	3.8.1
MAP (estimate)	maximum a posteriori (estimate)	3.8.2

models with sufficient clarity. As we will see later, different kinds of models may have different kinds of properties which need to be considered and prioritized by an analyst. Without clearly communicating the essential kind of model one has in mind, a discussion about its properties only contributes to the conceptual entropy in quantitative research. In this section, we attempt to resolve this issue by proposing the PAD taxonomy for Bayesian models (see Figure 1 for an overview; see also Table 1 for a quick reference of key concepts and corresponding notation). We will define four basic model classes and explain how they relate to each other. While the PAD taxonomy might be applicable and useful in other contexts, we will specifically expand on it from a Bayesian perspective.

2.1. P models

We define P models by a joint probability distribution $p(y, \theta)$ over all quantities of interest whose potential variation or uncertainty we express in terms of probability theory. We assume that y represents all observable quantities (i.e., data, observations, or measurements) and θ represents all unobservable quantities (i.e., parameters, latent states, or system variables) within a particular modeling context. In most cases, the joint distribution factorizes into a likelihood $p(y \mid \theta)$ and a prior $p(\theta)$ via the chain rule of probability:

$$p(y, \theta) = p(y \mid \theta)\, p(\theta) \tag{1}$$

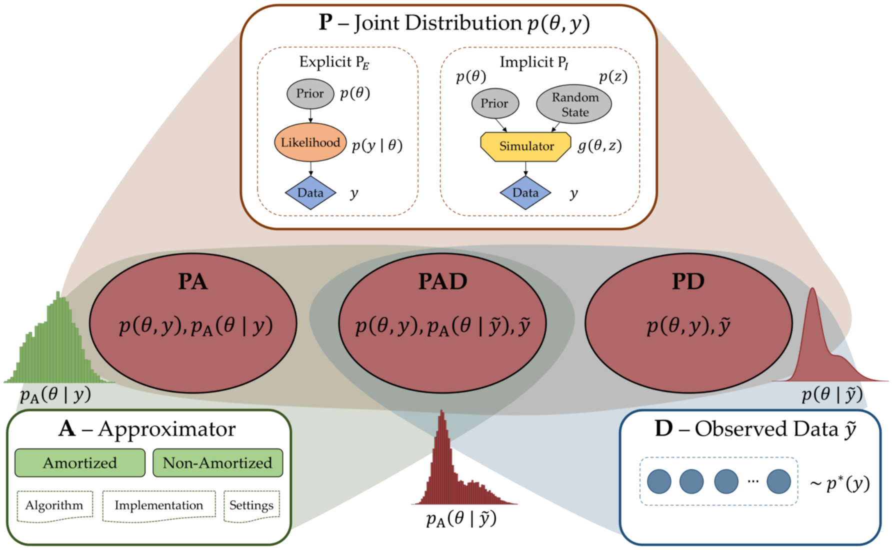

Fig 1. *The PAD Bayesian model taxonomy defines four basic kinds of Bayesian models. Each model kind represents a combination of the joint distribution of all random quantities (P), a posterior approximator (A), and observed data (D).*

This conceptually simple factorization serves as the basis for the common generative (forward) notation used to denote a "probabilistic recipe" for creating synthetic data by sequentially sampling from the prior and the likelihood:

$$\theta \sim p(\theta) \tag{2}$$

$$y \sim p(y \mid \theta) \tag{3}$$

The generative notation overloads the semantics of the "∼" operator, which attains a dual meaning of "distributed as" and "sampled from".

Not all P models are created equal, but most are built to mimic a real-world process or a system, $\mathbb{G}$, whose behavior we can observe or measure. Having some properties that are of interest to the analyst, the opaque generator $\mathbb{G}$ induces an unknown (true) data distribution $p^*(y)$, typically available only through finite observations $\tilde{y} \sim p^*(y)$ (i.e., real-world data). Accordingly, P models strive to encode probabilistic information about the true distribution $p^*(y)$ and/or structural information about the true generator $\mathbb{G}$. The former means that our model matches the statistical properties of p^* either *a priori*, $p(y) \approx p^*(y)$, or *a posteriori* $p(y \mid \tilde{y}) \approx p^*(y)$, where $p(y)$ and $p(y \mid \tilde{y})$ are the prior and posterior predictive distributions of P, respectively. The latter means that our parameters θ correspond to some relevant (hidden) properties φ of $\mathbb{G}$, for which we endeavor to learn something by analyzing $\tilde{y}$. We will expand on these goals in more detail in Section 4.

P models are typically *generative*, that is, we can obtain pseudo-random parameter and data draws via Monte Carlo simulations from Equation (1). The generative property presupposes that the prior is proper (i.e., its density function has a finite integral) and that efficient algorithms for sampling random draws from both $p(\theta)$ and $p(y \mid \theta)$ exist.

P models are the basic building blocks of all further model classes described in the upcoming sections. Moreover, due to their generative properties, standalone P models can be useful on their own for various *forward inference* tasks. These include, for instance, exploring the stability of complex mechanistic equations [170], testing different prior assumptions before a model sees any real-world data [23], or venturing into computational philosophy using simulation [199]. As part of a Bayesian workflow, the plausibility of P models can already be evaluated through prior predictive or prior pushfoward checks [266], which ultimately aim to determine whether the generative behavior of a P model is consistent with the available domain expertise.

2.1.1. Non-parametric P models

In contrast to the above introduced parametric formulation, non-parametric P models replace the finite joint model $p(y, \theta)$ with an infinite dimensional (functional) expression [188]. Practically speaking, the number of parameters in such models simply grows with the number of observed data points [191, 253, 188]. We may still assume that the observed data $\tilde{y}$ is drawn from some unknown distribution $\tilde{y} \sim p^*$, but then place a prior $p(f)$ over the set of all possible generating functions f, instead of over a finite-dimensional parameter space. The forward (generative) model is thus given by:

$$f \sim p(f) \tag{4}$$

$$y \sim p(y \mid f), \tag{5}$$

where the "likelihood" describes the probability of the data given a realization of the function f. For non-parametric regression models (e.g., Gaussian processes, [253]), the function f would also depend on additional inputs (i.e., predictors or covariates) and is thus restricted by the problem design. The corresponding prior typically prescribes some properties of f, for instance, smoothness or certain frequency characteristics [191], but may itself be non-analytic; still, it is often possible to obtain random draws from the generative model and compute marginal and conditional distributions.

In between the parametric and non-parametric worlds, we can encounter high-dimensional P models, such as Bayesian neural networks (BNNs) [190, 148]. The parameters θ of BNNs represent the set of trainable network weights and biases or a subset thereof, such as the weights and biases of the last hidden layer (for a practical overview of recent techniques, see [153]). The prior over network weights is typically chosen out of computational convenience [93], since there is hardly any domain expertise which can yield informative priors. The likelihood of BNNs can also be an ostensibly simple distribution (e.g., a Gaussian) whose parameters are obtained through a highly nonlinear transformation defined by the computational graph of the network. Thus, even though BNNs are formally parametric P models, their high-dimensionality and non-linearity makes them behave more like non-parametric P models [176].

This paper was conceptualized and written mainly with parametric P models in mind. That said, almost all of its aspects apply to non-parametric and high-dimensional P models as well, except, perhaps, for those that presuppose direct interest in the P model parameters (e.g., Section 3.2). Furthermore, despite theoretical differences [188], the practical treatment of parametric and non-parametric P models, when trained on finite data, is not radically different in the end. Finally, non-parametric P models commonly appear as local building blocks in otherwise parametric P models (e.g., a latent Gaussian process as part of an additive model; [168]), blurring the line even further.

2.1.2. Explicit vs. implicit likelihood models

Thus far, we have emphasized that both parametric and non-parametric P models can be analyzed through the lens of their generative properties. A common denominator in such forward inference tasks is that the P model's behavior (i.e., dynamic properties) may not be immediately obvious from the P model's specification alone (i.e., static properties). Thus, *simulation methods* bridge the gap between the specification and the realization of a P model [274, 131]. Indeed, from a simulation perspective, we can further draw a distinction between *explicit likelihood* (P_E) models and *implicit likelihood* (P_I) models.

P_E models are characterized by a likelihood function that has a tractable mathematical form. This means that the likelihood $p(y \mid \theta)$ is known analytically (e.g., Gaussian) and its value can be evaluated directly or approximated numerically for any pair (y, θ). The same logic applies to non-parametric P models using the pair (y, f). P_E models include popular statistical models, such as (generalized) linear and additive models [134], but also (stochastic) differential equation systems with simple statistical properties [152], finite mixture models [55], or feedforward neural networks [124].

P_I models are defined through a Monte Carlo simulation program $y = g(\theta, z)$ and a prior $p(\theta)$, rather than directly through an analytic likelihood function $p(y \mid \theta)$. The simulator g transforms its inputs θ into outputs y through a series of latent program states z. A Monte Carlo simulator only implicitly defines the likelihood density via the relation

$$p(y \mid \theta) = \int p(y, z \mid \theta) \, dz, \tag{6}$$

where $p(y, z \mid \theta)$ is the joint distribution of observables y and random latent program states z, if such a distribution exists. The above integral runs over all possible execution paths of the simulation program for a given input θ and is typically intractable, that is, we cannot explicitly write down the mathematical form of the implied likelihood $p(y \mid \theta)$. P_I models are usually built upon firm theoretical assumptions and computational considerations aimed at providing a faithful representation of the modeled real-world system or process. Common

P_I models include mechanistic neural models [147], particle physics simulators [65], population genetics algorithmic models [137], or agent-based models [125], to name just a few.

The distinction between P_E and P_I models is not a conceptual necessity, but rather an emerging practical convenience. While most standard statistical models can easily be specified in terms of known density or distribution functions, the behavior of complex computational models might be easier to emulate directly using a simulation program. Importantly, P_E and P_I models necessitate the use of different estimation methods and thus disparate modes of approximation and inference, as we will see in later sections.

2.2. PD models

PD models are defined as the combination of a P model and observed data $\tilde{y}$, that is, they represent a tuple $(\, p(y, \theta),\, \tilde{y}\,)$. The data can comprise any number of measurements $\tilde{y}$ with an arbitrary structure (e.g., sets, time series, graphs, etc.). Furthermore, the number of observed data sets (conditioning quantities for the posterior) will be determined by the structure of the P model: Data on a hundred countries represents a single data set from the lens of a multilevel (hierarchical) model, but it comprises a hundred data sets for a single-level (non-hierarchical) P model.

The goal of PD models is to integrate the joint distribution and the observed data to arrive at the corresponding *analytic posterior*:

$$p(\theta \mid \tilde{y}) = \frac{p(\tilde{y} \mid \theta)\, p(\theta)}{p(\tilde{y})} \propto p(\tilde{y} \mid \theta)\, p(\theta), \tag{7}$$

where the denominator $p(\tilde{y}) = \int p(\tilde{y} \mid \theta)\, p(\theta)\, d\theta$ represents the model-implied marginal likelihood (aka evidence) evaluated at $\tilde{y}$ and typically treated as a normalizing constant due to its independence of the model parameters.

If the P model is generative, the analytic posterior exists for every $\tilde{y}$ that satisfies the expected data structure of the P model, regardless of whether or not it represents the true real-world generator $\mathbb{G}$. In the non-representative case, the P model is said to be *misspecified.* In most quantitative sciences, except perhaps in some areas of the natural sciences, we can expect all P models to be misspecified to some (non-negligible) degree. This does not prevent the corresponding PD models from being useful, though, if they can at least express some relevant aspects of reality captured by $\tilde{y}$.

PD models represent the ideal endpoint of Bayesian inference. However, because we can rarely compute the marginal likelihood $p(\tilde{y})$ analytically, we do not have access to the actual PD model outside of textbook examples with limited generality and applicability (i.e., for conjugate P models, [114]). In other words, for most practically relevant and non-trivial P_E and P_I models, we cannot retrieve the analytic posterior $p(\theta \mid \tilde{y})$ and can only work with an approximate representation through the lens of an intermediary A which we call a *posterior approximator.*

2.3. PA models

PA models are defined as the combination of a P model with a posterior approximator A, that is, they constitute a tuple $(\, p(y, \theta),\, p_{\mathrm{A}}(\theta \mid y)\,)$, where the latter denotes any algorithm capable of *somehow* approximating the analytic posteriors of model-implied observations y for a given P model. Approximators themselves exist at both an algorithmic and an implementation level, and details on both levels can influence their behavior and performance. In the absence of actually observed data $\tilde{y}$, PA models can be useful for confirming the computational faithfulness of a workflow, for instance, via simulation-based-calibration (SBC, [284, 208]) or assessing the adequacy of a model for answering a particular research goal [266]. Importantly, the type of P model (i.e., P_E or P_I) will typically determine or necessitate the choice of a particular approximator A, as we will see shortly.

2.3.1. What is an approximator?

More precisely, we can define an approximator as a triple $\mathrm{A} = \{\mathcal{A}, \mathcal{I}, \mathcal{H}\}$, where $\mathcal{A}$ denotes the algorithmic representation (formal computer program), $\mathcal{I}$ denotes the actual implementation in a concrete programming language, and $\mathcal{H}$ denotes the set of admissible hyperparameters (i.e., adjustable settings or inputs) of the approximator. The first two components of A are often entangled when talking about approximators in general, but they require different levels of analysis. For instance, we can determine the computational complexity of $\mathcal{A}$ via standard algorithmic analysis and classify approximators according to their asymptotic run time or memory requirements [60]. However, the latter two will also be constrained by the particular implementation $\mathcal{I}$: Parallel computing can easily turn a scary-looking quadratic $\mathcal{O}(n^2)$ time complexity into a negligible constant run time in practice [60]. Thus, we deem it important to keep the distinction between $\mathcal{A}$ and $\mathcal{I}$ explicit.[1] In addition, the performance of an approximator will heavily depend on the choice of particular hyperparameters $h \in \mathcal{H}$ and these should be explicitly specified in any PA model.

2.3.2. Approximators for P_E models

Currently, the two most commonly used approximators for P_E models are Markov chain Monte Carlo (MCMC) samplers and variational inference (VI) methods, but there exist many more approximator classes, for example, integrated nested Laplace approximation (INLA, [261, 180]) or optimal transport applied to Bayesian inference [81, 160, 233].

[1] Naturally, hardware specifications will further influence the actual run time and space requirements of any approximator, so these specifications should be taken into account when comparing different approximators. The utility of an approximator will also be constrained by the available hardware budget: parallelism is of little use without access to a computing cluster.

MCMC sampling algorithms, such as the Metropolis-Hastings algorithm [135], Gibbs sampling [105], Hamiltonian Monte Carlo [HMC, 214], or its extension to the No-U-Turn (NUTS) sampler [139], belong to a family of stateful algorithms which generate a sequence of correlated draws that converge in distribution to a stationary target distribution [108]. Generally, our goal in MCMC is to construct a (geometrically) ergodic Markov chain on θ whose stationary distribution is the posterior $p(\theta \mid y)$ [108]. In practice, we then sample from the chain to obtain a finite number of random draws from the (hopefully accurate) stationary distribution $p_{\mathrm{A}}(\theta \mid y)$ and use these draws to approximate $p(\theta \mid y)$. More precisely, using the posterior draws, we can efficiently approximate expectations (e.g., mean or variance) and quantiles of the posterior marginals, but not the posterior density itself.

The idea of approximating a complicated distribution via dependent random draws, albeit rather straightforward in hindsight, has gradually transformed and shaped the field of Bayesian inference. Moreover, it constitutes the main logic behind major probabilistic programming languages such as Stan [50] or JAGS [242]. A *sampler* is thus a computer program which uses computer-generated randomness to generate draws from a (complicated) distribution, instead of deriving or estimating its algebraic form.

Differently, variational inference (VI) methods cast the problem of posterior inference as an optimization task. In contrast to MCMC, the resulting posterior approximation $p_{\mathrm{A}}(\theta \mid y)$ is in the form of a tractable density instead of random samples from the posterior. Our goal in VI is to specify a family of approximate densities $\mathcal{Q}$ over the parameters θ of P. Then, we try to retrieve the density $q^* \in \mathcal{Q}$ which minimizes the Kullback-Leibler (KL) divergence to the analytic posterior. Finally, we use $q^*(\theta)$ as our approximation $p_{\mathrm{A}}(\theta \mid y)$ to the analytic posterior.

MCMC and VI methods represent the two endpoints of a trade-off between theoretical guarantees and computational efficiency. MCMC methods enjoy the guarantee that under certain regularity conditions [108], the obtained draws represent the true parameter posterior $p(\theta \mid y)$. More precisely, the posterior expectations can be perfectly recovered if the MCMC chain is run infinitely long and, more practically important, expectations can be efficiently approximated already with a finite number of draws. Despite their favorable theoretical properties and major advances in recent years, MCMC algorithms are notoriously slow, which renders estimation of some complex models or applications to really big data practically infeasible [29]. On the other hand, VI methods can be very fast and offer a viable alternative to MCMC in applications to large data sets or real-time inference. However, VI approximators can suffer severe loss of posterior accuracy and, as of today, offer less guarantees for correct inference than MCMC methods ([29], but see [322, 321]). Thus, the choice between an MCMC or a VI approximator for a particular PA model will largely depend on the modeling context. In addition, highly complex P_E models might not be estimable with either MCMC or VI, in which case they might be treated as P_I models in practice and tackled via simulation-based approximators, as we discuss next.

2.3.3. Approximators for P_I models

Standard MCMC and VI solutions are not applicable to statistical inference with P_I models, since the latter lack an analytic likelihood function $p(y \mid \theta)$. Accordingly, approximators for P_I models leverage Monte Carlo (i.e., randomized) simulations for estimating the posterior based on the implicit likelihood defined by the simulator and Equation (6).

Approximate Bayesian computation (ABC) comprises a broad family of asymptotically correct methods for performing inference with P_I models. The core idea of ABC methods is to approximate the posterior by repeatedly drawing parameters from the prior and then running the simulator with the sampled parameters to obtain a synthetic data set. Whenever a synthetic data set is sufficiently similar to an actually observed data set (as defined by a fixed similarity criterion or a distance metric), the corresponding parameters are retained as a draw from the target posterior, otherwise rejected (i.e., rejection sampling).

In practice, ABC methods are notoriously inefficient and hindered by various methodological “curses”, such as the curse of dimensionality [254] or the curse of insufficiency [193]. Several more efficient methods employ various techniques, such as sequential Monte Carlo [SMC, 275, 165]) or ABC-MCMC [194] with kernel density estimation (KDE) [289] to optimize sampling or correct potential deficiencies, but the core idea of using simulations to aid real-world inference remains invariant across methods.

Recently, machine learning and deep learning innovations have permeated the field of simulation-based inference with the goal of scaling up or replacing standard ABC methods altogether [62]. Most of these innovations require simulation-based training of an expressive machine learning algorithm (e.g., random forests or neural networks) which is then used as a standalone approximator [54, 126, 123, 247], in combination with an ABC routine [151] or an MCMC sampler [136, 90, 184, 33].

For instance, neural density estimation (NDE) methods employ specialized neural architectures for analyzing complex high-dimensional distributions [e.g., natural images, 72, 162, 3]. In the context of Bayesian inference, NDE methods can approximate different components of intractable P_I models and currently represent a field of active and promising development [62, 173]. Specifically, neural posterior estimation (NPE) methods [4, 126, 247, 123, 225, 6] involve simulation-based training of a conditional generative neural network [e.g., normalizing flows, 166, 229]. The trained network then acts as a *functional* that can approximate the posterior across the entire prior predictive distribution of a P model without any re-training, enabling *amortized inference* (to be explained shortly). A shared feature between NPE methods is that they avoid MCMC sampling altogether and can perform exact inference under certain optimal conditions.

Ultimately, the utility of any simulation-based method will depend on a combination of various factors, such as generality, domain expertise, theoretical guarantees, efficiency, scalability, and software availability. The amount of available data will once again play a crucial role in the choice of approximator.

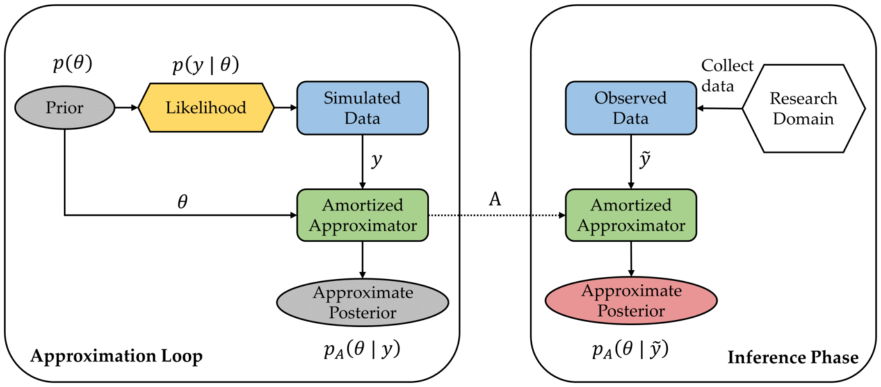

Fig 2. *Amortized approximators incorporate a simulation-based approximation loop (training phase) before any real data are collected. The subsequent inference phase involves no simulations or further optimization and could be carried out almost instantly. The upfront training effort therefore amortizes over arbitrarily many observed data sets from a research domain working on the same P model family.*

In this context, the distinction between *amortized* and *non-amortized* posterior approximators becomes crucial.

2.3.4. Amortized vs. non-amortized approximators

Arguably, there are numerous ways to devise a taxonomy for the ever-growing zoo of posterior approximators. A particularly useful and clear-cut classification views approximators as either *amortized* or *non-amortized*, with different degrees of amortization possible. Amortized approximators involve a costly simulation-based optimization (training) phase which renders subsequent inference on simulated y or real data $\tilde{y}$ extremely efficient (see Figure 2). In other words, the optimization/training effort *amortizes* over repeated inference queries (e.g., over multiple data sets or data set sizes). Differently, non-amortized approximators repeat all necessary computations for each data set or prior choice from scratch and utilize hardly any pooling of computational resources (see Figure 3).

Examples of amortized approximators include the BayesFlow method [247, 245, 245, 248], sequential neural posterior estimation (SNPE) methods operating in a single-round regime [126, 123, 80], machine learning-enhanced ABC [254], or the pre-paid estimation method [204]. Examples of non-amortized approximators include standard explicit inference algorithms, such as MCMC or VI, but also several common ABC methods, such as ABC-SMC [275, 165] or ABC-MCMC [194, 289]. In addition, some neural PA models might include both amortized and non-amortized components, such as multi-round SNPE methods (involving a separate training phase for each data set, [228, 126, 80, 67]), likelihood approximators or surrogates (involving MCMC sampling, [231, 184, 90, 33]), or inference compilation methods (involving SMC, [226, 174]).

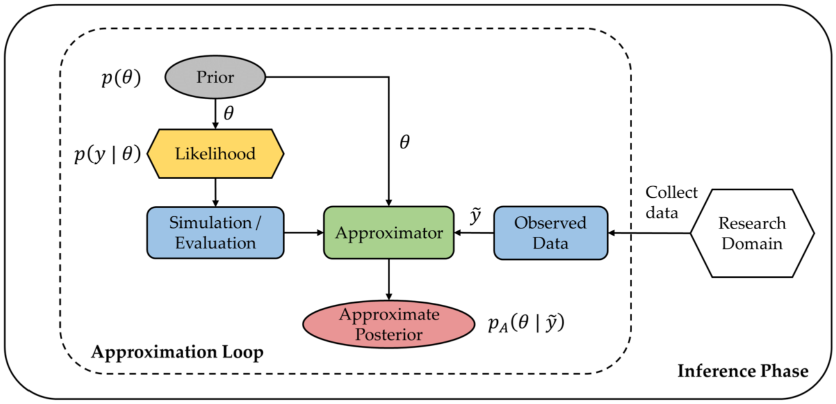

Fig 3. *Non-amortized approximators perform a separate approximation loop (dashed plate) for each observed data set from a given research domain. Likelihood-based approximations, such as MCMC will evaluate the likelihood, whereas simulation-based approximators, such as ABC rejection samplers, will only use random draws from the implicit likelihood (available through stochastic simulations). Approximation and inference are tightly intertwined and the observed data enters the approximation loop.*

Amortized approximators are typically employed to estimate implicit PA(D)[2] models, but are equally applicable to explicit PA(D) models. In the former case, their involvement often arises out of necessity, since P_I models are analytically intractable and state-of-the-art approximators, such as HMC-MCMC, are not applicable out of the box. In the latter case, amortized approximators might be the only resort to estimate multiple PAD models in the presence of multiple data sets, where non-amortized approximators, despite being feasible, would demand an inordinate amount of a researcher's lifetime [303].

2.4. PAD models

PAD models are defined as the combination of a P model, a posterior approximator A, and observed data D, that is, they constitute a triple $(p(y, \theta), p_{\text{A}}(\theta \mid \tilde{y}), \tilde{y})$. Ultimately, PAD models aim to approximate the corresponding PD model through a suitable approximator A, whereas the amount of data D, together with the type of P model, will largely determine the choice of approximator. As a consequence, the properties of a particular PAD model may be very different than what is expected from studying the corresponding PA model, since the observed data $\tilde{y}$ may not have been generated from P itself. This misspecified P model case can arise for both P_E and P_I models and can have different consequences for the validity of inference depending on the particular approximator A [197, 28, 96, 95].

[2] Henceforth, parentheses in the PAD taxonomy denote an "OR relationship". For instance, P(D) would mean "a P or a PD model" and P(A)D would mean "a PD or a PAD model".

For instance, amortized approximators face the challenge of dealing with *simulation gaps* [269, 224]. Simulation gaps occur when P model simulations do not accurately represent the real behavior of the modeled system or when they cannot adequately account for unexpected contamination of the observed data. Simulation gaps are especially critical for amortized approximators since the latter assume that simulations are faithful proxies of reality. Thus, simulations from misspecified P models may lead to subsequent problems for amortized inference on real data [269]. In these cases, the resulting $p_{\mathrm{A}}(\theta \mid \tilde{y})$ will not be representative of the analytic $p(\theta \mid \tilde{y})$ and any substantive conclusions based on the former will have little validity.

In contrast, principle limitations due to model misspecification do not exist for standard, non-amortized Bayesian approximators, such as MCMC. Under certain regularity conditions, MCMC samplers guarantee that the obtained samples represent the analytic posterior $p(\theta \mid \tilde{y})$ even when the underlying P model is misspecified [108]. However, misspecified models might still cause considerable difficulties and convergence problems for MCMC methods in practice. Thus, any trustworthy approximator should be equipped with diagnostics signaling improper convergence or invalid inference queries (see Section 3.8).

2.5. Intermediate summary I

Thus far, with our PAD taxonomy, we have defined four different classes of Bayesian models comprising different, yet interdependent, conceptual elements. Common to all has been the joint probability model (P), which represents the core probabilistic and structural assumptions of a Bayesian model. In addition, we proposed to treat the posterior approximator (A) and the data (D) as further constituents of Bayesian models. We consider this warranted, since all three elements not only determine the scope and validity of the substantial conclusions derived from model-based inference but also influence which assumptions we decide to (and could!) test and which we choose to keep untouched by reality.

3. What makes a good Bayesian model?

Below, we present a total of ten utility dimensions that, from our perspective, capture most relevant aspects of Bayesian models as defined by our taxonomy. For each of these dimensions, we explain (a) its definition and meaning, (b) the reason why we deem it relevant for Bayesian model building, and (c) how to practically measure it. The order in which we present each utility dimension does not indicate their importance but aims to ease their presentation. We discuss the relative importance of utility dimensions in Section 4.

3.1. Causal consistency

A common goal of scientific models is the investigation of a causal hypothesis, such as the improvement a certain treatment might bring to some medical condition or the effect an intervention has on an outcome of interest. Most people are aware of the widely recited folk wisdom that *correlation does not imply causation*. Yet, this adage bears the seeds of a far-reaching and nowadays generally acknowledged opinion that statistics alone simply cannot solve questions of causality [235].

While statistical inference can handle the static nature of associations in observational data, causality is a matter of changing conditions and handling these changing conditions requires causal assumptions to build upon [236]. Moreover, different P models may claim different degrees of causal sophistication. For example, some P_E models built only to make accurate predictions may pass without a single mention of causality, while some mechanistic P_I models may directly embody causal functional relationships, such that an input variable x is assumed to cause an observable y by construction or by derivation from scientific theory. Some complex P models may even hold standard unidirectional notions of causality inadequate, as the dynamics of certain natural systems appear to necessitate bidirectional or hierarchical forms of causal interplay [288, 216].

The scientific methods developed around the notion of causality help us determine whether a P model is a valid recipe for answering a particular causal query in principle. Put differently, we ask whether the probabilistic structure of a P model is consistent with a set of external causal assumptions. Thus, we refer to this implied model utility as *Causal Consistency*.

In this section, we will briefly present the foundation of causal theory based on the work of Pearl [236], as it is currently the most common causal framework. There are adoptions and adaptations for individual fields, such as the social sciences [210, 97] and public health research [294]. Moreover, recent Bayesian statistics textbooks have started discussing causality as a central aspect of statistical analysis [202]. In addition, the fields of causal discovery [143, 278, 120] and optimal experimental design [83, 88, 146] deserve a mention as well, since they tackle problems related to causality. Finally, other promising causal frameworks have been proposed [145] but are not discussed in detail here for reasons of brevity.

3.1.1. Structural causal models

Pearl [236] proposes a framework to express causal assumptions and construct requirements on probabilistic models that make them consistent with those assumptions. The mathematical objects that allow for causal analysis are called *structural causal models* [SCMs, 235] and they comprise structural equations (what we express via P models), causal graphs, as well as interventional and counterfactual logic [238]. For instance, linear regression P models, if combined with proper causal calculus, comprise a widely used and simple form of linear SCMs. However, vastly more complex SCM architectures are possible, such as causal generative neural networks [167], where a causal graph is connected to a generative adversarial network responsible for learning interventional distributions (see Section 3.1.2 for details on interventions).

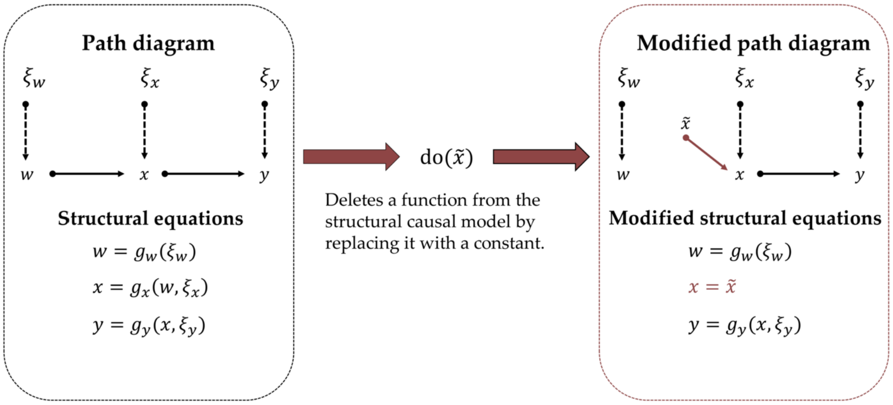

Fig 4. *An example structural causal model (SCM) with three variables. The left panel depicts the pre-intervention path diagram, whereas the right panel depicts the post-intervention path diagram (see text for further clarification).*

For the purpose of this paper, it is sufficient to discuss SCMs comprising a set of three endogenous variables whose causal relationships are to be studied. We refer to these variables as w, x, and y. For every endogenous variable, we assume there exists a corresponding exogenous (noise) variable, ξ_w, ξ_x, and ξ_y, respectively. Under the assumption of *causal sufficiency* (i.e., every exogenous variable affects no more than a single endogenous variable), a hypothesis of the form "*x causes y*" means that y is generated by a structural equation $y = g_y(x, \xi_y)$ for some function g_y. The corresponding causal graph is simply $x \to y$.

To extend this example, the left panel of Figure 4 illustrates a path diagram of the structural equations relating the endogenous variables w, x, and y, along with the corresponding causal graph $w \to x \to y$. Importantly, any set of structural equations also encodes assumptions about the *lack of causal influence*. For instance, the absence of w from the right-hand side of g_y conveys the assumption that y will remain invariant to changes in w, as long as variables x and ξ_y remain constant.

In general, a causal graph implied by a set of structural equations will be a directed acyclic graph (DAG). It can be constructed as follows: The variables that appear on the right-hand side of a structural equation become the parents of the variable that appears on the left-hand side of the structural equation. We can understand the structural equations as encoding explicit *structural* assumptions about the opaque (true) data generator $\mathbb{G}$, which in turn implies a (true) joint distribution of the endogenous variables, here $p^*(x, y, z)$. This distribution is realized by first assuming a joint distribution of the noise variables, $p^*(\xi_w, \xi_x, \xi_y)$, and then propagating this uncertainty to w, x, and y through the respective structural equations.

In P model terms, a DAG can be understood as defining a Bayesian network for the implied joint probability distribution of the endogenous variables [235]. The conditional distribution of an arbitrary endogenous variable v is given by

$p(v \mid \mathrm{N}_v)$, where N_v denotes a set of parent variables of v as implied by the DAG. For the current example, this would imply a generative likelihood that factorizes as $p(w, x, y \mid \theta) = p(y \mid x, \theta)\, p(x \mid w, \theta)\, p(w \mid \theta)$, where θ are our P model parameters (left unspecified in the DAG).

In our model taxonomy, a P model may or may not be consistent with the set of causal assumptions embodied in a DAG, which constitutes a binary metric of causal consistency. For example, consider again the simple DAG given by $x \rightarrow y$, with structural equation $y = g_y(x, \xi_y)$. The concrete approximation of g_y is part of the P model assumptions (see below), whereas adherence to the (external) DAG implies satisfying causal consistency. For example, consider the following linear P model

$$\begin{aligned} x &= \xi_x \\ y &= \beta x + \xi_y \end{aligned} \tag{8}$$

with unspecified distributional forms of ξ_x, ξ_y, and β for simplicity. The approximation $\hat{g}_y$ chosen for g_y is $\hat{g}_y(x, \xi_y) = \beta x + \xi_y$ while $\hat{g}_x(\xi_x) = \xi_x$ is just the identity function. The above P model is clearly causally consistent with the DAG $x \rightarrow y$. In contrast, another linear P model in which we had swapped x and y (i.e., assuming $x = \beta y + \xi_x$), would be causally inconsistent with the graph $x \rightarrow y$.

In linear P models, the regression coefficients represent path coefficients of structural equations and thus quantify the linear "causal effects" of certain variables on others. However, even when a linear P model is causally consistent with a given DAG, its linear functional form $y = \beta x + \xi_y$ may still be a poor approximation of the true (potentially highly non-linear) structural equation $y = g_y(x, \xi_y)$. Thus, an equally causally consistent, but more flexible, non-linear P model may be a better choice in the end, depending on other utility dimensions. This illustrates that causal consistency, as defined here by the formal agreement with a causal DAG, is only a necessary, but not a sufficient condition for a P model to provide trustworthy causal inference. Further requirements will be discussed in the context of parameter recoverability (see Section 3.2).

Causal graphs allow for an unambiguous communication of assumptions about causal relations, but on their own, they still represent static entities. In contrast, *interventions* and *counterfactuals* describe actions which enable us to answer *causal queries* based on (a subset of) these assumptions. Below, for the sake of brevity, we will elaborate solely on interventions (see [236] for more details of counterfactuals).

3.1.2. Interventions

An intervention is an operation that changes the underlying structural equations, hence the corresponding causal graph. Intervening on x means setting it to a fixed value $\tilde{x}$, say, administering the treatment $\tilde{x}$ to a patient. We denote an intervention as $\mathrm{do}(x = \tilde{x})$ or simply $\mathrm{do}(\tilde{x})$ for short. The effect of an intervention on the path diagram of our example three-variable SCM is shown in the right panel of Figure 4. An intervention $\mathrm{do}(\tilde{x})$ differs from conditioning on $\tilde{x}$ in the following way: The former removes the connections of node x to its parents, whereas the latter does not change the causal graph from which data is generated [236, 167]. If we set the value of x to some $\tilde{x}$, then it is no longer determined through the structural equation $g_x(w, \xi_x)$, that is, we have intervened in the generative mechanism. Importantly, the interventional distribution of interest, say, $p(y \mid \mathrm{do}(\tilde{x}))$ may differ from the corresponding conditional distribution $p(y \mid \tilde{x})$.

However, when we only have access to observational data because we cannot intervene in the causal graph (e.g., an experiment is too expensive to perform), our resort is to estimate conditional distributions. Thus, an important question arises: "Which causal queries can we answer (i.e., which interventions' effects can we estimate) based on observational data alone?" In the language of do-calculus, this translates to the question of whether we can circumvent the do operator and express the interventional distribution of interest $p(y \mid \mathrm{do}(\tilde{x}))$ via a conditional distribution [237]. For this purpose, we can use three basic rules of do-calculus that specify the conditions under which we can 1) ignore observations, 2) treat interventions as equivalent to observations, and 3) ignore interventions [237].

Against this background, we say that a P model is *causally consistent for a given causal query*, if that query can be answered by applying the rules of do-calculus to the underlying DAG and all necessary conditional distributions are part of the P model. A P model which is causally consistent with a DAG is also causally consistent for all valid causal queries of that DAG. In practice, however, we can rarely attain (or care about) the former but are only concerned with causal consistency for a few queries of interest. To illustrate this point, let us again consider the DAG $w \to x \to y$ from Figure 4. The linear P model (8) is not causally consistent with this DAG, since it does not include the structural equation $x = g_x(w, \xi_x)$ but only $y = g_y(x, \xi_y)$. However, it is causally consistent for the specific query $p(y \mid \mathrm{do}(\tilde{x}))$ because, after applying the second rule of do-calculus, we find that $p(y \mid \mathrm{do}(\tilde{x})) = p(y \mid \tilde{x})$ for this DAG. Correspondingly, the latter conditional distribution is part of the P model in the form of $y = \beta x + \xi_y$. Naturally, the conditions under which we can answer causal queries using conditional distributions become harder to test for causal graphs containing more than just three variables, but the underlying principles remain the same [58].

3.2. Parameter recoverability

A central goal of Bayesian modeling is to perform parameter inference, that is, to draw conclusions directly from the posterior of the latent parameters or other pushforward quantities of interest. But how can we assess whether our inferences are informative and capture all relevant layers of uncertainty? The *Parameter Recoverability* dimension captures the ability of P models (and of PA models; see Section 3.2.3) to gain information from data and perform faithful uncertainty quantification. Moreover, recoverability is a concept where frequentist statistics inevitably play a role, even in the context of purely Bayesian models.

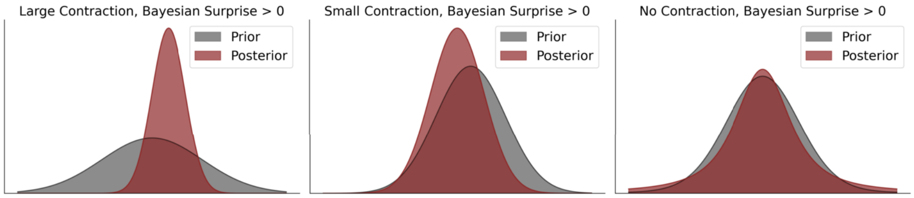

FIG 5. *Three hypothetical (univariate) PD model scenarios illustrating posterior contraction and Bayesian surprise. The leftmost panel depicts a PD model which yields both large posterior contraction and large Bayesian surprise. The middle panel depicts a PD model exhibiting both small posterior contraction and small Bayesian surprise. The rightmost panel depicts a PD model which has zero posterior contraction (i.e., equal prior and posterior variances), yet non-zero Bayesian surprise (i.e., owing to a different tail exponent). Posterior contraction is easier to compute and interpret, but Bayesian surprise is more general, as it captures differences beyond second moments (i.e., variances).*

For the purpose of generality, consider the task of estimating a quantity of interest φ based on a P model and (yet to be realized) data y using an estimator $\psi = \psi(\theta)$ of φ where $\theta \sim p(\theta \mid y)$. The epistemic uncertainty implied by the posterior $p(\theta \mid y)$ is naturally propagated to the posterior of ψ. Based on the implied posterior $p(\psi(\theta) \mid y)$, we can derive both point and uncertainty estimates, among other things, as detailed further below.

To make this notion more concrete, let us consider a simple example. Suppose we are interested in the (true) mean difference of y between two groups, $\varphi = \mathbb{E}[y_1] - \mathbb{E}[y_2]$, where y_1 and y_2 represent the responses of the two groups, respectively. One way to estimate φ here is via a linear regression P model with response vector $y = (y_1, y_2)$ and pointwise likelihood

$$\begin{aligned} y_n &\sim \text{Normal}(\mu_n, \sigma) \\ \mu_n &= \beta_1 \times \mathbb{I}(n \in C_1) + \beta_2 \times \mathbb{I}(n \in C_2), \end{aligned}$$

where $\mathbb{I}$ is the indicator function, C_1 is the index set of observations n belonging to Group 1, and C_2 is the corresponding index set of Group 2. Then, based on this P model, we define an estimator ψ of φ as $\psi = \beta_1 - \beta_2$. Accordingly, ψ does not need to be a model parameter itself but can be any pushforward quantity computable from the parameters. The Gauss-Markov theorem tells us that the chosen estimator ψ has the lowest sampling variance among the class of linear unbiased estimators in case of flat priors on β_1 and β_2. However, the properties of any estimator in general are not always that clear: Consider another example where the true data generator is given by $y_n = f(\varphi\, x_n) + \xi_n$, with x being a known continuous variable, f a monotonically increasing function, and ξ an additive error term. In the absence of knowledge about the exact form of f, we could set up a P model with a normal likelihood that is linear in ψ,

$$y_n \sim \text{Normal}(\psi\, x_n, \sigma).$$

Moreover, the properties of ψ as an estimator of φ will certainly depend on the unknown function f and is likely not as favourable as in the first example.

However, through ψ, we can at least hope to get the sign of φ right, which may as well turn out to be sufficient for meeting the goals of some applications.

3.2.1. Identifiability and Information Gain

Oftentimes, we are interested in learning something about the (true) real-world generator $\mathbb{G}$ through the P model-dependent quantity ψ, justified by its resemblance to the model-independent quantity φ that we assume to play a role in $\mathbb{G}$. As a first step, we need to study whether the data generated by the unknown process enables the P model to extract any information about ψ at all. If the data is not informative, there is no point in further studying the recoverability of φ through ψ. In a frequentist sense, we say that a quantity ψ is *identified* in the given P model, if all the possible values of ψ lead to unique conditional distributions, that is, for any $\psi_1 \neq \psi_2$ we have $p(y \mid \psi_1) \neq p(y \mid \psi_2)$ [52]. Thus, frequentist identification implies that, in the limit of infinite data, no ambiguity remains about possible values of ψ [178].

In a Bayesian context, the posterior captures all information about ψ gained from the data. Thus, the posterior should be a key object for defining identifiability. Since the posterior always exists (as long as the prior is proper), regardless of how informative the data are, the mere existence of the posterior is not a helpful measure of identifiability [see also 181, 122, 265, 264, 25, for discussions of Bayesian identification]. Instead, we have to define identifiability by a juxtaposition of prior and posterior. The transition from prior to posterior (i.e., Bayesian updating) essentially conveys a reduction in uncertainty brought about by observing some data. Equivalently, it can be seen as communicating the *information gain* achieved by accounting for the data. Thus, we expect the posterior to be narrower (sharper) than the prior, as the opposite would imply a loss of information through observation – a rather paradoxical scenario. In other words, the data should be sufficiently informative of ψ, otherwise, the posterior will just resemble the prior.

Bayesian surprise offers a way to quantify arbitrary differences between prior and posterior. The Bayesian surprise is typically defined as the Kullback-Leibler (KL) divergence between the two distributions

$$\mathrm{BS}(\psi \mid y) := \mathbb{KL}\left[p\left(\psi \mid y\right) || \, p\left(\psi\right)\right] \tag{9}$$

$$= \int p\left(\psi(\theta) \mid y\right) \log\left(\frac{p\left(\psi(\theta) \mid y\right)}{p\left(\psi(\theta)\right)}\right) d\theta, \tag{10}$$

but other divergence or integral metrics are also possible [212]. The Bayesian surprise, as defined above, is non-negative and equals zero if and only if $p\left(\psi \mid y\right) = p\left(\psi\right)$. Henceforth, to avoid commitment to the KL divergence, we will use the symbol $\mathbb{D}$ to denote any divergence with the above two properties. In information theory, this particular form of the Bayesian surprise is called a *relative entropy*, and, in Bayesian terms, represents the information gained by updating the prior to the posterior in units determined by the base of the logarithm.[3] Accordingly, in a Bayesian context, ψ is identified if the relative entropy is non-zero.

Further, the concept of *posterior contraction* provides a simpler and tractable empirical diagnostic to assess identification and degrees of informativeness [25]. Posterior contraction formalizes the idea that the posterior should get narrower as the amount of data increases and is computed as the ratio between posterior and prior variance:

$$\mathrm{PC}(\psi \mid y) := 1 - \frac{\mathrm{Var}_{p(\theta \mid y)}(\psi(\theta))}{\mathrm{Var}_{p(\theta)}(\psi(\theta))}. \tag{11}$$

If y contains no information about ψ, then $\mathrm{PC}(\psi \mid y) = 0$. Conversely, the more information (i.e., uncertainty reduction) we gain from y, the larger $\mathrm{PC}(\psi \mid y)$ becomes, up to a maximum of $\mathrm{PC}(\psi \mid y) = 1$. The posterior contraction can be combined with the posterior z-score (i.e., the difference between the true parameter and its posterior mean) as an intuitive two-dimensional estimate of the information gain that can be achieved by a P model when combined with data D [266].

Posterior contraction compares only the second moments (i.e., the variance) of the prior and the posterior, which means that it can be efficiently computed from random draws of those distributions. However, relevant differences between prior and posterior may manifest themselves only in higher moments: It is still possible that we learn something about a distribution, for instance, about its tail exponent or symmetry, while its variance remains largely unchanged (see Figure 5 for an illustration).

So far, we have only considered posterior contraction and Bayesian surprise brought about by a single data set y. Thus, Equation (10) provides only a measure for *local* (i.e., per-data) information gain. Whenever we are interested in *global* (i.e., in expectation over all possible observations) information gain, then the expected Bayesian surprise (EBS) should be considered:

$$\begin{aligned} \mathrm{EBS}(\psi) &:= \mathbb{E}_{p^*(y)}\left[\mathbb{D}\left[p\left(\psi \mid y\right) \,||\, p\left(\psi\right)\right]\right] \qquad (12)\\ &= \int \mathbb{D}\left[p\left(\psi \mid y\right) \,||\, p\left(\psi\right)\right] \, p^*(y)\, dy, \qquad (13) \end{aligned}$$

or, similarly, the expected posterior contraction (EPC). Global information gain assumes access to the distribution p^* of real-world generator outputs and so we can rarely compute this quantity in practice. Instead, we can obtain a Monte Carlo estimate of Equation (13) over multiple observed data sets as an approximation of the true EBS.

In many scenarios (e.g., during model development), we are interested in the recoverability of ψ over the full generative scope of a model P, in combination with a posterior approximator A, *before collecting any data.* In this case, we

[3] Whenever an approximation of the Bayesian surprise is intractable because it requires access to the analytic prior and posterior *densities*, we can define Bayesian surprise through an integral metric, such as the Maximum Mean Discrepancy [MMD, 127], that we can approximate efficiently from prior and posterior draws.

will be considering the approximate posterior $p_{\mathrm{A}}(\psi \mid y)$ and estimating the difference between prior and approximate posterior with respect to the joint distribution $p(\theta, y)$ implied by the P model:

$$\mathrm{EBS}_{\mathrm{P,A}}(\psi) := \mathbb{E}_{p(\theta,y)}\left[\mathbb{D}\left[p_{\mathrm{A}}(\psi \mid y) \,||\, p\,(\psi)\right]\right], \tag{14}$$

In other words, we assume the P model to be a good representation of $p^*(y)$ and evaluate the identification of ψ under this assumption for a given approximator A. Note that approximating the expectation over $p(y, \theta)$ will be computationally expensive for many PA models relying on non-amortized approximators (i.e., ABC or MCMC), since estimating the posterior $p_{\mathrm{A}}(\psi \mid y)$ repeatedly will dominate almost any approach (see also Section 3.2.3). Thus, well-calibrated amortized approximators [126, 247] can serve as remarkable catalysts for efficiently quantifying global information gain for a given PA model before committing to the (costly) process of data collection.

3.2.2. Ground-truth comparisons

We have hitherto assumed that we are dealing with a black-box (true) generator $\mathbb{G}$ whose actions give rise to the data-generating distribution $p^*(y)$. Thus, we did not require φ to play any actual role in the process of data generation. In this section, we will restrict our focus to scenarios where φ does in fact represent some intrinsic properties of $\mathbb{G}$. Thus, we assume an unknown conditional data-generating distribution $p^*(y \mid \varphi)$ and are interested in the similarity between φ and its P-model-based estimator ψ.

Obtaining the posterior of ψ for a single data set and verifying sufficient information gain will tell us nothing about the recoverability of φ given a P model, (i), because the resemblance between φ and its estimator ψ remains unclear and, (ii), because we need to consider the variation in y, that is, variation across possible data sets as well. This means that we ought to estimate the performance of an estimator in expectation over possible data:

$$\mathbb{E}_{p^*(y|\varphi)}[f(\varphi, \psi)] = \int f(\varphi, \psi \mid y)\, p^*(y \mid \varphi)\, dy, \tag{15}$$

where $f(\varphi, \psi \mid y)$ is some function comparing φ with the posterior of ψ, conditional on data y (see below for examples). If the applied P model were the actual data generator itself, then $p^*(y \mid \varphi)$ would be equal to $p(y \mid \varphi) = \int p(y, \theta \mid \varphi)\, d\theta$ and we could set $\psi = \varphi$. In this case, φ could be directly estimated through its own posterior distribution induced by $\varphi(\theta)$ with $\theta \sim p(\theta \mid y)$. However, in reality, we do not know how well P represents the actual generator, and so we continue to distinguish φ from its P model-based estimator ψ.

Notably, the evaluation of Equation (15) does not actually require any observed data and so can be done ahead of time, before commencing any data collection. Unfortunately, as for many things in Bayesian statistics, it is a lot easier to write down the target in mathematical notation than to actually compute it:

The integral in (15) is almost always intractable, even if the posterior of ψ itself were analytic. Thus, in statistical practice, we approximate the integral with a finite sum over M independently simulated data sets $y_1, \ldots, y_M \sim p^*(y \mid \psi)$:

$$\mathbb{E}_{p^*(y|\varphi)}[f(\varphi, \psi)] \overset{\text{MC}}{\approx} \frac{1}{M} \sum_{m=1}^{M} f(\varphi, \psi \mid y_m) \tag{16}$$

This Monte Carlo estimate is now conceptually easy to compute, but potentially very time-consuming, since the P model needs to be fit M times, whereby each single fit may itself demand a considerable amount of time.

In Equations (15) and (16), φ is held constant, which constitutes the typical setup in simulation studies where we fix the ground-truth to a single value per simulation instance. However, the conclusions we can draw from such studies are naturally limited to the few investigated simulation instances (chosen ground-truths). If the investigated instances were non-representative in reality, then we would learn little to nothing of value from our simulations, even if the data-generating distribution $p^*(y \mid \varphi)$ itself were faithful. To consider this implied uncertainty, we can make the criterion (15) fully Bayesian by adding a prior $p^*(\varphi)$ over φ. Thereby, we can now measure recovery in expectation over data y and *a priori* plausible values of the quantity of interest φ:

$$\mathbb{E}_{p^*(y,\varphi)}[f(\varphi, \psi)] = \int \int f(\varphi, \psi \mid y) \, p^*(y \mid \varphi) \, p^*(\varphi) \, dy \, d\varphi, \tag{17}$$

with Monte Carlo (simulation-based) approximation

$$\mathbb{E}_{p^*(y,\varphi)}[f(\varphi, \psi)] \overset{\text{MC}}{\approx} \frac{1}{M} \sum_{m=1}^{M} f(\varphi_m, \psi \mid y_m) \tag{18}$$

for M ground-truth simulations, each generated according to $\varphi_m \sim p^*(\varphi)$ and $y_m \sim p^*(y \mid \varphi_m)$.

Point estimation One central aspect of parameter recoverability that can be assessed in terms of expectations over the data-generating distributions is *point estimation*. We write $T(\psi \mid y)$ for a point estimator derived from the posterior of ψ. Most commonly, we compute the posterior mean $\int \psi(\theta) \, p(\theta \mid y) \, d\theta$, or alternatively the posterior median or mode. Due to aleatoric uncertainty in the data y, we cannot expect $T(\psi \mid y) = \varphi$ for all y, even if the former would be the best possible point estimator of φ. Instead, we can measure how far away our estimator is from the truth via a strict distance function d on $T(\psi \mid y)$ and φ, such that $d(T(\psi \mid y), \psi) = 0$ holds if and only if $T(\psi \mid y) = \varphi$. Common distance functions are the *bias* $T(\psi \mid y) - \varphi$, the *squared error* $(T(\psi \mid y) - \varphi)^2$, and the *absolute error* $|T(\psi \mid y) - \varphi|$. To estimate the performance of a point estimator in expectation over the data-generating process, we would set $f(\varphi, \psi \mid y) = d(T(\psi \mid y), \psi)$ and then apply Equations (15) to (18). Whenever we compare P models based on their point estimation capabilities, we would prefer the P model with the smallest expected distance of its point estimator to the assumed true φ.

Uncertainty estimation An uncertainty estimator is defined as a parameter region that is supposed to contain the true quantity of interest φ with a certain (user-defined) probability q. We write $U_q(\psi \mid y)$ to denote a q uncertainty region derived from the posterior of ψ. Common Bayesian uncertainty regions are quantile-based credible intervals and highest density intervals (HDIs) [108]. We say that an uncertainty region is *well calibrated* for a given φ (in a frequentist sense) if the following equality holds:

$$q = \mathbb{E}_{p^*(y|\varphi)}[\mathbb{I}(\varphi \in U_q(\psi)] = \int \mathbb{I}(\varphi \in U_q(\psi \mid y))\, p^*(y \mid \varphi)\, dy, \tag{19}$$

where $\mathbb{I}(\varphi \in U_q(\psi \mid y))$ is the indicator function evaluating to 1 if $\varphi \in U_q(\psi \mid y)$ and to 0 otherwise. In other words, an uncertainty region for probability q is well calibrated if it contains the assumed true parameter in a fraction of q data sets. If the above property holds for every uncertainty region $U_q(\psi \mid y)$, we say that the whole posterior of ψ is well calibrated for estimation of φ.

Bayesian uncertainty regions are not generally designed to satisfy this frequentist calibration and there is no guarantee that they will [108, 213]. Yet, it can be a perfectly valid approach to use them even to satisfy purely frequentist goals [103]. Interestingly, when considering expectations over $(y, \varphi) \sim p^*(y \mid \varphi)\, p^*(\varphi)$ as in Equation (17), a P model will exhibit perfect calibration as long as its generative behavior matches the unknown data generator and posterior computation is exact [284]. This property is extensively used in diagnosing the correctness of posterior approximations, a topic we will discuss in Section 3.2.3.

When comparing P models based on their uncertainty estimation of φ, we would prefer the model which yields uncertainty estimates closest to the equality in Equation (19) for some pre-selected, application-specific uncertainty regions. For example, if we were primarily interested in well-calibrated 95% credible intervals (perhaps more precisely stated, *compatible intervals*, [202]), then we would prefer the model for which these intervals had closest to $q = .95$ coverage of the assumed true φ. That said, for some specific analysis goals, for example in null-hypothesis significance testing [171], over-coverage (higher than q coverage) may be more acceptable than under-coverage, or vice versa, depending on the assigned utility values of the corresponding Type-I and Type-II errors [267].

Sharpness Multiple P models, say P_1 and P_2, may provide estimators ψ_{P_1} and ψ_{P_2} that are equally well calibrated for a quantity of interest φ, yet their uncertainty regions may differ in coverage [121]. This implies that calibration alone is insufficient to describe the appropriateness of uncertainty regions: Additionally, we need to introduce the concept of *sharpness*. We say that model P_1 is sharper than model P_2 for an uncertainty region $U_q(\psi \mid y)$ with finite bounds, if that region is better or equally well calibrated in P_1 than for P_2 and if the volume of $U_q(\psi_{\mathrm{P}_1} \mid y)$ is smaller than the volume of $U_q(\psi_{\mathrm{P}_2} \mid y)$ in expectation over the data-generating distribution:

$$\mathbb{E}_{p^*(y|\varphi)}\left[\mathrm{Vol}(U_q(\psi_{\mathrm{P}_1} \mid y))\right] < \mathbb{E}_{p^*(y|\varphi)}\left[\mathrm{Vol}(U_q(\psi_{\mathrm{P}_2} \mid y))\right], \tag{20}$$

where Vol indicates the volume in Euclidean space. For unidimensional φ and corresponding uncertainty region, say, a 95% credible interval, the volume is simply equal to the width of the interval. Of course, depending on whether we hold φ constant or assign a generating prior $p^*(\varphi)$ to it, we can also investigate sharpness in expectation over the joint distribution $p^*(y \mid \varphi)\, p^*(\varphi)$, instead of only focusing on $p^*(y \mid \varphi)$. If sharpness holds for all finite-volume uncertainty regions $U_q(\psi \mid y)$, then the posterior of ψ_{P_1} is sharper than the posterior of ψ_{P_2}. Well-calibrated uncertainty regions cannot be infinitely sharp and there exists a sharpest model and corresponding estimator if the set of well-calibrated models is non-empty [121]. However, in practice, we have no access to this sharpest model. Thus, in contrast to calibration, sharpness cannot be practically computed in an absolute sense, but can only be probed as a relative quantity in the context of two or more P models.

3.2.3. Calibration of posterior approximations

So far we have primarily focused on P models in the context of parameter recoverability and all of the estimators assumed access to the analytic posterior $p(\theta \mid y)$ to obtain the analytic posterior $p(\psi(\theta) \mid y)$ of the estimator ψ of φ. Since we do not have access to the analytic posterior in practice, our typical estimators are based on PA models.

Correspondingly, we define the estimator ψ_{A} of φ via the approximate posterior $p_{\mathrm{A}}(\psi(\theta) \mid y)$ of ψ obtained by the approximator A. If A were approximating the posterior via random draws $\theta^{(s)}$ from $p_{\mathrm{A}}(\theta \mid y)$, the approximate posterior $p_{\mathrm{A}}(\psi(\theta) \mid y)$ would be represented by the pushforward draws $\psi(\theta^{(s)})$. Thus, we can evaluate identifiability, point and uncertainty estimation, as well as the sharpness, of a PA model by replacing ψ with ψ_{A} in the corresponding equations. Ideally, we would like to separate the estimation of φ via ψ from the estimation of ψ via ψ_{A} and we can do so if we assume that the considered P model *is* the true generator itself. This is due to two related *self-consistency* properties. The first one is

$$p(\theta) = \int \int p(\theta \mid y)\, p(y \mid \theta^*)\, p(\theta^*)\, dy\, d\theta^*, \tag{21}$$

which states that a P model's prior (left-hand side) is equal to the P model's *data-averaged posterior* (right-hand side), that is, the posterior in expectation over its own generating distribution [284]. The second one states that all uncertainty regions $U_q(\psi \mid y)$ of all pushforward quantities ψ are well calibrated, as long the generating distribution of the assumed P model is equal to true data-generating distribution and posterior computation is exact [284]. Writing ψ^* instead of φ to explicate the direct correspondence between the quantity of interest and its estimator ψ, this property can be written as

$$q = \int \int \mathbb{I}(\psi^* \in U_q(\psi \mid y))\, p(y \mid \psi^*)\, p(\psi^*)\, dy\, d\psi^*. \tag{22}$$

Both self-consistency properties are useful, but Equality (22) provides a particularly convenient means to diagnose the calibration of the approximated posterior

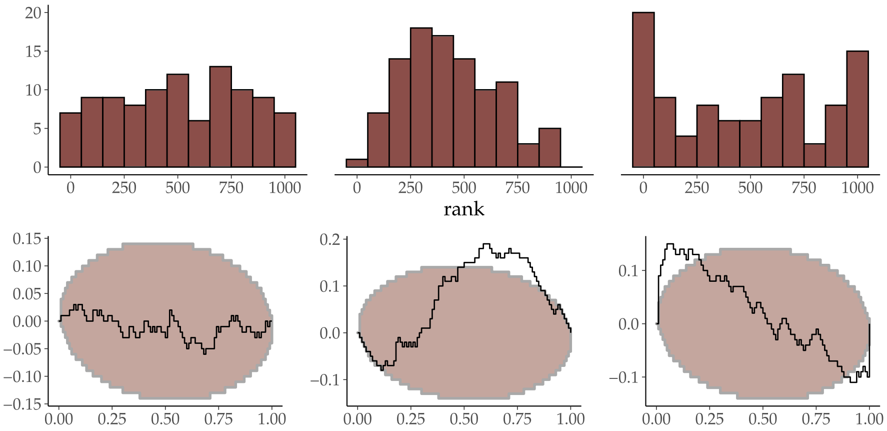

Fig 6. *Simulation-based rank histograms (top) and corresponding empirical cumulative distribution function (ECDF) difference plots [283] (bottom) for three hypothetical quantities of interest. The pink areas in the ECDF difference plots indicate 95%-confidence intervals under the assumptions of uniformity and thus allow for a null-hypothesis significance test of self-consistent calibration. Left: A well-calibrated quantity. Center: A miscalibrated quantity with too many lower ranks indicating a positive bias in the PA model-based posteriors. Right: A miscalibrated quantity with too many extreme ranks indicating overconfident PA model-based posteriors (i.e., variance underestimated).*

$p_A(\psi(\theta) \mid y)$: Under perfect (self-consistent) calibration, the posterior probability $\Pr(\psi^* \leq \psi)$ is uniformly distributed in the unit interval [284, 283]. If the approximate posterior can be expressed in terms of random draws, then uniformity can be tested empirically by comparing the empirical distribution of ranks

$$r(\psi^*, \psi(\theta_{1:S}) \mid y_m) := \sum_{s=1}^{S} \mathbb{I}(\psi^* \leq \psi(\theta^{(s)})) \quad \text{for} \quad \theta^{(s)} \sim p_{\mathrm{A}}(\psi(\theta) \mid y_m) \tag{23}$$

over M simulated data sets to a uniform distribution, a procedure known an *simulation-based calibration* [SBC, 284]. If the distribution of ranks is close enough to uniformity (e.g., according to a frequentist null-hypothesis significance test), we can conclude that the PA model is well calibrated for approximating the P model, assuming self-consistency of P. The required uniformity can be checked graphically, for example via histograms (top row of Figure 6) or by plotting the empirical cumulative distribution function (ECDF) of the ranks normalized against their expected values under uniformity (bottom row of Figure 6), a method known as ECDF difference plots [283].

Even though self-consistency tested via SBC is a powerful tool to ascertain the trustworthiness of a PA model if the underlying P model is well specified, it tells us nothing about the trustworthiness of PA if P is misspecified, that is, if its joint distribution cannot accurately represent the true data generating process $p^*(y)$. In the latter case, we currently have no general procedure to verify the trustworthiness of a posterior approximation, that is, how close a PAD model is to the PD model it attempts to approximate (but see [197, 322] for recent theoretical work). This is a subtly different problem than dealing with misspecified PD models, whose convergence characteristics have been established under certain regularity conditions [163, 164]. In the case of PAD models, we can only hope that self-consistent calibration of PA implies *good enough* calibration in a sufficiently large model neighborhood of P that also contains p^*. For posterior approximators coming with guarantees of asymptotic correctness, such as MCMC, this hope is probably better justified than for neural approximators that have been shown to perform poorly under P model misspecification [269, 307].

3.3. Predictive performance

Undoubtedly, predictive performance is the central utility in most machine learning research [134] and an essential goal of computational [227] and scientific models in general [101]. Moreover, predictive performance has recently been elevated to an indispensable condition for reproducible quantitative research in the social sciences [320]. In deep learning, enormous amounts of computing resources are spent even for just a second decimal improvement in predictive accuracy on domain benchmark data sets [116], notably at the expense of other utilities (e.g., parsimony, see Section 3.6, or estimation speed, see Section 3.9). In our Bayesian model taxonomy, we treat predictive performance as just one of the ten model utilities, but we still recognize it as an important one.

In a way, predictive performance would be nothing but a special case of parameter recoverability (see Section 3.2), if not for the fact that it targets observable variables that are comparable against observed data. This opens up the possibility to directly evaluate predictive performance in real-world scenarios instead of having to use simulations, as is often necessary for estimating parameter recoverability. Along similar lines, predictive P(D) model comparison or averaging can be seen as a form of parameter recoverability from the perspective of mixture modeling (with the individual P models as components) or in terms of continuous model expansion [107]. However, in practice, we approach these challenges mainly based on predictions from separate P(A)D models to reduce conceptual and computational costs [318].

In the following, we denote the set of "test" data to be predicted as y^*, whereas P(A)D model "training" data continues to be denoted by $\tilde{y}$. In principle, these two data sets are allowed to fully coincide, partially overlap, or be completely disjoint (see Section 3.3.3), and $\tilde{y}$ may even be empty (see Section 3.3.2). Further, we will allow the test data to be clustered into C mutually independent and exhaustive clusters $y^* = \{y^*_c\}_{c=1}^{C}$. In most applications, both y^* and $\tilde{y}$ are associated with observed input variables (aka features, predictors, or covariates), but we will keep these implicit to make the notation more readable.

The ocean of predictive performance metrics for Bayesian models is vast and we refer to [299] for a comprehensive overview. To illustrate some overarching points in this article, we will focus on a few important metrics that follow the general form

$$\mathcal{L}(y^*, \tilde{y}) := \sum_{c=1}^{C} l\left(\mathbb{E}_{p(\theta \mid \tilde{y})}[f(y_c^*, \theta)]\right) = \sum_{c=1}^{C} l\left(\int f(y_c^*, \theta)\, p(\theta \mid \tilde{y})\, d\theta\right), \tag{24}$$

where $f(y_c^*, \theta)$ is a predictive score comparing a test data cluster y_c^* with corresponding model-based predictions. We compute the *expected predictive score* by integrating over the PD model posterior $p(\theta \mid \tilde{y})$, where l is some function applied to each expectation before summation over clusters. Whenever we use a PAD model, we need to approximate the above expectation over $p_{\mathrm{A}}(\theta \mid \tilde{y})$, either by using random draws from $p_{\mathrm{A}}(\theta \mid \tilde{y})$ or by relying on an approximate closed-form density. Below, we examine predictive performance along multiple dimensions: absolute versus relative, prior versus posterior, and in-sample versus out-of-sample predictive performance.

3.3.1. Absolute and relative predictive performance

Evaluating absolute predictive performance requires knowing an optimally achievable value of the predictive metric, whereas relative predictive performance only involves comparing multiple P(D) models' predictions evaluated on the same test data y^*. As an example for the former, consider the per-observation squared difference $f(y_i^*, \theta) = (y_i^* - \hat{y}_i(\theta))^2$ as a predictive score, where $\hat{y}_i(\theta)$ is a P(D) model-implied prediction given parameter value θ (e.g., a single random draw or realization from the posterior predictive distribution, see [299]). In this case, we know that the optimal value of Equation (24) is zero. For the sake of increased interpretability, such squared differences can be further transformed to the canonical "percentage of explained variance" R^2 measures which are bounded between 0 and 1, the latter indicating optimal predictions [109].

However, optimal predictions are not achievable in practice, since even a Bayes-optimal decision maker may elicit suboptimal predictions in the presence of aleatoric uncertainty [134], at least when it comes to out-of-sample predictions (see Section 3.3.3). Moreover, since we nearly never know the Bayes-optimal decisions in practice (hence the need for predictive modeling in the first place), the expected optimal achievable predictive performance is also unknown to us. As a result, relative predictive performance is usually our only resort in practical applications [299].

That said, some models produce such strikingly poor predictions that they can be ruled out without the need to find a better model first, often via visual predictive checks [102]. For instance, if we consider the case illustrated in Figure 7, it is immediately obvious that the normal likelihood P model (left-hand side) is inappropriate for the given count data. As another example, consider a P(A)D model for binary classification that achieves just 50% accuracy, equal to random chance. Assuming a balanced data set (i.e., both classes occur with the same frequency), we would not need a competing model to conclude that the classifier is bad – unless our goal was to demonstrate that the two categories cannot be possibly differentiated given the available information.

3.3.2. Prior and posterior predictive performance

The distinction between prior and posterior predictive performance has often led to confusion in the past and still remains a rather precarious one to discuss. Prior and posterior predictive performance are distinguished based on whether we evaluate predictions before or after conditioning on the training data $\tilde{y}$, respectively [183]. In other words, we either compute (or approximate) expectations over the prior, $p(\theta)$, or over the posterior $p(\theta \mid \tilde{y})$. Since prior predictive performance does not require the training data (see Equation (24)), we consider it a utility of P(A) models, while we view posterior predictive performance as a utility of P(A)D models. We still require the test data y^*, but it is not a part of any model class in our taxonomy.

Statistically, the line between prior and posterior predictive performance is thin and more quantitative than qualitative [219]. As an illustration, suppose we observe N data points in total – then we could choose to use none, $\tilde{y} = \emptyset$, or any number between 1 and N for model training. For complex P models, the predictive result implied by using one or two observations for training, rather than none at all, will be almost identical, despite everything but zero training data technically counting as "posterior" predictive performance [219]. Yet, the metrics commonly applied to quantify prior and posterior predictive performance differ not only in the amount of available training data but also in some other non-trivial ways (to be explained below).

In general, any predictive metric should match the intended real-world prediction goals. Below, we will focus on certain (log-)probability metrics, which can be considered good general-purpose choices in the absence of any known task-specific option [299].

Prior predictive performance The canonical metric for evaluating prior predictive performance is the joint P model likelihood evaluated at the test data, $f(y^*, \theta) = p(y^* \mid \theta)$, with $C = 1$ and $l =$ identity, in which case the prior expectation above becomes the marginal likelihood:

$$p(y^*) = \mathbb{E}_{p(\theta)}[p(y^* \mid \theta)] = \int p(y^* \mid \theta)\, p(\theta)\, d\,\theta. \tag{25}$$

When used for model comparison, the marginal likelihood then gives rise to well-known comparative metrics known as Bayes factors evaluated by comparing two P models P_j and P_k as

$$\mathrm{BF}_{jk} := \frac{p(y^* \mid \mathrm{P}_j)}{p(y^* \mid \mathrm{P}_k)} \tag{26}$$

and posterior model probabilities over a set of J models $\{\mathrm{P}_j\}_{j=1}^J$ as

$$p(\mathrm{P}_j \mid y^*) = \frac{p(y^* \mid \mathrm{P}_j)\, p(\mathrm{P}_j)}{\sum_{k=1}^J (y^* \mid \mathrm{P}_k)\, p(\mathrm{P}_k)}, \tag{27}$$

where $p(y^* \mid \mathrm{P}_j)$ denotes the marginal likelihood of P model P_j and $p(\mathrm{P}_j)$ denotes the corresponding prior probability with $\sum_{j=1}^J p(\mathrm{P}_j) = 1$, following a closed-world assumption [21, 318].

Although the marginal likelihood is formally an expectation and thus, in theory, we can approximate it arbitrarily well using sufficiently many random draws from the prior, it is practically impossible to evaluate due to its unfavorable pre-asymptotic behavior for any non-trivial model [203, 299, 129]. The main reason for this is that the parameter subset for which $p(y^* \mid \theta)$ contributes to the integral in Equation (25) (i.e., the typical parameter set implied by the test data; [24]) is very narrow and thus we need a very high number of prior draws to ensure sufficiently many of them occupy that narrow space. In addition, numerical issues caused by $p(y^* \mid \theta)$, such as floating-point underflow, can also be hindering.

For these reasons, the practical computation of marginal likelihoods currently rests on bridge sampling [15] relying on posterior draws from a corresponding PAD model [203, 129]. In contrast to estimating posterior expectations or quantiles, bridge sampling requires about an order of magnitude more posterior draws to yield reliable results [129] and is still largely missing principled convergence diagnostics or uncertainty quantification [128], leaving room for future research.

An alternative prior predictive metric arises if one uses the log-likelihood $f(y^*, \theta) = \log p(y^* \mid \theta)$ as a (predictive) score instead of the likelihood itself, which leads to the Gibbs loss [308] that, for factorizable likelihoods [39], evaluates to

$$\mathrm{Gibbs}_{p(\theta)}(y^*) := \mathbb{E}_{p(\theta)}[\log p(y^* \mid \theta)] = \sum_{i=1}^{N^*} \mathbb{E}_{p(\theta)}[\log p(y_i^* \mid \theta)]. \tag{28}$$

The Gibbs loss is not only simpler to evaluate for exponential family models [299] and numerically more stable than the marginal likelihood but also exhibits better pre-asymptotic behavior for factorizable likelihoods when estimated via prior draws since the integrands become much simpler. However, the Gibbs loss cannot be used to obtain actual predictions because it does not evaluate to a predictive distribution over y^* [299].

The latter problem can be avoided by taking expectations with respect to individual test observations y_i^* first and only taking the log afterwards ($C = N^*$ and $l = \log$), which leads to the expected log predictive density (ELPD) metric [299, 296], evaluated over the prior:

$$\mathrm{ELPD}_{p(\theta)}(y^*) := \sum_{i=1}^{N^*} \log p(y_i^*) = \sum_{i=1}^{N^*} \log \mathbb{E}_{p(\theta)}[p(y_i^* \mid \theta)]. \tag{29}$$

Comparing equations (25) and (29), we see that the marginal likelihood considers the joint predictive density of all test data y^*, while the ELPD considers marginal predictive densities of y_i^*, marginalized over all other test data. Even though the ELPD has found wide application in the context of posterior predictive performance [296], it does not yet seem to play a noteworthy role in the context of prior predictive performance. However, together with the Gibbs loss, it may become a computationally favourable competitor to metrics based on the marginal likelihood.

Posterior predictive performance When assessing posterior predictive performance, we apply the same metrics we encountered in the context of prior predictive performance but evaluate expectations over the posterior induced by the training data $\tilde{y}$. However, the practical popularity of the metrics seems to be reversed when it comes to posterior predictions. For example, the posterior ELPD

$$\mathrm{ELPD}_{p(\theta \mid \tilde{y})}(y^*) := \sum_{i=1}^{N^*} \log p(y_i^* \mid \tilde{y}) = \sum_{i=1}^{N^*} \log \mathbb{E}_{p(\theta \mid \tilde{y})}[p(y_i^* \mid \theta)] \tag{30}$$

finds widespread application [296], while the "conditional marginal likelihood"

$$p(y^* \mid \tilde{y}) = \mathbb{E}_{p(\theta \mid \tilde{y})}[p(y^* \mid \theta)] = \int p(y^* \mid \theta)\, p(\theta \mid \tilde{y})\, d\theta \tag{31}$$

has not yet attained wide popularity, despite having several useful properties [219, 18, 130, 183].

The choice between prior or posterior predictive performance seems to depend on the modeling goals for which a P model is specified. While prior predictive performance seems to be favored for the purpose of testing scientific theories [316, 132, 85, 130], posterior predictive performance is the perspective of choice in almost all machine learning scenarios (but see [320]), where we first obtain a PAD model based on training data (and perhaps only minimal prior information) and then utilize the model in downstream predictive tasks [134].

3.3.3. In-sample and out-of-sample predictive performance

We measure in-sample predictive performance if the test data is a subset of the training data, $y^* \subseteq \tilde{y}$, but measure out-of-sample predictive performance if test and training data do not overlap, that is, $y^* \cap \tilde{y} = \emptyset$. Whenever we evaluate prior predictive performance, we have no training data per definition and thus always measure out-of-sample predictions. Accordingly, the difference between in-sample and out-of-sample predictive performance only matters in the context of posterior predictions.

From a posterior predictive perspective, the decision between using in-sample and out-of-sample predictive performance is based on whether or not we want to generalize our inferences from a data set to a wider population. If a given data set included the entire problem space, then in-sample predictive performance would be sufficient. However, as most introductory statistical courses teach, a data set is typically only a small sample from a much larger population, to which we would like to extend our inferences. Thus, out-of-sample predictive performance (aka generalization ability) is almost always what we are after [134, 299, 296, 320]. That said, we can still learn from in-sample predictive performance, as it provides an upper bound for out-of-sample predictive performance in expectation, such that when in-sample predictions are poor, out-of-sample predictions are likely to be even worse [134, 296, 102].

In the presence of only a single overall data set y_{total}, estimating out-of-sample predictions is practically realized via data splitting, such that $y_{\text{total}} = \{\tilde{y}, y^*\}$. To reduce the dependency of the predictive results on a single realized data split, we typically perform cross-validation by repeating the data splitting several times (folds), evaluating out-of-sample predictions for every fold, and then aggregating the results across folds [280, 299, 296].

The type of cross-validation scheme employed should resemble the envisioned prediction goals for which the PD model has been created [299]. For example, the predictive goal of time series models is usually to predict future values based on past values, making leave-future-out cross-validation a sensible choice [46]. Regardless of the type of cross-validation employed, it involves the repeated fitting of the same P(A) model to different data sets. Depending on the number of such refits, the individual data sizes, and the applied approximator, the required estimation time can quickly become prohibitive for any practical use. As such, approximate cross-validation procedures that require no or only a few refits have proven to be highly popular in practice [296, 300, 46]. However, key cross-validation schemes, such as leave-group-out cross-validation, cannot yet be robustly approximated, so there is more research needed in that direction [223].

Although evaluating out-of-sample predictive performance is often our best shot at preventing overfitting to the training data, it is not always sufficient to fully achieve good generalization within commonly applied model-building workflows [113]. In these workflows, we typically fit different P models to the same data in an iterative fashion. For example, we might first compare two models, decide which one to retain, and only then fit a third model to compare it with the winner of the first round. Even if each model choice was based on local out-of-sample predictive performance, subsequent results can be informed by out-of-sample results from previous iterations, making it not strictly out-of-sample for any future iteration steps from the perspective of the analyst's knowledge. As such, in an iterative workflow, local out-of-sample predictive metrics may still lead to overfitting, but the degree to which this biases the end results remains a topic for future research.

3.3.4. Predictions in a dynamic world

Time is one of the most precipitous sources of uncertainty and any attempt to forecast the future with a static, time-independent P(A)D model will only be meaningful if the opaque generator $\mathbb{G}$ is strictly stationary (i.e., its regularities are invariant to time). Otherwise, a P model needs to have an appropriate temporal resolution to deliver reasonable out-of-sample predictions beyond the empirical snapshot of the collected data. Moreover, since the precise details of temporal shifts are extremely hard to anticipate, a P(A)D model which claims universal predictive performance should regularly be subjected to the falsification of time.

This brings us to an important distinction when it comes to assessing out-of-sample predictive performance. Whenever we make our P(A)D model "blind" to certain observations in the original data set D and use these observations to assess our-of-sample predictive performance (as we do in any form of cross-validation, even those built for time series data [46]), we are essentially testing the model's ability to perform *induction* about the statistical regularities of $p^*(y)$ in a temporal snapshot determined by data collection. In such a scenario, however, we are not probing the model's ability to faithfully forecast the future, since the "left-out" observations are new only from the perspective of the model, but not from that of the modeler. Thus, cross-validation can sometimes be overly optimistic in estimating out-of-sample predictive performance, since a sample collected at a future date might exhibit surprisingly different properties (i.e., the P model would no longer be structurally faithful) than the sample currently at hand.

Why would the empirical distribution $p^*(y)$ change over time? One reason can be that the hidden properties of the generator $\mathbb{G}$ itself may change, bringing about alterations in the statistical properties of $p^*(y)$. For instance, strong auto-correlations in financial time series are notoriously short-lived due to feedback processes and market adaptation [276]. Yet another reason can be that new sources of noise contaminate future data D in unexpected ways. For instance, a sensor in a measurement device may break and yield incorrect data or case reporting policies during an ongoing pandemic may switch between waves. However, the P(A)D model may have no mechanism to adapt to any of these changes and its out-of-sample predictive performance would likely suffer.

Within our model taxonomy, prediction failures due to changes in $p^*(y)$ concern misaligned assumptions about temporal invariances embodied in the P model's structure. One way to revise these assumptions is to include time-varying parameters θ_t in the P model, with the corresponding time-invariant parameterization being a special (and more parsimonious) case. For instance, this can be achieved within the *superstatistics* framework [13], which aims to represent heterogeneous dynamics through a superposition of multiple stochastic processes at different temporal scales [195]. In any case, researchers should bear in mind that static P(A)D models are not designed to deal with *things that move*, so, as simple as it sounds, time remains a key arbiter of the quest for universal substantial conclusions or robust predictive systems.

3.4. Fairness

Fairness in the context of model building aims to ensure that model-guided decisions are equitable, with a specific focus on groups that differ in protected attributes, such as sex, gender, or ethnic background [59, 9]. In a relatively narrow sense, fairness is a primary concern for P(A)D models, as it applies to real-world outcomes and their real-world reverberations owing to the connection between a P model's structure and data D. However, purely simulation-based P(A) models are not exempt from fairness considerations, especially when used to guide important public policies and decision support systems [48, 8, 218]. In the following, due to its predominant share in the literature, we will examine the fairness of P(A)D models from two different perspectives, namely, from the perspectives of psychometric measurement and predictive modeling.

3.4.1. Measurement fairness

In psychometric measurement theory, the aim is to estimate people's scores on latent psychological traits, for example, general intelligence, creativity, or aptitude for university programs [76]. In the model-based literature of psychometric measurement, namely Item Response Theory (IRT; [290, 82, 42]), two major aspects of fairness have received considerable attention.

First, we need to ensure that the observable features (i.e., items) have been selected and administered in a fair way [200, 36, 5]. This aspect does not appear to be immediately model-based, since it concerns the data collection process as well as causal assumptions about the latent traits' influence on the item responses [36]. However, some of its requirements can be checked via P(A)D models in the form of differential item functioning (DIF) analysis [140, 222]. When investigating DIF, the item parameters ζ_i of item i are allowed to vary across groups g and their P(A)D model's posteriors are compared to verify their statistical equivalence. That is, we aim to examine whether $p(\zeta_i \mid \tilde{y}, g) \approx p(\zeta_i \mid \tilde{y}, g')$ holds for all pairs of considered groups g and g' and all items i.

Second, we need to estimate the latent traits of all individuals with a similar degree of uncertainty [43]. In the context of P(A)D models, this means that the posterior of trait η_j for person j has approximately the same entropy across all individuals being compared, that is, $\mathbb{H}(\eta_j \mid \tilde{y}) \approx \mathbb{H}(\eta_{j'} \mid \tilde{y})$ for all pairs of individuals j and j'. This turns out to be a difficult, sometimes even unachievable goal: Due to floor and ceiling effects arising in almost all psychometric tests, the resulting information is non-uniform across the latent trait space in non-linear IRT models [290, 48, 43]. As a result, more extreme latent trait scores will be estimated less precisely than more average scores. As a partial remedy, one may try to ensure that the information gain about all individuals' trait scores at least exceeds a minimal, application-specific threshold [43].

3.4.2. Predictive fairness

What we term *predictive fairness* has its origins in the field of machine learning [258, 9]. We will define predictive fairness directly on PD models because there is no hope that a P model can yield fair decisions for all possible training data; after all, training data may themselves be biased against protected groups [258, 9]. And while we define it as a utility of PD models, it also automatically pertains to a corresponding PAD model, unless the posterior has a simple analytic form.

Mathematically, for individual-level decisions, we consider a PD model-specific decision rule $d(x \mid \tilde{x}, \tilde{y})$ that outputs a decision for each admissible vector of attribute values x given training data $D = (\tilde{x}, \tilde{y})$ consisting of observed attribute values $\tilde{x}$ and corresponding decision-relevant outcomes $\tilde{y}$ in a supervised learning context. If we consider only binary decisions to simplify notation, we can write the decision rule as

$$d(x \mid \tilde{x}, \tilde{y}) := \begin{cases} 1 & \text{if} \quad \bar{r}(x \mid \tilde{x}, \tilde{y}) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

with

$$\bar{r}(x \mid \tilde{x}, \tilde{y}) := \int r(x, \theta)\, p(\theta \mid \tilde{x}, \tilde{y})\, d\theta \tag{33}$$

being a real-valued (expected) *risk score* of x that is obtained as an expectation over the PD model's posterior $p(\theta \mid \tilde{x}, \tilde{y})$. The decision (e.g., whether to give someone a loan or release a defendant while they await trial) is then made by comparing the risk score against a pre-defined threshold τ. The conditional risk score $r(x, \theta)$ determines how the P model and its parameters θ are used for assessing risk. For example, the risk score could be the mean of the PD model's predictive distribution given feature value x and parameter value θ:

$$r(x, \theta) := \int y\, p(y \mid x, \theta)\, dy. \tag{34}$$

Conditional risk scores do not necessarily have to rely on the predictive distribution. Rather, they may also be based on latent model quantities, such as psychometric trait scores obtained from IRT P(A)D models [290, 82, 42], which bridges the gap between measurement and predictive fairness.

There are different classes of predictive fairness criteria considered in the literature, among others *anti-classification* [35, 59] and *classification parity* [59, 19] (also known as *statistical parity*; [57]). Even within these classes, criteria are partially incompatible and neither of them can actually ensure universal fairness, but we can still learn from their limitations [59, 57, 9, 19]. In the context of such criteria, we differentiate between protected attributes x_p (e.g., sex, gender, or ethnic background) and other, unprotected attributes x_u such that $x = (x_p, x_u)$. Anti-classification requires that protected attributes x_p (or their proxies; [35]) are not used in model-based decisions at all, which mathematically translates to

$$d(x \mid \tilde{x}, \tilde{y}) = d(x' \mid \tilde{x}, \tilde{y}) \quad \text{for all} \quad x, x' \quad \text{with} \quad x_u = x'_u. \tag{35}$$

In our PAD model taxonomy, this can simply be realized by using a PD model with $p(\theta \mid \tilde{x}, \tilde{y}) = p(\theta \mid \tilde{x}_u, \tilde{y})$ and conditional risk score $r(x, \theta)$ that is independent of x_p as well. Anti-classification approaches have two main drawbacks. First, protected attributes can often be predicted fairly well from unprotected attributes, which makes it impossible to be completely agnostic about them [89]. Second, empirical risk distributions (after removing all unfair risk influences) may differ across values of x_p, such that ignoring the latter may actually lead to unfair decisions against the groups one originally attempted to protect [59].

Differently, classification parity comprises a class of fairness criteria that requires the population distribution of certain decision metrics to be the same across all values of the protected attributes [59, 19]. Using *demographic parity* [89] as an example, we would require that the decision's distribution itself, as implied by the distribution of attributes x in the considered population, to be independent of the protected attributes:

$$p(d(x \mid \tilde{x}, \tilde{y}) \mid x_p) = p(d(x \mid \tilde{x}, \tilde{y})). \tag{36}$$

Contrary to anti-classification, we usually have to incorporate the protected attributes into the P model in the first place to ensure any kind of classification parity [19]. In the context of psychological tests, for example, this could be achieved by imposing group-specific norms of comparison [263]. Yet, classification parity does not guarantee universal fairness either, whenever the true risk score distribution (after removing all unfair risk influences) varies between groups defined by the protected attributes [59].

The shortcomings of these predictive fairness definitions highlight that requiring a certain outcome – the decision itself (anti-classification) or aspects of its population distribution (classification parity) – to be independent of the protected attributes may be insufficient. Towards the goal of achieving fairness through a PD model, the underlying P model needs to be causally consistent (see also Section 3.1) in a way that considers how the protected attributes x_p relate to the causal graph that includes all the valid, unprotected attributes x_u and the outcome y [35]. In addition, the training data D needs to be representative of the true (unbiased) outcome distribution $p^*(y)$. It goes without saying that these are complicated, application-specific tasks that require contributions from various scientific fields and considerable domain expertise.

What is more, fair decisions, regardless of their modeling context, need to take into account that the same decision may affect different people (and their surroundings) differently and that these differences may be related to both protected and unprotected attributes. More formally, we need to consider the decision $d(x \mid \tilde{x}, \tilde{y})$ in a context $C(x)$ that only together determine the output of a utility function $U(d(x \mid \tilde{x}, \tilde{y}), C(x))$, which offsets all possible gains and losses caused by the decision. Obtaining such a function could steer a decision towards fairness as quantified by equal utility outcomes across protected groups.

At an even higher level, we should consider taking sufficient precaution that (anticipated) political decisions or societal processes triggered by anonymous modeling results do not lead to unfair treatment of protected groups. However, such considerations may come into conflict with the principle of scientific freedom, in which case a careful ethical analysis of the specific situation becomes mandatory.

3.5. Structural faithfulness

In most data analysis scenarios, we have a reasonable amount of qualitative prior knowledge about the data structure and the data generating process, even if we don't know the precise analytic relation between the two. In particular, this knowledge concerns the scales of variables to be modeled, the dependencies between observations, as well as physical constraints, such as symmetries or invariances. The *Structural Faithfulness* utility captures how well a P model incorporates such knowledge. Structural faithfulness is at the core of statistical modeling, be it Bayesian or otherwise, as it determines the probability distributions we assign to our observed and unobserved variables, the parameters we add to our P models, and the assumptions we can justifiably make to simplify reality.

Moreover, we can roughly distinguish between probabilistic structure and functional structure, which are related to the modeler's degree of ignorance regarding the problem at hand. Purely statistical models aim to capture the *probabilistic structure* of $p^*(y)$, without making reference to *functional structure* of the hidden generator $\mathbb{G}$. Non-deterministic mechanistic models, on the other hand, aim to capture the functional structure of $\mathbb{G}$ (usually represented by physical constraints), such that the probabilistic structure of $p^*(y)$ can be reproduced or explained. For instance, when we study the dynamics of a phenomenon via stochastic differential equations, functional faithfulness refers to the mathematical form of the differential equation and probabilistic faithfulness refers to the fidelity of the stochastic assumptions.

To us, it remains unclear how to measure structural faithfulness in an absolute sense and we see it primarily as a relative metric. What is more, structural faithfulness consists of multiple components that may each favor a different P model. For example, model P_1 might take a known symmetry into account that model P_2 ignores, while P_2 might assign a more appropriate distribution to a response variable than P_1 does. In this case, none of the two P models would actually be more structurally faithful than the other, at least not uniformly so.

3.5.1. Variable scales

The scale of a variable determines not only what information it represents but also how it should ideally be treated within a P model. For example, if the response variable consists of count data without a known or practically reachable upper bound, we should model this data via an appropriate (unbounded) discrete distribution (e.g., Poisson, or some of its generalizations) to sensibly capture the aleatoric (irreducible) uncertainty in those count responses [302, 99, 313]. What is more, this ensures that the variables' natural boundaries are respected (e.g., lower bound of zero for count data), such that the corresponding model predictions cannot go beyond the data space that is possible in reality (see Figure 7 for an illustration). As another example, if our response variable is ordinal, that is, it consists of discrete ordered categories without guarantees that the categories can be considered equidistant, we should model such data via an ordinal distribution [201, 179, 49]. The same points hold also for predicting variables even if they are not explicitly modeled with a distribution [44, 115]. Failure to consider the variable scales in P models can have detrimental consequences for the validity of the obtained results [115, 44, 179].

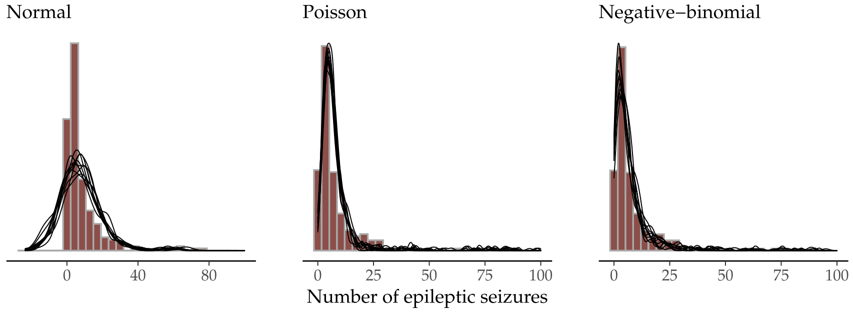

Fig 7. *Posterior predictive checks [102] of epilepsy treatment data [287]. The response variable is the number of epileptic seizures of patients in a given time interval, that is, a count variable without a known upper bound. Results are shown for three PAD models with different likelihoods (shown as facets) and posteriors approximated via MCMC in Stan [286]. Histograms indicate observed data and each black line indicates one draw from the posterior predictive distribution of the corresponding PAD model, smoothed via continuous density estimation. For Poisson and negative-binomial likelihoods, posterior predictions are in fact counts but are still displayed as smoothed continuous densities to ease readability and comparability across facets. As is clearly visible on the left-hand side, the PAD model with normal likelihood predicts a lot of theoretically impossible negative counts and can neither predict the spike at counts close to zero, nor the heavy right tail.*

Equivalently, respecting the intrinsic scales of all quantities included in a P model can help to avoid unreasonable parameter estimates or implausible (or worse, impossible) predictions.

3.5.2. Probabilistic structures

Observed data often exhibits specific probabilistic structures that can be inferred from (qualitative) understanding of the data-generating process. For example, if we collect psychometric data from multiple students in the same class, it is highly unlikely that the data points will be mutually independent (e.g., because students share the same teacher, rooms, peers, etc.). This situation is prototypical for the application of *multilevel models*, which aim to capture such dependencies [110, 12, 40, 41]. Multilevel models treat such dependencies of observations belonging to the same group as equivalent to variation between groups [110]. In other words, if there were no variation between groups, there would be no structural dependency of observations within groups (at least none elicited by this grouping structure).

There are three major types of structural dependence between groups that can be expressed as multilevel models: exchangeable, directed, and undirected [260, 100, 103], illustrated schematically in Figure 8.

Exchangeable groups are the most common assumption in multilevel models and imply that (before seeing any data) we hold the same prior beliefs about each of the groups but assume they are all drawn from the same population (e.g., students within classes, classes within schools, schools within cities, etc.). In the

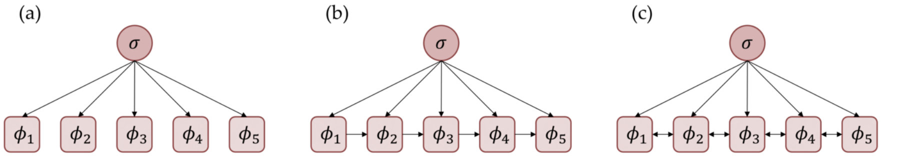

FIG 8. *Graphs illustrating common probabilistic structures. Rectangles depict nested parameters within a given probabilistic structure. Circles depict the corresponding hyperparameters. (a) Exchangeable parameters; (b) Conditionally dependent parameters with a directed (e.g., temporal) dependency structure. (c) Conditionally dependent parameters with bidirectional (e.g., spatial) dependency structure.*

most simple case (i.e., two-level structure, univariate and normally distributed parameters), we would specify a univariate normal prior for each group indexed by i and group parameter ϕ_i as

$$\phi_i \sim \text{Normal}(\mu, \sigma), \tag{37}$$

where μ and σ are the mean and standard deviation parameters shared across groups, respectively. Typically, we would estimate the across-group parameters from the data along with the group-specific parameters ϕ_i themselves.

In directed dependency structures, adjacent groups are assumed to have directed influence on each other in a way that group i can affect group j, but not vice versa. The most common example is temporal autocorrelation where a variable at time i can potentially be influenced by a variable at time $i-1$ [278, 103]. For a univariate Gaussian random walk, we would formalize this assumption with the following prior

$$\phi_i \sim \text{Normal}(\phi_{i-1}, \sigma). \tag{38}$$

In undirected dependency structures, the influence of adjacent groups can go both ways, with spatial autocorrelation being the most common example [22, 106, 211]. For example, in (spatial) conditional autoregressive (CAR) structures [22], we could write down the prior on the group coefficients as

$$\phi_i \sim \text{Normal}\left(\frac{1}{|\mathbf{N}_i|}\sum_{j\in\mathbf{N}_i}\phi_j,\, \sigma\right), \tag{39}$$

where $\mathbf{N}_i$ is the set of groups that are neighbours of group i. Importantly, a shared feature of these dependency structures is that they are agnostic towards the underlying causal mechanisms – their purpose is purely to accurately represent the inherent probabilistic structure of the observed data [106, 312, 103].

But what if the data-generating process suggests a certain kind of dependency for which we find no empirical support? For example, shall we retain a grouping term of classes even if the PAD model suggests close to zero variation between groups? There are good arguments for both choices. On the one hand, excluding such a term implies a simpler model with higher parsimony [11] (see also Sections 3.6), although the increase in parsimony will be quite small due to the *partial pooling* property of multilevel models induced by their hierarchical priors if there is a sufficient number of groups [110, 138]. On the other hand, including the term sets a good example for future replications and applications of the same P model, in the same or different contexts. That is, if someone applies this P model to a new data set, they may very well find the between-group variation under question to be non-zero, thus justifying the inclusion of the corresponding grouping term.

3.5.3. Physical constraints

In the domains of physics and natural sciences, we tend to have strong prior knowledge about the functional P model structure in the form of known hard constraints such as symmetries, invariances, or conservation laws [292, 251, 157, 7]. For example, a harmonic oscillator expressed by the second-order differential equation

$$\ddot{x}(t) = k\, x(t), \tag{40}$$

with functional solution x, second derivative $\ddot{x}$, as well as constant k, represents an isolated system that is energy conserving [187].

Similar to a harmonic oscillator, most physical hard constraints can be expressed via differential equations whose direct inclusion in a P model is computationally demanding if we do not have access to an analytic solution [62, 173, 282]. Accordingly, building a more flexible, data-driven P model as a surrogate is a computationally attractive choice [173, 47]. Still, even for such a surrogate, it remains beneficial to incorporate known physical constraints to eliminate the need to learn them directly from data. This is likely to increase the model's data efficiency, that is, the amount of data required by the model to achieve a certain predictive goal [251, 173]. The discussion about physics-informed modeling is particularly prominent in core areas of high-dimensional machine learning, such as neural networks that tend to be very data-hungry [251], but in principle applies to all P models created for representing data with known physical constraints.

3.6. Parsimony

Parsimony refers to the formal simplicity of a Bayesian model; some might define it as the conceptual or mathematical elegance of the underlying interpretative framework. Here, we view parsimony as a quantifiable property of a Bayesian model. We treat it also as a relative quantity – it is always possible to propose a more complex model (or possibly a simpler one) which is equally consistent with the available data.

Within our PAD framework, we will distinguish two types of parsimony: P-parsimony and A-parsimony. P-parsimony characterizes the formal simplicity of a P model and should be measurable from the structure of the joint distribution

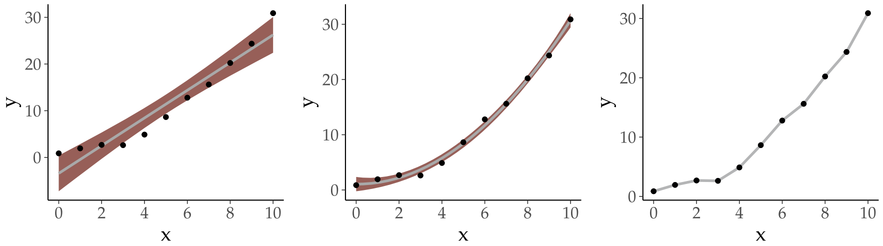

FIG 9. *P models of different complexity applied to a data set D of 11 observations following a quadratic relationship in expectation. Left: Most parsimonious, linear model with a 3-parameter likelihood $y \sim$ Normal$(\beta_0 + \beta_1 x, \sigma)$. This model is too simple for the data. Center: Slightly less parsimonious, quadratic model with a 4-parameter likelihood $y \sim$ Normal$(\beta_0 + \beta_1 x + \beta_2 x^2, \sigma)$. This model's complexity is just right for the data. Right: Least parsimonious, linear interpolation model between adjacent points that has as many parameters as observations in the data (1 intercept and 10 linear slopes). This model is too complex for the data. Shaded areas indicate 95% credible intervals of the regression line for models where this uncertainty can be computed.*

$p(y, \theta)$. A-parsimony characterizes the simplicity of an approximator and should be measurable through the interface of A. The former is directly related to the theoretical appeal of a P model's probabilistic assumptions; the latter is directly associated with the usability of an approximator.

3.6.1. P-parsimony

In many real-world modeling scenarios, we have limited data and strive for P models that can capture all relevant latent properties with as little data as possible (see Figure 9 for a simple illustration). One particular aspect of this goal is captured by the dimensionality of the parameter space, whereby higher parsimony simply means lower parameter dimensionality. Canonical examples for high parsimony are physical simulators defined by complex (white-box) forward models with intractable likelihoods [62]. The latter are informed by strong subject matter knowledge and are thus able to maintain low parameter dimensionality (e.g., consider the harmonic oscillator Equation (40), which only requires a single parameter to describe highly non-linear, non-monotonic behavior). On the other end of the spectrum are neural network models that tend to use simple likelihoods (e.g., Gaussian or categorical), but are characterized by an extremely high parameter dimensionality and large compositions of non-linear transformations, such as GPT-3 featuring 175 billion parameters [92]. In a way, we need to compensate for our lack of *a priori* knowledge (or inability/unwillingness to use it) by applying less parsimonious models that replace more restrictive model structures with a heightened hunger for data.

The motivation for parsimony is related to other utilities as well, since more parsimonious P(A)D models tend to require less data to achieve the same reduction in epistemic uncertainty (parameter recoverability; Section 3.2) and predictions (predictive performance; Section 3.3), and tend to be easier to comprehend in real-world applications (interpretability; Section 3.7). Still, we can construct chaotic models – where minimal changes in the parameters lead to strong changes in the predictions – that are highly parsimonious, yet uninterpretable and extremely flexible in terms of the function space they can approximate [239]. However, most P models applied in current practice do not exhibit such chaotic behavior.

Despite its intimate connection to other utilities, we think that parsimony deserves to be a utility in its own right, harmonized with Occam's razor: Given two models, and other things being equal, one should choose the more parsimonious one [31]. Increasing the parsimony of a model (or a scientific theory, for that matter) implies making more restrictive assumptions (i.e., reducing the function space that can be theoretically approximated by the model), thus increasing its *falsifiability*: We can more easily create situations where the model is wrong. Furthermore, in applied settings, sparser models may lead to more efficient data collection and more economical measurement designs (i.e., fewer variables to measure or less acquisition trials in design optimization) [234]. Nevertheless, the strive for parsimony may not always be a useful guide to our scientific exploration, if the aesthetics of parsimonious P models make us blind for potentially more appropriate (e.g., in terms of other utilities), but less parsimonious representations. For example, the strive for parsimony may be one of the factors that has stalled the scientific progress in the foundations of physics during the past decades [141].

Effective number of parameters There are different ways to measure parsimony, with simply counting the number of parameters[4] of a P model being the most straightforward approach. For simple models, such as linear regression, this measure of parsimony matches the concept of *degrees of freedom* (DoF) in frequentist statistics. In the same way, the DoF concept becomes awkward even for slightly more complex models [150], the former is not a generally useful measure of parsimony either [239]. The reason for this is that, from a Bayesian perspective, any prior information on a parameter increases a P model's parsimony, such that the *effective number of parameters* (ENP), might be substantially smaller than the nominal number of parameters [296]. The same mechanism also underlies the difficulty in computing the DoF of test statistics in frequentist multilevel models, because random effects distributions are equivalent to priors [138].

There are several ENP measures in the literature [277, 309, 296, 240], often defined in the context of information criteria. For the information criterion based on leave-one-out cross-validation (LOO-CV), ENP is measured as the sum of the differences between the pointwise log predictive densities of the full posterior

[4] More precisely, we have to count the minimal number of unconstrained parameters that can be invertably transformed to the space of the original model parameters. For example, a simplex parameter vector of length K is equivalent to only $K-1$ unconstrained parameters because the K-th one is determined by the sum-to-one constraint.

and the pointwise log predictive densities of the LOO posteriors [296]:

$$\begin{aligned}\mathrm{ENP}_{\mathrm{LOO}} &= \sum_{n=1}^{N} \left(\log p(y_n \mid y) - \log p(y_n \mid y_{-n})\right) \\ &= \sum_{n=1}^{N} \left(\log \int p(y_n \mid \theta)\, p(\theta \mid y)\, d\theta - \log \int p(y_n \mid \theta)\, p(\theta \mid y_{-n})\, d\theta\right).\end{aligned} \tag{41}$$

The notation y_{-n} indicates that the n-th data point in y has been excluded. As more parameters are added to the model, the in-sample predictive performance represented by $\log p(y_n \mid y)$ grows more quickly than the out-of-sample predictive performance represented by $\log p(y_n \mid y_{-n})$ such that the sum of their pointwise differences grows. This provides an intuition why $\mathrm{ENP}_{\mathrm{LOO}}$ can be considered a measure of parsimony. Its concrete interpretation as an effective number of parameters is inspired by the following observation: When using very wide or even completely flat priors over all parameters, $\mathrm{ENP}_{\mathrm{LOO}}$ will roughly coincide with the nominal number of parameters, but becomes smaller than the latter in the presence of prior information [296].

Bayesian LOO-CV can usually be computed efficiently via importance sampling without any model refitting, and so can $\mathrm{ENP}_{\mathrm{LOO}}$ be computed without any actual refitting [296, 300]. For a large number of observations N, $\mathrm{ENP}_{\mathrm{LOO}}$ can be asymptotically approximated by the sum of the full posterior variances over the pointwise log-likelihood values, which is the ENP estimate used in the widely applicable information criterion (WAIC) [309]:

$$\mathrm{ENP}_{\mathrm{LOO}} \approx \mathrm{ENP}_{\mathrm{WAIC}} = \sum_{n=1}^{N} \mathrm{Var}_{p(\theta \mid y)} \left[\log p(y_n \mid \theta)\right] \tag{42}$$

Intuitively, as the number of parameters grows, so does the epistemic uncertainty in the posterior, which leads to an increase in the variance of posterior predictive quantities, such as $\log p(y_n \mid \theta)$. The WAIC approximation of LOO-CV performance can be quite unreliable so using $\mathrm{ENP}_{\mathrm{LOO}}$ is highly recommended whenever possible [296]. What becomes apparent in these equations is that parsimony, at least when measured through these ENPs, may depend on the specifically realized data $\tilde{y}$, and as such needs to be defined over PD models. This is specifically true for models with hierarchical priors, where the amount of hierarchical shrinkage (i.e., the influence of the hierarchical priors) is data-dependent [110]. Practically, the posterior integrals in (41) and (42) for PAD models are efficiently approximated via Monte Carlo estimates based on posterior draws from an approximator [296].

The huge advantage of these ENP measures is that they do not need to be aware of the internal structure of a P model, but only require its predictive outputs in the form of pointwise log-likelihood values. However, the need for the latter has the drawback that ENP measures do not work natively with P_I models due to their lack of tractable likelihoods; unless one has learned not only

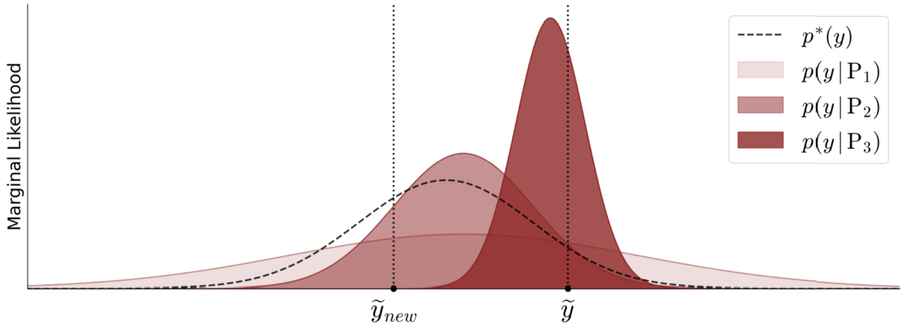

FIG 10. *Hypothetical scenario with three P models of descending complexity: P_1, P_2, and P_3. The most complex model P_1 can account for the broadest range of observations at the cost of diminished sharpness of its marginal likelihood; in contrast, the simplest model P_3 has the sharpest marginal likelihood which concentrates onto a narrow range of possible data. Even though the observed data $\tilde{y}$ is well within the generative scopes of models P_1 and P_2 too, the simplest model P_3 has the highest marginal likelihood at $\tilde{y}$ among the three candidates and is therefore favored from a marginal likelihood perspective. However, the higher relative marginal likelihood of the simplest model P_3 is a poor proxy of its predictive performance for new data sets, as it assigns close to 0 density to the new data set $\tilde{y}_{\mathrm{new}}$, suggestive of overfitting. The model P_2, whose marginal likelihood is closest to the data-generating distribution p^*, would have been favored, had $\tilde{y}_{\mathrm{new}}$ instead of $\tilde{y}$ been used for computing the associated Bayes factors.*

the model's posterior but also its likelihood density during training [314]. What is more, if the model includes residual dependencies between observations, the pointwise (log-)likelihood may not be available, even if the joint likelihood is analytic [39].

Prior P-parsimony In the above-described ENP definitions, we integrate over the posterior distribution and so, in this sense, measure *posterior parsimony*. This naturally raises the question of whether we can define measures of *prior parsimony* as well. In a Bayesian setting, prior parsimony is automatically embodied in the marginal likelihood (sometimes called Bayesian evidence) [158, 189, 183], which we already encountered in our discussion on prior predictive performance (see Section 3.3.2). As a reminder, we obtain the marginal likelihood by marginalizing the joint P model over its prior

$$p(y) = \mathbb{E}_{p(\theta)}\left[p(y \mid \theta)\right] = \int p(y \mid \theta)\, p(\theta)\, d\theta. \tag{43}$$

Accordingly, we can interpret the marginal likelihood as the expected probability of generating data y from a P model when we randomly sample from the prior $p(\theta)$. Through the prior's role as a weight on the likelihood, the marginal likelihood encodes a probabilistic version of Occam's razor by penalizing the prior complexity of a P model [158, 189].

However, the marginal likelihood is not an explicit measure of parsimony; rather, it represents an implicit relative quantity which combines prior parsimony with the ability of a P model to fit the data by considering its entire generative scope (see Figure 10). Following [189, Chapter 28], we can illustrate the above conflation by assuming that the posterior of a P(A)D model is well represented by a (multivariate) Gaussian. In this case, the marginal likelihood can be approximated as:

$$p(y) \approx p(y \mid \theta_{\mathrm{MP}}) \times p(\theta_{\mathrm{MP}}) \det(\boldsymbol{H}(\theta_{\mathrm{MP}})/2\pi)^{-\frac{1}{2}}, \tag{44}$$

where θ_{MP} is the posterior mode and $\boldsymbol{H}(\theta_{\mathrm{MP}})$ is the Hessian of the likelihood evaluated at θ_{MP}. The multiplicand $p(\theta_{\mathrm{MP}}) \det(\boldsymbol{H}(\theta_{\mathrm{MP}})/2\pi)^{-\frac{1}{2}}$ is termed an *Occam factor* and represents the factor by which a P(A)D model's parameter space contracts as the prior is updated to the posterior based on the information contained in D. Thus, under the Gaussian assumption, the magnitude of the Occam factor is an explicit measure of prior complexity (i.e., inverse prior parsimony) related to the information gain a P model can achieve over its generative scope [189, 183]. Consequently, a P model with a vague prior will incur a larger penalty by the Occam factor than a different P model with a sharper prior, provided that both models share the same likelihood. However, if the Gaussian assumption is inadequate, the approximation of Equation (44) can sustain a large error and may no longer be useful. Unfortunately, we are not aware of a more general decomposition of the marginal likelihood into a prediction factor and a parsimony factor, as is the case with $\mathrm{ENP}_{\mathrm{LOO}}$.

A closely related concept is the principle of Minimum Description Length [MDL, 255, 133], which views parsimony through the lens of information theory. In the MDL framework, a probabilistic model represents a *coding scheme* designed to describe the data $\tilde{y}$. Accordingly, a parsimonious P model provides a concise description of the data in terms of code length (relative to a competing P model). Note, that MDL is not a unique measure, but rather an umbrella framework for deriving measures of parsimony/complexity in various application contexts (see [133] for a comprehensive exposition). For instance, in a Bayesian context, one can show [133] that a canonical measure of description length for model P is given by

$$\mathrm{DL} = -\log \int p(y \mid \theta)\, p(\theta)\, d\theta, \tag{45}$$

which we recognize as the negative logarithm of the marginal likelihood introduced in Equation (43). In this way, MDL not only highlights the theoretical connection between Bayesian model comparison and information theory but also provides a principled way for deriving new measures of prior parsimony in future basic research.

Sparsity-inducing priors Another perspective on P-parsimony is provided by sparsity-inducing priors, especially global-local shrinkage (GLS) priors [293, 26, 291]. These priors will shrink redundant coefficients towards values close to zero, inducing sparsity in the posterior.[5] GLS priors can be applied in many

[5] Shrinkage priors will not shrink coefficients exactly to zero but only close to it. Thus, such coefficients remain in the regression equation but exert a minimal impact on predictions. If desired, exact sparsity can be achieved in a second step via a variable selection procedure [241, 53, 234].

model classes, including linear and generalized linear models, non-linear and non-parametric function estimation, time series, as well as deep neural networks [293, 118, 26, 268]. Here, we focus our discussion on Gaussian linear models as this case is most intuitive and theoretically best understood. Given a linear regression model in its simplest form, GLS priors are defined on the K regression coefficients β_k as follows:

$$\beta_k \sim \text{Normal}\left(0, \lambda_k^2 \tau^2\right), \quad \lambda_k \sim p(\lambda_k), \quad \tau \sim p(\tau), \tag{46}$$

where λ_k denotes the local scale parameter unique to each coefficient and τ denotes the global scale parameter that is shared across all coefficients. The choice of the hyperpriors $p(\lambda_k)$ and $p(\tau)$ determines the specific properties of the GLS prior, leading to, for example, the horseshoe [51, 240] or the R2D2 prior [325, 1]; see [293] for a comprehensive overview.

The implied posterior of the coefficients has a highly interesting relationship with the maximum likelihood (ML) estimate $\hat{\beta}_k$ that can be obtained from the same likelihood and data but under the assumption of flat priors on the coefficients. Concretely, and assuming that the ML estimate exists, the posterior mean $\mathbb{E}_{\theta|y}(\beta_k)$ can be computed as follows [240, 1]:

$$\mathbb{E}_{p(\theta|y)}[\beta_k] = (1 - \kappa_k)\hat{\beta}_k, \tag{47}$$

with

$$\kappa_k = \frac{1}{1 + a_k \lambda_k^2 \tau^2}. \tag{48}$$

Here, a_k is some constant that depends on the response's and the k-th predictor's scales. Accordingly, the smaller λ_k and τ, the stronger the shrinkage of β_k to zero, relative to the ML estimate $\hat{\beta}_k$. Conversely, the larger λ_k and τ, the closer the posterior mean of β_k will be to $\hat{\beta}_k$. Given these properties, κ_k are called *shrinkage factors* [240, 1].

The model leading to the ML estimate has K coefficients, which are all counted fully when it comes to determining the number of parameters (see above). Since the posterior mean β_k implied by the GLS prior is equal to $(1 - \kappa_k)\hat{\beta}_k$, we see that summing over all $(1 - \kappa_k)$ terms can be considered a measure of the *effective number of coefficients* [ENC, 240]:

$$\text{ENC}_{\text{GLS}} = \sum_{k=1}^{K} (1 - \kappa_k). \tag{49}$$

In contrast to the above ENP measures, ENC_{GLS} is essentially limited to linear models. What is more, ENC_{GLS} only considers regression coefficients, not necessarily all P model parameters (e.g., it ignores the residual standard deviation σ). These are not the only differences between these measures though. Even though both are derived as generalizations of simply counting parameters, the ENC measures focus on posterior variance (which is explicit in the definition of ENP_{WAIC}), while ENC_{GLS} focuses on the posterior mean. Thus, they consider different aspects of the posterior when measuring parsimony. Studying the relationships between these measures more closely would be an interesting endeavor for future research.

3.6.2. A-parsimony

As we discussed in Section 2.3 concerning PA models, posterior approximators can range from relatively simple optimization algorithms to high-dimensional parametric models (e.g., neural networks) which themselves can be viewed as standalone P models (e.g., Bayesian neural networks). The notion of A-parsimony intends to capture our intuition that these different approximators have varying degrees of complexity. Here, we propose a very straightforward definition of A-Parsimony: The cardinality of the hyperparameter space $\mathcal{H}$ available for fine-tuning through the implementation interface $\mathcal{I}$ of the underlying mathematical algorithm $\mathcal{A}$. For instance, the widespread use of MCMC in Bayesian inference is partly because probabilistic programming languages provide relatively simple interfaces, which abstract away a staggering multitude of hyperparameters of complex MCMC samplers [e.g., NUTS, 139]). On the other hand, neural approximators [e.g., 249, 126] inherit the vast hyperparameter spaces of deep neural networks and are thus currently still rather challenging to apply or fine-tune [301].

A-parsimony is not only relevant for the usability of approximators, but also plays an important and limiting role in comparison or benchmarking studies assessing the relative performance of different approximators. Suppose we wish to compare approximator A_1 having no hyperparameters with approximator A_2 having a single continuous hyperparameter $h \in [0, 1]$, in the context of some P model. A comparison of approximators must naturally be based on some metric (or a set of metrics) $q(\mathrm{A}, \mathrm{P})$ which quantifies the approximation quality of A with respect to a given P model (e.g., the distance between corresponding PD and PAD models or the estimation speed of the approximator). However, even for the simple scenario outlined above, it is not clear how to systematically carry out such a comparison due to the presence of hyperparameters. One approach would be to approximate the average approximation quality given by $\int_0^1 q(\mathrm{A}_2(h), \mathrm{P})\, p(h) dh$ of A_2 and compare it to $q(\mathrm{A}_1, \mathrm{P})$. Another approach would be to seek the best approximation quality given by $\max_{h \in [0,1]} q(\mathrm{A}_2(h), \mathrm{P})$ and compare it to that of $q(\mathrm{A}_1, \mathrm{P})$. Needless to say, the difficulty of ranking and benchmarking approximators with large hyperparameter spaces drastically increases, which makes A-parsimony a key limiting factor as well as a desirable utility to improve upon.

Finally, A-parsimony is related to robustness (see Section 3.10) and convergence (see Section 3.8), as the presence of multiple hyperparameters raises the question of how to choose hyperparameter settings which i) lead to stable results and ii) generalize to various applications of a PA(D) model. For some approximator classes (most notably, MCMC) and P models (e.g., linear models), empirical guidelines and theoretical considerations may suggest relatively robust default choices. For newer approximator classes (e.g., neural density estimators) or more exotic applications, some form of sensitivity analysis or hyperparameter search might be necessary to ensure sufficient robustness or generalizability.

3.7. Interpretability

Interpretability of a P(A)(D) model can be qualitatively defined as "the degree to which a human can understand the cause of a [model-based] decision" [206] or as "the degree to which a human can consistently predict the model's result" [159]. A more precise, perhaps even mathematical, definition is difficult to provide given the context and expertise-dependent nature of interpretability, but there is progress in this direction [73]. In any case, achieving interpretability will help us understand *why* a P(A)(D) model behaves the way it does (e.g., in terms of predictive performance; see Section 3.3). Such understanding can have not only profound epistemological, but also far-reaching ethical and social implications [232, 73, 209].

According to [209], we can distinguish between intrinsic and post-hoc interpretability. The former is related to the intelligibility of the P(A)(D) model itself (i.e., its structure and parameters), whereas the latter is related to the *explainability* of the PAD model's results using auxiliary methods, such as permutation feature importance for neural networks [317] or random forests [149]. However, there is a conceptual ambiguity regarding the term in the recent literature. Some accounts use explainability as a synonym for interpretability in general [209], while others use explainability to refer solely to post-hoc interpretability [38]. In our PAD model taxonomy, we view only intrinsic interpretability as a utility of the P(A)(D) model. Differently, post-hoc interpretability is a utility of an *explanator* that is applied to the original PAD model's results – in fact, the explanator may just be another, more interpretable P(A)(D) model that is used as a surrogate [38]. Accordingly, the following discussion focuses only on intrinsic interpretability, to which we hitherto refer simply as *interpretability*.

P model interpretability relates to the general meaning of its parameters, so it makes sense to differentiate between the interpretability of P$_I$ and P$_E$ models since the two model classes often put different demands on the epistemic value of their parameters. Further, as we will see below, there are P models whose interpretability can be influenced by both data D and approximator A. As such, it can be necessary to further distinguish the interpretability of P, PD, and PAD models.

3.7.1. Interpretability of P$_I$ models

In P$_I$ models, most parameters correspond to real-world quantities or emergent properties, whose meaning can be understood independently of the P$_I$ model that is used to estimate them (see Section 2.1). For example, in a harmonic oscillator [187], the object's mass that serves as a parameter carries a meaning independent of the differential equation that describes the oscillator's behavior. As such, while the transformations performed to generate data from an P_I model are highly non-linear and often not analytically tractable [62], the interpretability of P_I models tends to be high (at least in the eyes of domain experts in the field).

However, even for domain experts, it can be exceptionally challenging to predict the generative behavior of a high-dimensional P_I model given a particular parameter configuration. This can be the case, even when a P model has a small number of readily interpretable parameters. Consider, for instance, the prototypical *logistic map* equation [198] given by

$$y_{t+1} = \rho\, y_t\, (1 - y_t) \tag{50}$$

and having only a single parameter $\rho \in [0, 4]$ which can be interpreted as *growth rate* in population dynamics modeling [281]. Despite its beguilingly simple form, the logistic map is known to develop chaotic behavior as the parameter ρ varies in the range from approximately $\rho \approx 3.56995$ to $\rho \approx 3.82843$. The model's generative behavior in this range is characterized by a periodic phase, intercepted by bursts of aperiodic fluctuations. And even though such behavior can be generally abstracted and described for a single parameter, for instance, with the help of bifurcation diagrams [119], it can quickly become less amenable to high-level descriptions when it results from the interaction of two or more parameters [10]. Unsurprisingly, Bayesian analysis of PD or PAD models based on an underlying chaotic P_I model has long been recognized as a challenging endeavor [20], requiring sophisticated approximators with surrogate likelihoods [279].

As alluded to above, the interpretability of high-dimensional P_I models will often depend on whether we focus on individual parameters and their functional role for data generation in isolation (i.e., first-order interpretability) or try to understand interactions between parameters as well as their joint contribution to the generation of y (i.e., higher-order interpretability). Accordingly, even for complex P_I models with dozens of parameters, we may still retain relatively high first-order interpretability through the theoretical embedding of each individual parameter, but higher-order interpretability may suffer, since multiple parameters can act similarly on y and interact in surprising ways due to non-linearity. For instance, the compartmental model of the early COVID-19 pandemics in Germany set up by [246] has 34 free parameters, each of which has a direct isolated interpretation, for instance, infection rate, number of initially exposed people, weekly modulation, or probability of detection. However, the exact interplay between these parameters in determining the actual reported number of daily cases might not be immediately obvious from the understanding of individual parameters alone or from the model equations themselves.

Finally, the higher-order interpretability of P_I models may change once they have been connected to data due to dependencies between parameters. Oftentimes, we choose a prior $p(\theta)$ which factorizes into independent components, reflecting our assumption of disentanglement or independent generative factors of variation. However, the resulting PD or PAD models will rarely conserve independence in their joint posteriors (e.g., due to loss of information or an inherent lack of disentanglement in the inverse model). A canonical example would be a strong posterior correlation between two parameters with initially independent priors, indicating that the parameters do not fulfil orthogonal functional roles for generating the data.

3.7.2. Interpretability of P_E models

In P_E models, the parameters do not need to correspond to real-world quantities or mechanisms. Rather, their meaning can often only be understood within the P_E model they are part of [112]. The archetypal P_E model is linear regression, where a regression coefficient β describes the linear relationship between a predictor variable and the response whilst holding all other predictors constant. As such, β has a clear meaning to an analyst with some statistical knowledge, provided that the measurement scales of predictor and response variables make sense for the task at hand. However, the requirement to hold all other predictors constant becomes impossible to fulfil if the predictors cannot be varied independently from each other, for example, because they are correlated in purely observational data or because some of them constitute interactions between already included predictors. As such, even for as few as four or five predictors, interpretability of their joint contribution becomes highly challenging unless predictors are mutually independent [209].

The use of non-linear, monotonic transformations in P_E models, such as link functions in generalized linear models [215] or non-linear activation functions in neural networks [272] further complicates the interpretability of an originally linear predictor structure. For example, when using the logarithmic link (equivalently, the exponential response/activation function), the originally additive relationships become multiplicative, resulting in exponential growth, which is much harder to comprehend for humans [304]. This then reduces the interpretability of the P_E model's parameters from both their signs and magnitudes to only their signs. If one were to apply non-monotonic transformations, the interpretability of the parameters' signs would be lost as well. In addition to non-linear transformations of the whole linear predictor term, every structural deviation from a (latent) linear structure further reduces interpretability. For example, interactions, polynomial terms, hierarchical structure [110], Gaussian processes [311, 253], or splines [98, 315] all make interpretation of a P_E model's parameters harder, if not impossible in some cases.

The interpretability of a P_E model may also be affected by the data utilized for parameter estimation. Accordingly, PD models may differ in their interpretability even if their underlying P_E model is the same. For example, when employing shrinkage priors for high-dimensional linear P_E models with lots of irrelevant predictors, the posterior of most regression coefficients will shrink to values very close to zero, effectively eliminating the corresponding predictors from the regression equation [240, 325] (see also Section 3.6.1). If only a few coefficients are substantially different from zero, the interpretability of the resulting PD model would be much higher than that of the original P_E model.

Finally, an approximator A may create a situation where a PA(D) model's interpretability deviates from that of the underlying P(D) model. However, that may only happen if the posterior approximation $p_{\mathrm{A}}(\theta \mid y)$ is qualitatively different from its analytic counterpart $p(\theta \mid y)$ due to an incomplete posterior exploration. A common case arises when the analytic posterior is multi-modal but the approximator collapses to a single mode [104]. Notably, mode collapse represents a case where the interpretability of the PAD model may be higher than that of the underlying PD model, at the cost of other utilities, such as predictive performance (see Section 3.3). An example of an P_E model class that produces highly multi-modal posteriors are artificial neural networks [104, 77, 148]. While the interpretability of the underlying P_E model is usually low [38, 324, 323], some of their PAD models can exhibit much higher interpretability if they are steered in the right direction [324, 323].

The above-described notions of interpretability are largely qualitative. While some attempts at a quantitative treatment have been made [73], we are not aware of any sufficiently general definition that allows for a more objective, quantitative comparison between P(AD) models with respect to their interpretability. Thus, we hope that our PAD model taxonomy may inspire more focused research on the quantification of interpretability.

3.8. Convergence

Convergence is a utility of PA and PAD models which rely on complex approximators, such as MCMC, variational inference, or neural density estimators. As explained in Section 2.3, approximators provide certain guarantees under specific assumptions, such as infinite draws in the case of MCMC [108] or infinite training and representational capacity in the case of neural density estimators [249, 245, 269]. In practice, however, modelers cannot wait a lifetime of infinity for approximators' promises to come true; for the time being, we need to work with finite posterior draws and non-convex optimization objectives teeming with local optima.

Thus, our convergence utility pertains to the relative distance between a particular (finitely instantiated) PA(D) model and the optimal PA(D) model attainable under perfect conditions for A. For the above definition to be useful, we need a proxy measure of how close the current approximation is to the optimal approximator outcome. We call such measures *convergence diagnostics* and they are indispensable for ascertaining the validity of PA(D) models. Ideally, good convergence diagnostics should also indicate that the approximation is close to the analytic posterior, but only within the space of distributions the approximator can reach. Accordingly, the relation between convergence and analytic posterior approximation is only indirect for approximators that may be asymptotically biased [319, 70]. Below, we briefly detail common convergence diagnostics for different types of approximators.

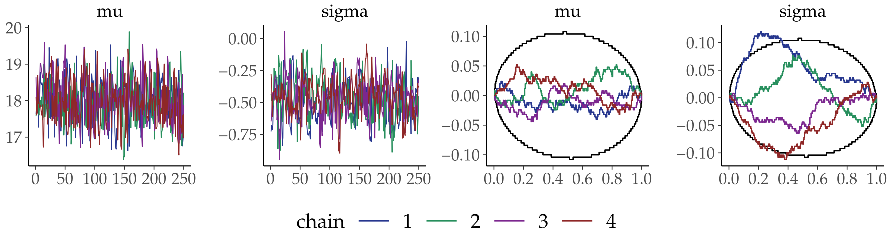

Fig 11. *Graphical convergence checks of two parameters μ and σ. Left: Traditional trace plots. Right: ECDF difference plots with 99%-confidence envelopes [283]. Both kinds of plots use the same posterior draws, but only the rightmost ECDF difference plot highlights that Chains 1 and 4 have some mixing problems for σ. Example draws obtained from the* `bayesplot` *R package [102].*

3.8.1. Convergence diagnostics for Markov chain Monte Carlo

Convergence diagnostics are fundamentally important for posterior approximators that rely on MCMC since these approximators can be arbitrarily bad before full convergence [172]. Thus, all model-based inference relies on the quality of the approximation being close enough to the analytic posterior with respect to some minimally required precision. For a quick graphical check, trace plots or ECDF difference plots [283] can be used, as illustrated in Figure 11.

In terms of numerical approaches, three related classes of MCMC convergence diagnostics are applied in today's practice, namely scale reduction factor $\widehat{R}$, effective sample size (ESS) and Monte Carlo standard error (MCSE) [111, 61, 256, 91, 108, 74, 297]. They all provide convergence measures for univariate quantities of interest $\psi = \psi(\theta)$ that are functions of the P model's parameters θ (see also Section 3.2). There is not a single "global" $\widehat{R}$, ESS, or MCSE measure for ψ, but one for each summary statistic $T(\psi)$ of ψ, where T can be any posterior expectation or quantile [297]. As such, for example, a set of S posterior draws $\psi^{(s)}$ might yield a very precise estimate for the posterior mean of ψ, while at the same time, the estimates of some tail quantiles of ψ (e.g., 5% and 95% quantiles) have much less precision [297]. Accordingly, each of these convergence measures is a function of the quantity of interest ψ and the summary statistic T, computed from the S posterior draws $\psi^{(s)}$.

Broadly speaking, the scale reduction factor $\widehat{R}$ compares the between-chain variance $B = B(f_T(\psi))$ to the within-chain variance $W = W(f_T(\psi))$:

$$\widehat{R}_T(\psi) := \sqrt{\frac{B(f_T(\psi)) + W(f_T(\psi))}{W(f_T(\psi))}}, \tag{51}$$

where the dependence of B and W on T is realized by an appropriate transformation $f_T(\psi)$ that is applied to each posterior draw $\psi^{(s)}$ before the variances are computed, usually on split chains [108, 297, 45]. We can conclude that convergence has been reached if $\widehat{R} \approx 1$, that is, if the within-chain variance dominates the between-chain variance.

The ESS estimates the number of independent draws that contain the same amount of information about $T(\psi)$ as the S dependent posterior draws obtained via an MCMC approximator. As a result, we usually see ESS $< S$, although the opposite can also happen in case of antithetic (negatively auto-correlated) chains [297]. We can obtain the ESS from all autocorrelations $\rho_t = \rho_t(f_T(\psi))$ of lag t of the chains as

$$\mathrm{ESS}_T(\psi) := \frac{S}{1 + 2\sum_{t=1}^{\infty} \rho_t(f_T(\psi))}, \tag{52}$$

where, in practice, we would truncate the infinite sum at some finite value [117]. In modern versions of ESS, ρ_t implicitly depends also on $\widehat{R}$ to take variation across chains into account [108, 297]. In case of independent draws, we have $p_t = 0$ such that ESS $= S$.

The MCSE describes how much (reducible) uncertainty in $T(\psi)$ remains due to the fact that we only have a finite set of dependent MCMC draws for estimation [91, 74, 108, 297]. If T represents an expectation, we can write down the corresponding MCSE schematically as an overall variance $V = V(f_T(\psi))$ across the S draws divided by the corresponding ESS [91, 297]:

$$\mathrm{MCSE}_T(\psi) := \sqrt{\frac{V(f_T(\psi))}{\mathrm{ESS}_T(\psi)}}. \tag{53}$$

MCSE estimates for quantiles need to be computed a little differently and are provided in [297].

Ideally, we should define convergence of MCMC as reaching or undercutting the maximal MCSE that we find minimally acceptable for the given summary of interest $T(\psi)$. However, The MCSE is scale-dependent as it has the same scale as $T(\psi)$, which requires an understanding of how much of an error is acceptable for a certain quantity, in the context of a particular model and research question. This inherently makes MCSE harder to use in practice and hence the scale-free alternatives $\widehat{R}$ and ESS are often preferred [297].

All of the above measures are univariate in the sense that they only concern a univariate T applied to a univariate ψ. Recently, a more comprehensive measure, called R^* [172], has been developed that measures convergence in a multivariate way across multiple model parameters or quantities of interest. It is able to detect non-convergence in the joint posterior that may be overlooked by only investigating convergence of a small, non-exhaustive set of univariate quantities [172]. This is achieved by training an expressive machine learning model (i.e., random forest) to predict chain indices from posterior draws. If the predictive performance of the machine learning model on (unseen) test draws does not exceed chance level, we can assume that the MCMC chains have converged.

In addition to all these sampler-agnostic convergence metrics, there are also few sampler-specific metrics. Most notably, this concerns *divergent transitions* in Hamiltonian Monte-Carlo (HMC) [24], where every occurring divergent transition in the Markov chain may bias the MCMC results and indicate difficulties of the sampler with exploring the target posterior. Divergent transitions tend to occur in regions of high curvature of the explored posterior; regions that most other MCMC samplers struggle to explore as well, only that they fail more silently compared to HMC [24].

3.8.2. Convergence diagnostics for optimization-based algorithms

Many classes of posterior approximators are based on optimization algorithms. The simplest of such approximators aim to find a single point estimate to approximate the analytic posterior, namely the posterior mode, also known as maximum a posteriori (MAP) estimate [108, 189]. Variational inference (VI) approximators also use optimization, but instead of finding the MAP, they aim to find a parametric distribution (e.g., a multivariate Gaussian) that approximates the analytic posterior as closely as possible [94, 252, 30, 310]. The optimization then targets the parameters of this parametric distribution (e.g., the means and standard deviations in Gaussian mean-field VI). Expectation propagation [EP, 221, 207, 298] and integrated Laplace approximation [INLA, 261, 180, 262] work in a conceptually similar fashion, but the structure of their parametric approximators and their target distributions are different (e.g., for INLA, the conditional posteriors of the parameters, instead of their joint posterior). Again, highly similar in terms of their use of optimization, neural approximators (e.g., invertible neural networks; [4, 249]), use optimization to find the neural network parameters that yield the best posterior approximation within the generative scope of the network [230, 185, 126, 229, 249, 269] (but see Section 3.8.5 for specifics in diagnosing convergence of amortized neural approximators).

Regardless of how optimization is applied for posterior approximation, the aim is always to find a single point in a potentially high dimensional space that leads to the best approximation of the analytic posterior within the set of realizable approximations. Accordingly, all traditional convergence criteria for iterative point optimization apply. That is, for non-stochastic optimization algorithms (e.g., gradient-decent or L-BFGS; [217]), small absolute or relative changes in the point estimate, small absolute or relative changes in the target function, or small absolute or relative closeness of the target function's gradient to zero (if the gradient is available) [217, 286], would indicate convergence. For stochastic optimization algorithms (e.g., stochastic gradient-decent or more sophisticated versions, such as Adam; [217, 161]), measuring convergence becomes less straightforward due to the stochasticity in the objective's trajectories. If the step size is held constant, they yield a Markov chain around the target point, once the algorithm comes close enough, instead of converging directly to the target [250, 84]. The latter implies that MCMC convergence diagnostics, in particular $\widehat{R}$, can be applied to diagnose convergence of stochastic optimization algorithms [70].

3.8.3. Convergence diagnostics for sequential Monte Carlo

Sequential Monte Carlo (SMC; aka particle filtering) comprises a heterogeneous class of posterior approximators for PD models whose underlying P models can be expressed in the form of a sequence of conditional distributions (i.e., time series P models) [75, 68]. Most SMC samplers can be shown to provide asymptotically correct inference as the number of draws (particles) approaches infinity [64]. However, empirical convergence diagnostics in the pre-asymptotic regime appear to be relatively scarce still [63, 175, 64]. Perhaps this is because SMC approximators consist of multiple iteratively applied components [64], each with their own pre-asymptotic behavior requiring their own local convergence diagnostics: To assess the convergence of importance sampled (IS) particles at a given step, ESS estimates for weighted samples [169, 326] or variance measures driven by the number of siblings per particle (i.e., the number of particles with the same ancestor node at step zero) [175] can be applied. The trustworthiness of the IS weights themselves could be diagnosed via the Pareto-k-diagnostic of Pareto-smoothed importance sampling (PSIS; [300, 46]), although we are not aware this has been tried so far in the context of SMC (for a closely related application, see [46]). Convergence of MCMC kernels that are part of many SMC algorithms [64] could be assessed via MCMC convergence diagnostics (see Section 3.8.1). While each of these diagnostics may be locally informative for a given SMC component at a given step, whether and how they convey global convergence to the target joint posterior remains to be studied further.

3.8.4. Convergence diagnostics for approximate Bayesian computation

The standard ABC rejection algorithm [259, 71, 285, 243] requires a distance function which quantifies the difference between simulated data y (generated from a P model with a particular parameter configuration θ) and observed data $\tilde{y}$. Further, it needs a tunable tolerance level ϵ according to which the algorithm rejects a fraction of $1-\epsilon$ simulated parameter values. The algorithm then keeps the remaining parameter values as random draws from an approximate posterior $p_\epsilon(\theta \mid y)$.

The ESS of standard ABC rejection samplers is thus typically equal to $(1-\epsilon)\, S$, with S denoting the total simulation budget, since vanilla ABC samplers perform *independent* sampling. However, this does not mean that their sampling efficiency is particularly appealing, especially for high-dimensional P models. That is because ABC samplers notoriously suffer from the curse of dimensionality: Most simulated data sets from a high-dimensional P model will be rejected and so it becomes challenging to obtain enough random draws from $p_\epsilon(\theta \mid y)$ for a reasonable reduction of the MCSE.

More sophisticated ABC algorithms, such as ABC-SMC [275, 165] or ABC-MCMC [194, 289] alleviate some of these issues and inherit the convergence diagnostics of SMC and MCMC. However, whenever hand-crafting of distance functions and summary statistics of the data $H(\tilde{y})$ (i.e., dimensionality reduction) is involved, ABC algorithms can converge *at best* to $p_A(\theta \mid H(\tilde{y}))$. This

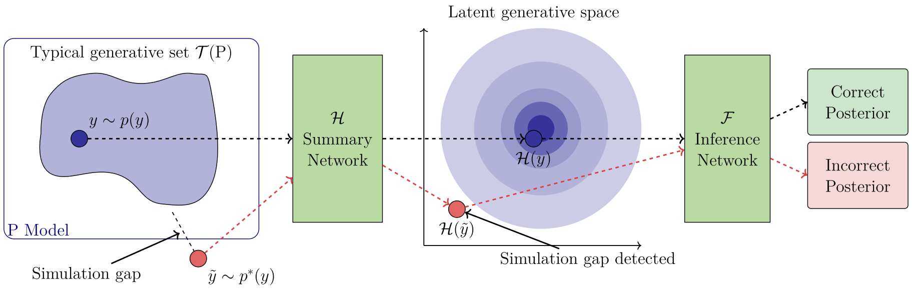

Fig 12. *Detecting model misspecification (i.e., simulation gaps) with amortized neural approximators [269]. A summary (aka embedding) network $\mathcal{H}$ transforms the typical generative set $\mathcal{T}(P)$ of a P model (i.e., the finite set of "in-distribution" data simulations that a P model typically generates) into the typical set of a simple distribution (e.g., multivariate Gaussian). Discrepancies between the model-implied data distribution $p(y)$ and the true data distribution $p^*(y)$ manifest themselves as detectable anomalies, causing potential posterior errors by the inference network $\mathcal{F}$. We can detect these anomalies via standard our-of-distribution (OOD) detection techniques.*

issue does not exist whenever $H(\tilde{y})$ is a sufficient summary statistic, but it can potentially lead to overestimation of $V(f_T(\psi))$ if $H(\tilde{y})$ results in considerable loss of information about the parameters θ.

Recent work on ABC focuses on building robust ABC approximators and exploring the possibility of utilizing hand-crafted summary statistics as a key element of misspecification analysis and error correction [95, 196]. A related line of work suggests comparing posterior moments recovered by differently configured ABC approximators as an empirical diagnostic [96]. It remains an interesting open question whether similar "ensemble approaches" can be generalized to other approximators for simulation-based inference, such as amortized neural surrogates [249, 126, 231] or ABC with learned summary statistics [56, 151].

3.8.5. Convergence diagnostics for amortized approximators

In contrast to MCMC-based approximators, there are no standard convergence diagnostics for amortized approximators yet, as the latter are grounded in diverse, and still fast-evolving theoretical frameworks. However, whenever we employ neural posterior approximators, we can resort to convergence checks used commonly in deep learning applications [124]; see also Section 3.8.2.

Since most modern neural architectures are trained using some form of stochastic gradient-based optimization, optional stopping (i.e., discontinuation of training once the cost function does not improve over some tolerance period) and other convergence heuristics can be used to determine when a neural approximator has reached a stable local minimum of its cost function. That said, due to the nature of non-convex optimization, simply assessing local convergence is not enough for trusting PA(D) models coupled with amortized neural approximators.

Moreover, amortized neural approximators require simulations to be faithful proxies of reality and might yield arbitrarily bad posterior approximations when confronted with data that are atypical under the assumed P model [269]. The latter case is also known as a simulation gap and it occurs when a P model does not accurately represent the behavior of the modeled real-world system (i.e., when the model is misspecified). Consequently, amortized approximators must be able to detect simulation gaps and potential posterior errors, so that they can warn users about suspicious input data and resulting inference.

Generally, there are two broad types of empirical convergence diagnostics we need to utilize in the context of amortized neural approximators: those of a PA model and those of a PAD model. Model-agnostic tools, such as different variants of SBC (see Section 3.2.3), are only applicable to PA models, as they assume that we have access to the actual data-generating parameters. On the other hand, if the neural approximator is a generative neural network [247, 126], the latent space can be used as a source of convergence information for the PA model. For instance, flow-based networks [162, 229] are trained to transform an intractable posterior into a simple base distribution from which random draws can be easily obtained. Thus, convergence of the PA model can generally be determined by a divergence between the prescribed and the learned base distribution.

Unfortunately, neither of the above PA diagnostics can tell us whether the corresponding PAD model will be able to yield faithful estimation due to potential discrepancies between the simulation model and reality (see above). Thus, further diagnostics are necessary to promote the trustworthiness of amortized posteriors. One such diagnostic is the maximum mean discrepancy [MMD, 127] between summary statistics of simulated and real data which tells us whether the observed data belongs to the typical generative set of the simulator or not [269, cf. Figure 12]. As the field of simulation-based inference is still in its infancy, we expect a rapid development of convergence diagnostics for amortized approximators in the future.

3.9. Estimation speed

For non-amortized approximators (cf. Figure 3), we can define *Estimation Speed* as the time from the start of running the approximator A of a PAD model (or a particular instance of a PA model) until convergence, defined by approximator-specific convergence diagnostics (see Section 3.8). In certain cases (see Section 4), it may be sensible to define estimation speed less strictly as the time until termination of the approximator run after which useful results are obtained, without necessarily having achieved convergence. For a given PAD model, estimation speed tends to vary by several orders of magnitude across different classes of approximators. For example, MAP estimators, VI, or other (non-amortized) optimization-based approximators, will usually require a fraction of what sampling-based approximators such as MCMC or SMC need [261, 40]. When considering estimation speed in isolation, there is no doubt that “faster is better”. However, to obtain faster approximators, we often have to give up accuracy or asymptotic guarantees of the resulting posterior approximation [94, 319].

Thus, increasing speed by changing the approximator class may have an adverse effect on other utilities of PA(D) models, specifically on parameter recoverability and predictive performance (see Sections 3.2 and 3.3).

Within a given class of approximators, hyperparameter choices can greatly affect the estimation speed as well [139, 319, 249]. As an example, consider static HMC where the number of leapfrog steps per Markov transition has to be chosen *a priori* [139, 24]. On the one hand, if the number of leapfrog steps is too small, MCMC draws will be highly auto-correlated and thus more draws are required to achieve convergence. On the other hand, if the number of leapfrog steps is too large, a lot of computation time is wasted by unnecessary leapfrog steps; or auto-correlation might even get worse again when the HMC sampler eventually makes a so-called "U-turn" to come back to its starting point [139]. Such problems due to hyperparameter choice can be mitigated by automatically tuning hyperparameters in a "warm-up" phase or adapting them on the fly, conditional on the local geometry of the approximated analytic posterior. For example, when using the No-U-turn sampler [NUTS, 139], a generalization of HMC, the number of leapfrog steps is adaptive. It removes the requirement for the user to choose this hyperparameter manually and may even have better estimation speed than optimally tuned, static HMC [139].

The above definition of estimation speed is straightforward but can be misleading in practice if convergence is not achieved (or unachievable) within a given run of A until its termination, such that A has to be restarted [113]. A typical reason is sub-optimal choices of A's hyperparameters, for example, if the leapfrog step size in HMC is too large leading to divergent transitions whose occurrence implies irrecoverable non-convergence for the current approximator run [24]. Thus, estimation speed in practice may be strongly affected by an approximator's ability to run reliably out of the box without much tuning. Tuning demand can be reduced by adapting hyperparameters automatically on the fly or by having only a small number of sensitive hyperparameters (see Section 3.10). The latter property can further be understood as determining approximator parsimony (see Section 3.6.2).

A manually but skilfully tuned approximator A_1 might beat an auto-tuned approximator A_2 in terms of estimation speed when considering only the final, converging run. However, the overall time (including failed runs) it can take to get A_1 to this optimal state may easily more than offset its final speed advantage. As a result, in an honest comparison of practical estimation time, it may be A_2 that comes out ahead by a substantial margin. Along similar lines, the particular P model implementation may be more or less favourable for different approximators, which can also strongly affect estimation times [24, 297, 16, 66].

Sometimes, the reason for an approximator's termination without convergence may also lie in the computational environment, for example, time or memory constraints on a computing cluster that limit the resources available for a single job. For example, if the estimation of a PA(D) model takes longer than expected, estimation might be terminated prematurely, in the worst case leaving no intermediate result to restart from. In this scenario, even if the second run would then be successful, we still had to deal with at least a doubled estimation time. Accordingly, both the predictability of the expected resource requirements and the small variance in resources between repeated runs of the same PA(D) model can imply substantial practical speed improvements.

3.9.1. Sampling efficiency

For sampling-based approximators, convergence in terms of reaching a given MCSE value (for given quantities of interest and summary statistics), is strongly application-dependent, and so is the estimation speed associated with it (see also Section 3.8.1). For more general comparisons of sampling-based approximators, the concept of *sampling efficiency* is easier to handle and we define it as the average ESS per unit time (for a given quantity of interest ψ and summary statistic T):

$$\mathrm{Eff}_T(\psi) = \frac{\mathrm{ESS}_T(\psi)}{t_{\text{end}} - t_{\text{start}}}, \tag{54}$$

where t_{start} and t_{end} are the start and end times of the approximator run, respectively. While most approximators, even optimization-based ones, *can* be used to obtain posterior draws upon convergence [261, 94], we restrict the class of sampling-based approximators to those that return only posterior draws as their immediate endpoint instead of the parameters of (closed-form) density functions. MCMC, SMC, and rejection sampling are the most important members of this class [259, 108, 64].

Within a class of sampling-based approximators, say MCMC, the same convergence diagnostics, in particular the same ESS, can be applied to all competitors, which simplifies comparisons [16, 66]. Here it is important to not only use the same *implementation* for these diagnostics across all approximators, but also to ensure that this implementation follows the current state-of-the-art of diagnostic development [295]. Otherwise, comparisons may be biased by outdated diagnostics. Additionally, one needs to verify empirically that the obtained stationary distribution for a given PAD model is the same for all the compared approximators. Otherwise, sampling efficiency will be misleading since at least one approximator would not have estimated the analytic posterior of the underlying PD model well enough. Comparisons between approximators belonging to different sampling-based classes may require even more care to ensure that ESS diagnostics across classes are comparable, for example, when comparing MCMC with SMC approximators.

Whenever we are performing sampling efficiency comparisons for PA instead of PAD models, we not only have, in principle, an infinite number of data sets as a basis for comparison but can utilize SBC to falsify the correctness of the achieved stationary distributions (see Section 3.2.3). However, when we investigate sampling efficiency on a set of simulated data sets, different data-generation scenarios should be studied, since well-specified P models may have different efficiency than misspecified P models.

When studying sampling efficiency, the same practical caveats apply as for estimation speed in general. For instance, to achieve a comparison that is ecologically valid for real-world situations, we have to investigate practical sampling efficiency that considers both optimized and non-optimized P model implementations, as well as failed or prematurely terminated approximator runs.

3.9.2. Estimation speed of amortized approximators

Amortized approximators, such as the pre-paid estimation method [204] or neural density estimation methods [247, 126], require a slightly modified view on estimation speed, since they tend to split inference into two phases (cf. Figure 2). Convergence in the context of amortized neural approximators typically happens before any posterior draws have been obtained [303, 123], so the primary computational load falls into the upfront simulation-based training phase. In contrast, the computational cost of applying a pre-trained amortized approximator to obtain thousands of posterior draws or perform density estimation on real data is typically negligible and only a matter of seconds even for high-dimensional posteriors [247]. In this way, amortized approximators can be extremely useful for studying the (global) information gain (see Section 3.2.1) or calibration (see Section 3.2.3) as part of the parameter recoverability utility of PA models, since these demand inference on many, potentially thousands of data sets simulated from the underlying P model.

Due to the properties of amortized approximators, we can modify the definition of estimation speed as the time until convergence of the training phase plus the time for obtaining a sufficient number of posterior draws on real data to reduce the MCSE beyond a pre-defined threshold. In this context, estimation speed will greatly depend on the *simulation time*, that is, the computational cost of performing a sufficient number of model simulations. Some amortized methods make it possible to further subdivide estimation speed into three parts: simulation time, training time, and inference time,[6] for instance, when using BayesFlow in an offline training regime [249] or when applying sequential neural estimators [126] with the prior as a sole proposal distribution throughout training.

In any case, estimation speed for amortized approximators will be dominated by the time spent before obtaining posterior draws. Accordingly, it is often important to determine the break-even point between an amortized and a non-amortized method, that is, after how many observed data sets does the training effort amortize in terms of ESS per unit time? Naturally, this break-even point will heavily depend on the modeling context. For some P models, the break-even point between neural estimation and ABC can occur after as few as 5 or even fewer observed data sets [245], but it can also occur only after as many as dozens when comparing different neural approximators [247]. In addition, amortized neural samplers can often yield independent posterior draws upon convergence [247, 123, 126], so their sampling efficiency during the inference

[6] Other amortized approximators, such as the pre-paid method [204], only entail a simulation phase and an inference phase.

phase (cf. Figure 2) will be generally superior to non-amortized approximators (i.e., stateful samplers) yielding dependent draws.

Importantly, comparisons between amortized and non-amortized approximators (but also comparisons within the same A class) should take implementation factors into account. For instance, the estimation speed of neural approximators will be greatly enhanced by using GPU parallelization and even standard ABC rejection samplers can be quite efficient when run on a computing cluster with hundreds of nodes [165]. For simulation-based inference, the implementation of the simulation model presents a further potential bottleneck, which can be alleviated via parallelization, model reformulation reducing the algorithmic complexity of the simulator, or calibration of large-scale simulators via simpler surrogates [186]. In addition, recent hybrid methods employ a mixture of amortized and non-amortized components, such as amortized likelihood ratio approximators within non-amortized MCMC [136] or neural likelihood surrogates [231]. These hybrid methods blur the distinction between amortized and non-amortized methods and render the definition of estimation speed even more challenging. The dependence on these various implementation factors should make us wary of comparisons between approximators in terms of estimation speed and appreciate the challenges of building scalable PA(D) models.

3.10. Robustness

A common question that arises when we discuss substantive conclusions derived from model-based inference is how fragile these conclusions are with respect to crucial aspects of P, A, or D. Can we "break" the analysis by a barely perceptible change in the data or by using a slightly different approximator? Or are the main results of the analysis largely impervious to such seemingly unsubstantial changes? The *Robustness* utility attempts to answer such questions by measuring how much a PA, PD, or PAD model's implications change as we (systematically) perturb some of its components.

In the above definition, we use the term "component" very generally. It can refer to (structural) aspects of the P model, most notably, to priors or their hyperparameters [69, 154, 244, 17, 86, 87, 257] or to aspects of the likelihood function [17, 34]. It can also refer to the choice of data D, for example, the percentage of left-out observations [154, 219, 300, 296], or to hyperparameters of the posterior approximator A [139, 249, 75, 319]. Thus, the term essentially refers to any aspect in which a P(A)(D) model can be sensibly modified.

More formally, we want to investigate the sensitivity (inverse robustness) of the posterior of some quantity $\psi = \psi(\alpha)$ with respect to some variable α which exerts a potential influence on a component of interest. If we are only interested in investigating the sensitivity of a specific (point) summary of the posterior, we convey this by writing $T(\psi(\alpha))$ for an arbitrary summary statistic T.

For example, we can study likelihood or prior sensitivity by power-scaling the respective components of the P model, that is, replacing the joint model $p(y \mid \theta)\, p(\theta)$ with $p(y \mid \theta)^{\alpha}\, p(\theta)$ or $p(y \mid \theta)\, p(\theta)^{\alpha}$, respectively [219, 144, 28, 154].

Of course, one can also choose to power-scale only parts of the likelihood or parts of the joint prior. Although it is just one of many ways to systematically perturb a P model, power-scaling is a popular approach due to its simplicity and natural integration with existing workflows [28, 154].

Regardless of the exact perturbation method, we can define (local) sensitivity as a measurable distance (represented by a function f) between the results of the current P(A)(D) model at value α_0 and an alternative value α_1 that implies a different P(A)(D) model, diverging from the original one only in the choice of α [154, 257]:

$$\mathrm{Sen}_\alpha(T(\psi), \alpha_0, \alpha_1) := f(T(\psi(\alpha_0)), T(\psi(\alpha_1))). \tag{55}$$

For example, if T were a posterior expectation or a quantile of some univariate quantity ψ and α were a hyperparameter of the prior $p(\psi) = p(\psi \mid \alpha)$, then f could simply be the absolute difference between these expectations or quantiles as implied by α_0 and α_1, respectively. This definition can further be generalized to sets of alternative α values in the neighborhood of α_0 [257].

If α can be perturbed continuously (e.g., using power-scaling) and if $T(\psi(\alpha))$ is differentiable at α_0, we may also define sensitivity as a function of the derivative of $T(\psi(\alpha))$ evaluated at α_0 [154, 244]:

$$\mathrm{Sen}_\alpha(\nabla T(\psi), \alpha_0) := f\left(\left. \frac{d\,T(\psi(\alpha))}{d\,\alpha} \right|_{\alpha=\alpha_0} \right). \tag{56}$$

The latter definition has the advantage that no further value α_1 has to be chosen, but it has a smaller range of applicability and potentially more difficult interpretation. In both of the above definitions, we can always choose f such that the sensitivity is non-negative with a value of 0 indicating complete insensitivity.

For complex models, small amounts of sensitivity are almost always expected but may not practically matter. Accordingly, we would say that $T(\psi(\alpha)$ is *practically sensitive* with respect to α if

$$\mathrm{Sen}_\alpha(T(\psi), \alpha_0, \alpha_1) > \delta \qquad (\text{or } \mathrm{Sen}_\alpha(\nabla T(\psi), \alpha_0) > \delta) \tag{57}$$

for some chosen threshold δ that depends on the sensitivity measure and the modeling context [154]. For example, let T denote a posterior mean and ψ denote a standardized effect size that we would deem sensitive if a change exceeds $\delta = 0.2$ standard deviation units. Then, the results would be practically sensitive to the given perturbation, if changing α from α_0 to α_1 implied $|T(\psi(\alpha_0)) - T(\psi(\alpha_1))| > 0.2$.

When evaluating practical sensitivity related to hyperparameter choice within a class of posterior approximators, robustness is highly desirable, since approximators should ideally converge to the same target (see Section 3.8). What is more, as PA models grow in complexity, analyses based on a single approximator may gain trustworthiness by some form of multiverse analysis employing multiple approximators [305]. However, it is currently unclear how to systematically weigh the relative contribution of different approximators when trying to aggregate results from multiverse analysis, since some approximators might yield very poor posterior approximations and thus skew any substantial conclusion.

Differently, when it comes to perturbations in P model assumptions or the observed data, neither practical sensitivity nor insensitivity is desirable *per se*. Rather, we would like results to be practically robust to perturbations if (a) the perturbations affect only nuisance components of the P(A)D models that are equally justifiable within the given context, or (b) if the perturbations are so small that they could very well have occurred due to uncontrolled or uncontrollable influences. In contrast, when different P model assumptions represent competing substantive theories of interest, we want the corresponding P(A)D models to be sensitive to these assumptions.

Examples for (a) include different choices of non-equivalent likelihood families that have an overall similar complexity (e.g., Log-Normal vs. Gamma distribution for continuous positive data) or different P(A)D models that are capable of similar predictive performance (see Section 3.3), in case the latter is not already the quantity of interest itself. Examples for (b) include adding small amounts of noise to the data [192], leaving out a small subset of the data [300, 296], or slightly changing prior hyperparameters when the goal is to specify weakly informative priors [154]. As the magnitude of the perturbations increases, we expect results to become practically sensitive to these perturbations and observed insensitivity would then be a reason for concern. For example, if drastically increasing the amount of data would not reduce the posterior standard deviation of ψ, this would be an indication of empirical non-identifiability ([112]; see also Section 3.2.1) or an error in our model code [113].

Another type of sensitivity, arising in modeling dynamic systems, is sensitivity to initial conditions [182, 271], which in our taxonomy can be understood as part of P_I models. Sensitivity to initial conditions, popularly known as the *butterfly effect*, implies that an arbitrarily small change in initial conditions can result in considerably different subsequent system states (or observed trajectories). Moreover, this type of sensitivity can be considered as a hallmark of deterministic chaos [271]. In the context of dynamic models, the so-called Lyapunov exponent can measure a model's sensitivity to initial conditions [14]. Lyapunov exponents characterize the rate of exponential divergence from perturbed initial conditions and the maximal Lyapunov exponent can be used to summarize the overall sensitivity of a model into a single number [155]. For more details on dynamic systems and deterministic chaos, we refer the interested readers to [271, 32].

3.11. Intermediate summary II

In the preceding sections, we have proposed to focus on ten general utility dimensions pertaining to the different Bayesian model classes within our PAD model taxonomy. Table 3 summarizes the applicability of each utility dimension to each model class; the justification for this classification can be gathered

Table 3
Applicability of the ten utility dimensions for each model class of the PAD-taxonomy.

	P	PA	PD	PAD
Causal Consistency	✓	✓	✓	✓
Parameter Recoverability	✓	✓	✗	✗
Predictive Performance	✗	✗	✓	✓
Fairness	✗	✗	✓	✓
Structural Faithfulness	✓	✓	✓	✓
Parsimony	✓	✓	✓	✓
Interpretability	✓	✓	✓	✓
Convergence	✗	✓	✗	✓
Estimation Speed	✗	✓	✗	✓
Robustness	✗	✗	✓	✓

from the treatment of individual utilities. Throughout, we have tried to elucidate the rather diverse facets of these utilities compactly, while providing a comprehensive list of references for the interested readers. We believe that considerations regarding these utilities are already *implicit* in much of the applied Bayesian literature. However, besides disambiguation of core concepts, a major goal of collecting these utilities has been to stimulate their *explicit* consideration in Bayesian workflows. In the following section, we will highlight the interactions between some of these utilities and discuss their relative importance in terms of application-dependent utility hierarchies.

4. Utility hierarchies and trade-offs

For comprehensive Bayesian model comparison across all utility dimensions, we need a thorough understanding of how the latter relate to each other. Thus, we begin by highlighting important connections between selected utilities. Some connections have been discussed already in different places in Section 3, but we summarize some of them here again for the reader's convenience.

4.1. Interplay between utilities

Predictive performance and parameter recoverability Any modeling goal can be summarized by a set of quantities whose inferred model-based approximations are then used in subsequent decision-making. These quantities of interest may be manifest (observable) or latent (unobservable), leading either to a focus on predictive performance (observable quantities) or parameter recoverability (latent quantities). Statistically, these two utilities can be evaluated similarly by comparing model-based approximations with their real-world or ground-truth counterparts. However, despite the statistical similarity in evaluation, prioritizing either of the two leads to substantially different modeling workflows, as detailed further below.

Parameter recoverability and convergence Both parameter recoverability and convergence aim to quantify the difference between model-based results and some ideal theoretical target that we want to approximate as best as possible. However, the two utilities differ in what the *ideal target* is. For parameter recoverability, it is a known ground-truth from which the data D was implicitly or explicitly generated. For convergence, it is the best possible posterior approximation that a particular approximator A can achieve for a given PD model. Indeed, these utilities are different for two main reasons. First, even the analytic posterior of a PD model may perform very poorly in terms of parameter recovery, for example, when D contains too little information relative to the complexity of P. Second, for biased approximators, the ideal convergence target is not even the analytic posterior itself but rather the "closest possible" distribution within the scope of A.

Convergence and estimation speed In most cases, convergence determines the definition of estimation speed by defining the latter as the time from the start of running the approximator to convergence. While this definition usually works well, there are some scenarios where it falls short. First, sometimes we may want to define estimation speed less strictly as the time until termination of the approximator run after which useful results can be obtained. This definition can be sensible especially when the goal is to achieve good predictive performance. If that goal was already achieved with a formally non-convergent model, there may be no need to bother with convergence anymore. Second, when using amortized approximators, we split the approximation process into a time-intensive training step and a subsequent inference step, which is then almost instant. In this context, convergence can be easily defined only for the training step, while estimation speed has clear definitions both for the training and inference step.

Causal consistency and fairness Fairness is not detached from causal considerations. For example, measurement fairness aims to ensure that the items in a test battery measuring a psychometric construct relate to this construct in the same way for all groups differing in protected attributes. In the associated causal graph, the causal effect of the latent construct on the measured items should be conditionally independent of the protected attributes given the unprotected attributes. However, fairness considerations reach beyond causality, as they carry important ethical, political, and societal aspects that causal modeling alone cannot appropriately account for (i.e., questions of fact vs. questions of value).

Causal consistency and structural faithfulness Both causality and structural faithfulness aim to represent an empirical phenomenon or the characteristics and constraints of an assumed stochastic process as closely as possible via an adequate P model. However, they differ in what aspects of the process they try to encompass. Causal consistency focuses on the specification of conditional independence between (sets of) variables given (sets of) other variables.

This structure is then reflected in the P model's probabilistic factorization without referring to specific distributions or functional forms. Differently, structural faithfulness deals with the details of the P model itself, for example, what functional forms connect the conditionally dependent variables or which probability distribution they follow.

Structural faithfulness and parsimony Changing structural faithfulness affects parsimony although in different directions depending on the kind of P models being considered and how structural faithfulness is increased. For example, when adding physical constraints such as symmetries, parsimony is increased along with structural faithfulness as the model does not have to learn the constraints from data. In contrast, when modeling additional probabilistic structures (e.g., of time and space), parsimony is usually decreased as new parameters have to be added to the P model to account for these structures.

Parsimony and interpretability Higher parsimony is often associated with higher interpretability. However, there are notable expectations. For example, a highly regularized P model (e.g., through continuous shrinkage priors) may be considered highly parsimonious in terms of the effective number of parameters, but at the same time remain largely opaque because the parameter dimensionality itself remains high. As another example, low dimensional yet highly non-linear systems may feature directly interpretable parameters (e.g., growth rate in logistic map models), but their joint influence on the system's behavior may be very hard to understand without the aid of model simulations.

4.2. Utility trees

Having highlighted some key connections between utilities, we will now move on to evaluating the utilities' relative importance depending on the goal of inference. We will differentiate between a utility *hierarchy*, where one utility is strictly more important than another, and a utility *trade-off*, where we can achieve a gain of one utility at the cost of a loss of another. Utility hierarchies, utility trade-offs, as well as the relative importance of different utilities, are inevitably application-specific and contingent on the particular modeling goals.

The first branching point in ranking the different utilities is whether we are interested in observable or latent quantities for subsequent decision-making, leading to what we call observable and latent inferential goals. This distinction is equivalent to focusing on either predictive performance or parameter recoverability as a primary utility. Related binary perspectives have been put forward, for example, as the difference between two "statistical cultures" [37] or the prediction-explanation dilemma [320, 273]. However, the distinction between the two goals has not been discussed in the context of explicit model utilities. Below, we present two utility trees defining hierarchies and trade-offs for the two kinds of inferential goals.

The way we would like to see these utility trees being used in practice is that analysts (a) build and improve their models in a way that respects utility

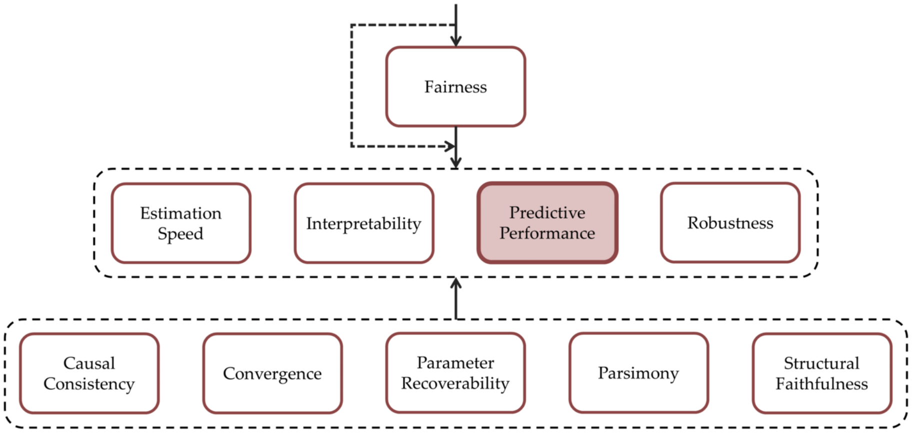

Fig 13. *Utility tree for model-based inference of observable quantities (prediction).*

hierarchies and (b) talk explicitly about the utility trade-offs they have been making in the process. This should enable users of the models and consumers of statistical inference to understand which model-building decisions have been made, why they have been made, and how they affect the trustworthiness of model-based decisions.

4.3. Utility tree for observable inferential goals

The workflow for observable inferential goals centers around the predictive performance of PAD models as a central utility. This is the prevalent perspective on modeling in machine learning research. Given its practical nature, this perspective would require a practically usable representation of a posterior distribution from which predictions can subsequently be obtained, hence the focus on PAD models. Below, we discuss our proposed model utility tree for observable inferential goals (see Figure 13).

4.3.1. Primary utilities

Fairness The fact that predictive performance is the central utility under this perspective does not mean that it would be the sole or even the most important utility to consider. Rather, on the top of the utility hierarchy, we need to check whether the predictive goals concern certain aspects related to fairness. If they do, our PAD model needs to satisfy the fairness criteria agreed upon in the corresponding domain, otherwise, it would be considered invalid from an ethical and/or legal perspective (see Section 3.4), regardless of how good its predictive performance is. If fairness concerns do not apply to the particular PAD model, the fairness utility can be circumvented.

4.3.2. Secondary utilities

The secondary level of our predictive utility tree includes (in addition to predictive performance itself), estimation speed, interpretability, and robustness (in alphabetical order) as utilities across which trade-offs can be made. Notably, we do not require convergence of the PAD here and view estimation speed simply until termination of the approximator, regardless of whether or not convergence had been reached (see Section 3.9). While the three additional central utilities may exhibit trade-offs among each other, increasing them may in particular justify (some) reduction in predictive performance.

Estimation speed If achieving high predictive performance requires either a very high dimensional parameter space (e.g., as in a neural network) or the repeated evaluation of a complex simulator (e.g., in a differential equation-based mechanistic P_I model), then estimation speed will exhibit a trade-off with predictive performance. In other words, PAD models with easy-to-evaluate or easy-to-simulate likelihoods may obtain faster posterior approximation, but may also yield worse predictive performance. Whenever estimation speed becomes prohibitive for the practical application of a PAD model, the context may justify the use of another P model, even if the latter sacrifices (some) predictive performance. The same logic can justify using approximators that speed up the approximation of a PAD model, even at the expense of losing (some) predictive performance (e.g., using VI instead of MCMC-based approximators; see [29] or Section 2.3).

Interpretability Interpretability is often higher in P models with lower parameter dimensionality and linear structure (see Section 3.7). However, depending on the complexity of the real data generator, low-dimensional and/or linear(ish) models may have worse predictive performance than higher-dimensional and/or more non-linear models. Yet, even if predictive performance is the main goal, it may still be legitimate (or even legally required) to use a more interpretable PAD model, even at the cost of some predictive performance.

Robustness A PAD model yielding high predictive performance on some test data may yield surprisingly poor predictions on slightly modified data (e.g., adversarial attacks on deep neural networks, see [2]). In a similar vein, a PAD model that is well predicting given some reasonable initial values of an approximator A may deliver worse predictions for some other (equally reasonable) initial values [177]. Both of these sensitivity types are not desirable and it can be legitimate to sacrifice (some) predictive performance for an increase in robustness against small perturbations.

4.3.3. Tertiary utilities

At the third level of the predictive performance tree, we find supporting utilities that may serve as proxies for the central utilities. Specifically, these include causal consistency, convergence, parameter recoverability, parsimony, and structural faithfulness (in alphabetical order). Tertiary utilities are often easier to evaluate and available “early” in the model-building workflow, for example, when they are only requiring a P model instead of a PAD model. Using these utilities can thus help speed up the model-building process. However, these supporting utilities should only guide final modeling choices whenever multiple models are equally justifiable with respect to primary and secondary utilities.

Causal consistency If the quantities of interest are purely predictive, enforcing a P model to satisfy causal consistency (or even thinking about a causal graph in the first place) is not required and may even have detrimental effects on predictions [202]. In other words, for predictions, it would usually be entirely irrelevant how the association between two variables came to happen, as long as the input variables are predictive of the outcome variables. However, causal consistency can still be a supporting utility to reduce the *a priori* admissible model space by ruling out variables or interactions, as well as related P model terms, for which a causal graph implies a lack of relation to the target variables (i.e., no path between covariates and target, regardless of path direction). Considering a genetic association study as an example, we can rule out gene areas that only encode genes whose effects are known and understood to have no plausible relationship to the phenotypes being predicted [306].

Convergence Convergence is not strictly required in the predictive utility tree, since a non-converged PAD model may still exhibit satisfactory predictive performance. That said, achieving convergence will likely imply an improvement in central utilities as well. Not only is this true for predictive performance itself, but also for robustness. For instance, a non-converged PAD model may vary arbitrarily from another non-converged PAD model, whereas we can expect them to be (more) similar upon convergence, at least for approximators that have the potential to explore the full posterior (see Section 2.3). Accordingly, before studying central utilities that may be costly to evaluate, we can use convergence as a shortcut to rule out PAD models with low potential to score high in those central utilities.

Parameter recoverability Parameter recoverability can indirectly enhance predictive performance since nearly non-identifiable P models and poorly calibrated PA models can be discarded early in a model-building workflow. Such models can neither yield good predictions, nor trustworthy uncertainty representation, as some information gain is necessary to achieve posterior predictions that are different from prior predictions (see Section 3.2.1). However, strict parameter recoverability still plays a secondary role for the central goal of achieving good predictions. For instance, highly over-parameterized P models may achieve zero Bayesian surprise (i.e., no difference between prior and posterior) for a large fraction of their parameters, but still yield reasonable predictions based on the few identifiable parameters [246].

Parsimony In addition to having aesthetical value in itself (see Section 3.6), parsimony as a supporting utility bears close relations to estimation speed, interpretability and predictive performance: More parsimonious models tend to be (a) faster to estimate, at least when comparing P models that are nested (i.e., one model is a special case of the other), (b) more interpretable, as fewer parameters have to be considered simultaneously, and (c) less prone to overfitting (although they may be prone to underfitting). Some forms of parsimony (both plain number of parameters and *a priori* effective number of parameters; see Section 3.6.1) are available before running any approximator. Correspondingly, we can utilize parsimony as an *a priori* proxy for the central utilities.

Structural faithfulness Structural faithfulness comprises several P model characteristics that we ideally know and understand before running any approximator: variable scales, probabilistic symmetries, and physical constraints (see Section 3.5). Structural faithfulness is related to multiple central utilities. Most importantly this concerns predictive performance, as structurally faithful models are more likely to predict more accurately while requiring less data and showing better uncertainty calibration [251, 312]. But structural faithfulness is also related to estimation speed (for the better or worse; [110, 40]) as well as robustness, for example, small perturbations of the training data [192]. Accordingly, we can also treat structural faithfulness as a proxy for central utilities to reduce the *a priori* considered model space to (sufficiently) structurally faithful models.

4.4. Utility tree for latent inferential goals

The utility tree for latent inferential goals centers around the parameter recoverability of P or PD models as the central utility. Modeling goals following this perspective are almost entirely of theoretical, epistemic nature, and so the approximator is not itself part of the modeling goal. Yet, in practice, we will still almost always rely on PA and PAD models for practical evaluation, hence the indispensable role of the approximator. Below, we discuss our proposed model utility tree for latent inferential goals (see Figure 14).

4.4.1. Primary utilities

Fairness Once again, on top of the hierarchy, we find fairness for ethical and/or legal reasons whenever the modeling context has fairness-related implications (see Section 3.4). First, at an individual (person-specific) level, we need to ensure that estimated latent parameters are fair with regard to individuals of protected groups. Second, at a more general (person-unspecific) level, we need to keep in mind how our inferences about latent parameters might trigger political decisions or societal processes affecting protected groups.

Causal consistency Next, we need to ensure that the P model is causally consistent with the assumed, and theoretically justified, causal graph (see Section 3.1). We argue that thinking about causal consistency is required for any

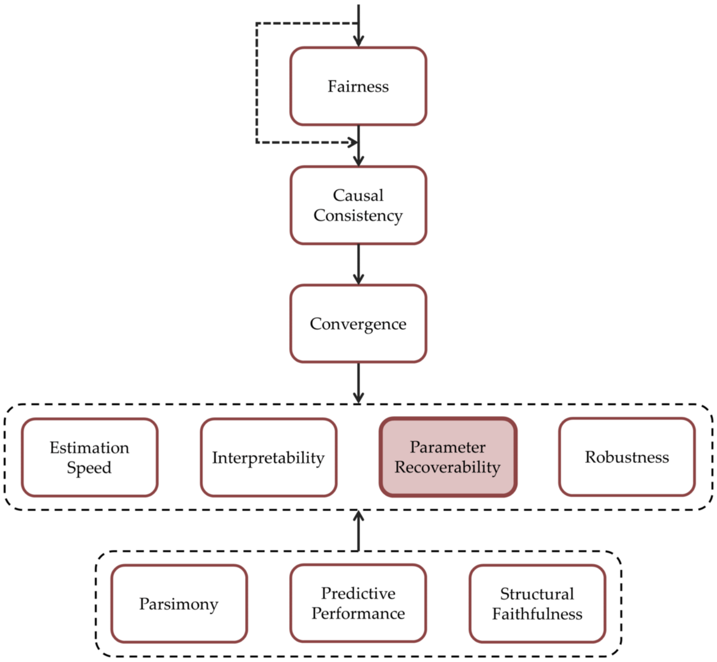

FIG 14. *Utility tree for model-based inference of latent quantities (e.g., parameter estimation).*

latent modeling goal. Even if studies do not engage in (sufficient) causal analysis, and then correctly state that their results cannot be interpreted causally, there remains the (perhaps implicit) wish that a causal claim would be possible. What is more, even a pure measurement goal (e.g., estimating intelligence or personality traits without the need to relate latent parameters to each other) would need a causal model to decide and justify which observable variables to use for the estimation of the latent variables [156]. Finally, even if one might find a latent inferential goal that would be honestly satisfied with association only, causal analysis and discussion would still be required to prevent people from interpreting results causally.

Convergence Convergence of PAD models is a prerequisite for any practically trustworthy result of a latent inferential goal because we have no external validation criterion available during inference on real data as we would have when considering observable inferential goals. In fact, before convergence, posterior approximations may be almost arbitrarily incorrect, regardless of the kind of approximator being used (see Section 3.8). Specifically, for asymptotically biased approximators (e.g., VI), the approximated posterior upon convergence may still be a bad representation of the analytic posterior if the expressive scope of the approximator is limited. But even in such cases, a converged approximator is more likely to be closer to the analytic posterior than an arbitrary, non-converged approximator, and so the former is to be treated as more trustworthy.

4.4.2. Secondary utilities

When it comes to inferential goals, parameter recoverability is the central utility of the secondary hierarchy. However, except for the identification sub-utility, it cannot be studied directly on real data because knowledge of the ground-truth is missing (see Section 3.2). As a result, many studies on parameter recoverability occur in the form of simulations or, if possible, mathematical analysis. This also concerns studying trade-offs with other central utilities, namely, estimation speed, interpretability, and robustness. These remain instrumentally the same as for observable inferential goals and can reveal trade-offs with parameter recoverability for the same reason as for predictive performance (see Section 4.3.2).

4.4.3. Tertiary utilities

Due to the lack of ground-truth latent parameters at real-data inference time, the tertiary, supporting utilities not only aim at speeding up model building but may also function as observable proxies of parameter recoverability. These supporting utilities are parsimony, structural faithfulness, and predictive performance. The reason for the relevance of the former two is the same as for observable inferential goals (see Section 4.3.3), and so only predictive performance requires separate explanation and justification.

Predictive performance The relation between predictive performance and parameter recoverability is complicated and using the former as an observable proxy for the latter in a valid way requires great care [273, 270]. Most importantly, we should not choose causally inconsistent P models, even if they predict better [202, 270]. Fortunately, when taking the here-presented hierarchy of utilities seriously, this danger is banished by giving causal consistency priority over almost all other utilities for latent inferential goals. Within the class of causally consistent P models, it seems that using predictive performance as a proxy for parameter recoverability in (converged) PAD models represents a valid approach [270]. For example, we can utilize predictive performance to determine whether an extra probabilistic structure (e.g., accounting for potential temporal or spatial dependencies; see Section 3.5.2) is worth including or not [39]. This affects the balance between structural faithfulness and parsimony, which in turn serve as proxies for parameter recoverability. As another example, the choice of likelihood functions driven by predictive performance can be used to improve parameter recoverability in the context of regression P models [270].

5. Conclusion

We proposed answers to two fundamental questions of Bayesian modeling, namely (1) “What actually *is* a Bayesian model” and (2) “What makes a *good*

Bayesian model"? Ultimately, we hope that both of these questions and the answers we provided will aid in thinking and talking about Bayesian models, as well as enhance the overarching model-building process, regardless of the specific methods and fields of application.

As an answer to the first question (Section 2), we proposed the PAD model taxonomy that defines four different kinds of Bayesian models as subsets of the triple of joint distribution of all involved variables (P), the training data (D), and the posterior approximator (A). In this way, we put forward our view that modern Bayesian models are more than just likelihood and prior, but comprise a variety of "external components" that influence, and, in turn, are influenced by, the goals and the results of any statistical analysis.

As an answer to the second question (Sections 3 and 4), we first argued that there are ten utility dimensions along which we can evaluate Bayesian models, namely, (1) causal consistency, (2) parameter recoverability, (3) predictive performance, (4) fairness, (5) structural faithfulness, (6) parsimony, (7) interpretability, (8) convergence, (9) estimation speed, and (10) robustness. Then, we proposed two utility trees that embody utility hierarchies and trade-offs depending on the particular inferential goals. We hope that our list of utility dimensions and structure of possible inferential goals is exhaustive (up to using synonyms and regrouping sub-utilities differently). However, it may as well become incomplete in the future, as new ideas are born and rapidly developed, and we will be happy to incorporate these into our taxonomy.

Acknowledgments

We thank Rudolf Debelak for helpful feedback on earlier versions of this paper. Our work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2075 - 390740016 (the Stuttgart Cluster of Excellence SimTech) and EXC-2181 - 390900948 (the Heidelberg Cluster of Excellence STRUCTURES). The authors gratefully acknowledge the support and funding.

References

[1] Aguilar, J. E. and Bürkner, P.-C. (2023). Intuitive joint priors for Bayesian linear multilevel models: The R2D2M2 prior. *Electronic Journal of Statistics* **17** 1711–1767. Publisher: Institute of Mathematical Statistics and Bernoulli Society. https://doi.org/10.1214/23-EJS2136. MR4609453

[2] Akhtar, N. and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **6** 14410–14430.

[3] Ardizzone, L., Kruse, J., Lüth, C., Bracher, N., Rother, C. and Köthe, U. (2021). Conditional invertible neural networks for diverse image-to-image translation. In *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42* 373–387. Springer.

[4] Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C. and Köthe, U. (2018). Analyzing inverse problems with invertible neural networks. *arXiv preprint*.

[5] Association, A. E. R., ed. (2011). *Standards for Educational and Psychological Testing*. American Educational Research Association, Washington, D.C. OCLC: ocn826867074.

[6] Avecilla, G., Chuong, J. N., Li, F., Sherlock, G., Gresham, D. and Ram, Y. (2022). Neural networks enable efficient and accurate simulation-based inference of evolutionary parameters from adaptation dynamics. *PLoS Biology* **20** e3001633.

[7] Baddoo, P. J., Herrmann, B., McKeon, B. J., Kutz, J. N. and Brunton, S. L. (2021). Physics-informed dynamic mode decomposition (piDMD). *arXiv preprint*. MR4574819

[8] Bak, M. A. (2022). Computing fairness: ethics of modeling and simulation in public health. *Simulation* **98** 103–111.

[9] Barocas, S., Hardt, M. and Narayanan, A. (2019). *Fairness and Machine Learning*. fairmlbook.org `http://www.fairmlbook.org`.

[10] Barrientos, P. G., Rodríguez, J. Á. and Ruiz-Herrera, A. (2017). Chaotic dynamics in the seasonally forced SIR epidemic model. *Journal of Mathematical Biology* **75** 1655–1668. MR3712325

[11] Bates, D., Kliegl, R., Vasishth, S. and Baayen, H. (2015). Parsimonious mixed models. *arXiv preprint*.

[12] Bates, D., Mächler, M., Bolker, B. and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* **67** 1–48.

[13] Beck, C. and Cohen, E. G. D. (2003). Superstatistics. *Physica A: Statistical Mechanics and its Applications* **322** 267–275. `https://doi.org/10.1016/S0378-4371(03)00019-0`. MR1980540

[14] Benettin, G., Galgani, L., Giorgilli, A. and Strelcyn, J.-M. (1980). Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. Part 1: Theory. *Meccanica* **15** 9–20.

[15] Bennett, C. H. (1976). Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics* **22** 245–268. MR0471852

[16] Beraha, M., Falco, D. and Guglielmi, A. (2021). JAGS, NIMBLE, Stan: A detailed comparison among Bayesian MCMC software. *arXiv preprint*.

[17] Berger, J. O., Moreno, E., Pericchi, L. R., Bayarri, M. J., Bernardo, J. M., Cano, J. A., De la Horra, J., Martín, J., Ríos-Insúa, D., Betrò, B., Dasgupta, A., Gustafson, P., Wasserman, L., Kadane, J. B., Srinivasan, C., Lavine, M., O'Hagan, A., Polasek, W., Robert, C. P., Goutis, C., Ruggeri, F., Salinetti, G. and Sivaganesan, S. (1994). An overview of robust Bayesian analysis. *Test* **3** 5–124. `https://doi.org/10.1007/`

BF02562676. MR3420903

[18] Berger, J. O. and Pericchi, L. R. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association* **91** 109–122. MR1394065

[19] Berk, R., Heidari, H., Jabbari, S., Kearns, M. and Roth, A. (2021). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research* **50** 3–44. https://doi.org/10.1177/0049124118782533. MR4198551

[20] Berliner, L. M. (1991). Likelihood and Bayesian prediction of chaotic systems. *Journal of the American Statistical Association* **86** 938–952. MR1146342

[21] Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. Hoboken: Wiley. MR1274699

[22] Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)* **36** 192–225. https://doi.org/10.1111/j.2517-6161.1974.tb00999.x MR0373208

[23] Best, N., Dallow, N. and Montague, T. (2020). Prior elicitation. *Bayesian Methods in Pharmaceutical Research* 87–109.

[24] Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint*.

[25] Betancourt, M. (2018). Calibrating model-based inferences and decisions. *arXiv preprint*.

[26] Bhadra, A., Datta, J., Li, Y. and Polson, N. (2020). Horseshoe regularisation for machine learning in complex and deep models. *International Statistical Review* **88** 302–320. https://doi.org/10.1111/insr.12360

[27] Bhadra, A., Datta, J., Polson, N. G. and Willard, B. (2016). Default Bayesian analysis with global-local shrinkage priors. *Biometrika* **103** 955–969. MR3620450

[28] Bissiri, P. G., Holmes, C. and Walker, S. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **78** 1103–1130. https://doi.org/10.1111/rssb.12158. MR3557191

[29] Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association* **112** 859–877. MR3671776

[30] Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association* **112** 859–877. https://doi.org/10.1080/01621459.2017.1285773. MR3671776

[31] Blumer, A., Ehrenfeucht, A., Haussler, D. and Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters* **24** 377–380. MR0896392

[32] Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H. and Maza, D. (2000). The control of chaos: Theory and applications. *Physics Reports* **329** 103–197. MR1752753

[33] Boelts, J., Lueckmann, J.-M., Gao, R. and Macke, J. H. (2022). Flexible and efficient simulation-based inference for models of decision-making. *eLife* **11** e77220.

[34] Bonat, W. H., Jr, P. J. R. and Zeviani, W. M. (2013). Regression models with responses on the unity interval: Specification, estimation and comparison. *Biometric Brazilian Journal* **30** 18.

[35] Bonchi, F., Hajian, S., Mishra, B. and Ramazzotti, D. (2017). Exposing the probabilistic causal structure of discrimination. *International Journal of Data Science and Analytics* **3** 1–21.

[36] Borsboom, D., Mellenbergh, G. and Heerden, J. (2004). The concept of validity. *Psychological Review* **111** 1061–71. https://doi.org/10.1037/0033-295X.111.4.1061

[37] Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science* **16** 199–231. https://doi.org/10.1214/ss/1009213726. MR1874152

[38] Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research* **70** 245–317. https://doi.org/10.1613/jair.1.12228. MR4224661

[39] Bürkner, P.-C., Gabry, J. and Vehtari, A. (2021). Efficient leave-one-out cross-validation for Bayesian non-factorized normal and Student-t models. *Computational Statistics* **36** 1243–1261. MR4255808

[40] Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software* **80** 1–28.

[41] Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal* **10** 395–411.

[42] Bürkner, P.-C. (2021). Bayesian item response modelling in R with brms and Stan. *Journal of Statistical Software* 1–54.

[43] Bürkner, P.-C. (2022). On the information obtainable from comparative judgments. *Psychometrika* 1–34. Publisher: Springer. MR4504998

[44] Bürkner, P.-C. and Charpentier, E. (2020). Modeling monotonic effects of ordinal predictors in Bayesian regression models. *British Journal of Mathematical and Statistical Psychology* 1–32.

[45] Bürkner, P.-C., Gabry, J., Kay, M. and Vehtari, A. (2022). posterior: Tools for Working with Posterior Distributions. R package version 1.3.0.

[46] Bürkner, P.-C., Gabry, J. and Vehtari, A. (2020). Approximate leave-future-out cross-validation for Bayesian time series models. *Journal of Statistical Computation and Simulation* 1–25. MR4145352

[47] Bürkner, P.-C., Kröker, I., Oladyshkin, S. and Nowak, W. (2022). The sparse Polynomial Chaos expansion: A fully Bayesian approach with joint priors on the coefficients and global selection of terms. *arXiv preprint*. MR4594126

[48] Bürkner, P.-C., Schulte, N. and Holling, H. (2018). On the statistical and practical limitations of Thurstonian IRT models. *Educational and Psychological Measurement* **79** 827–854. Publisher: Los Angelos: Sage.

[49] Bürkner, P.-C. and Vuorre, M. (2019). Ordinal regression models in

psychology: A tutorial. *Advances in Methods and Practices in Psychological Science* **2** 77–101.

[50] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software* **76**.

[51] Carvalho, Polson and Scott (2010). The horseshoe estimator for sparse signals. *Biometrika* **97** 465–480. MR2650751

[52] Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Cengage Learning. MR1051420

[53] Catalina, A., Bürkner, P.-C. and Vehtari, A. (2022). Projection predictive inference for generalized linear and additive multilevel models. *Artificial Intelligence and Statistics (AISTATS) Conference Proceedings*.

[54] Chan, J., Perrone, V., Spence, J., Jenkins, P., Mathieson, S. and Song, Y. (2018). A likelihood-free inference framework for population genetic data using exchangeable neural networks. *Advances in Neural Information Processing Systems* **31**.

[55] Chen, H., Chen, J. and Kalbfleisch, J. D. (2004). Testing for a finite mixture model with two components. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **66** 95–115. MR2035761

[56] Chen, Y., Zhang, D., Gutmann, M., Courville, A. and Zhu, Z. (2020). Neural approximate sufficient statistics for implicit models. *arXiv preprint*.

[57] Chouldechova, A. and Roth, A. (2018). The Frontiers of Fairness in Machine Learning.

[58] Cinelli, C., Forney, A. and Pearl, J. (2020). A crash course in good and bad controls. *SSRN Electronic Journal*. `https://doi.org/10.2139/ssrn.3689437`

[59] Corbett-Davies, S. and Goel, S. (2018). The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint*.

[60] Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2022). *Introduction to Algorithms*. MIT Press. MR2572804

[61] Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association* **91** 883–904. MR1395755

[62] Cranmer, K., Brehmer, J. and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*. MR4263287

[63] Cusumano-Towner, M. F. and Mansinghka, V. K. (2017). Measuring the non-asymptotic convergence of sequential Monte Carlo samplers using probabilistic programming. *arXiv preprint*.

[64] Dai, C., Heng, J., Jacob, P. E. and Whiteley, N. (2020). An invitation to sequential Monte Carlo samplers. *arXiv preprint*. MR4480734

[65] de Oliveira, L., Paganini, M. and Nachman, B. (2017). Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science* **1** 1–24.

[66] de Valpine, P. (2021). A Close Look at Some Linear Model MCMC Comparisons – NIMBLE. https://r-nimble.org/a-close-look-at-some-linear-model-mcmc-comparisons.

[67] Deistler, M., Goncalves, P. J. and Macke, J. H. (2022). Truncated proposals for scalable and hassle-free simulation-based inference. *Advances in Neural Information Processing Systems* **35** 23135–23149.

[68] Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68** 411–436. https://doi.org/10.1111/j.1467-9868.2006.00553.x MR2278333

[69] Depaoli, S., Winter, S. D. and Visser, M. (2020). The importance of prior sensitivity analysis in Bayesian statistics: demonstrations using an interactive Shiny App. *Frontiers in Psychology*.

[70] Dhaka, A. K., Catalina, A., Andersen, M. R., Magnusson, M., Huggins, J. and Vehtari, A. (2020). Robust, accurate stochastic optimization for variational inference. *Advances in Neural Information Processing Systems* **33** 10961–10973.

[71] Diggle, P. J. and Gratton, R. J. (1984). Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)* **46** 193–212. MR0781880

[72] Dinh, L., Sohl-Dickstein, J. and Bengio, S. (2016). Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

[73] Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint*.

[74] Doss, C. R., Flegal, J. M., Jones, G. L. and Neath, R. C. (2014). Markov chain Monte Carlo estimation of quantiles. *Electronic Journal of Statistics* **8** 2448–2478. https://doi.org/10.1214/14-EJS957. MR3285872

[75] Doucet, A., de Freitas, N. and Gordon, N. (2001). An Introduction to Sequential Monte Carlo Methods. In *Sequential Monte Carlo Methods in Practice* (A. Doucet, N. de Freitas and N. Gordon, eds.) 3–14. Springer, New York, NY. https://doi.org/10.1007/978-1-4757-3437-9_1. MR1847784

[76] Drasgow, F., Levine, M. V., Tsien, S., Williams, B. and Mead, A. D. (1995). Fitting polytomous item response theory models to multiple-choice tests. *Applied Psychological Measurement* **19** 143–166. https://doi.org/10.1177/014662169501900203

[77] Draxler, F., Veschgini, K., Salmhofer, M. and Hamprecht, F. (2018). Essentially no barriers in neural network energy landscape. In *Proceedings of the 35th International Conference on Machine Learning* 1309–1318. PMLR.

[78] Duerr, O., Sick, B. and Murina, E. (2020). *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*. Simon and Schuster.

[79] Durán, J. M. (2020). What is a simulation model? *Minds and Machines* **30** 301–323.

[80] Durkan, C., Murray, I. and Papamakarios, G. (2020). On con-

trastive learning for likelihood-free inference. In *International Conference on Machine Learning* 2771–2781. PMLR.

[81] El Moselhy, T. A. and Marzouk, Y. M. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics* **231** 7815–7850. MR2972870

[82] Embretson, S. E. and Reise, S. P. (2000). *Item Response Theory*. Psychology Press.

[83] Emery, A. F. and Nenarokomov, A. V. (1998). Optimal experiment design. *Measurement Science and Technology* **9** 864.

[84] Erdogdu, M. A., Mackey, L. and Shamir, O. (2018). Global nonconvex optimization with discretized diffusions. *Advances in Neural Information Processing Systems* **31**.

[85] Etz, A. and Wagenmakers, E.-J. (2017). J. B. S. Haldane's contribution to the Bayes factor hypothesis test. *Statistical Science* **32**. https://doi.org/10.1214/16-STS599. MR3648962

[86] Evans, M. and Jang, G. H. (2011). Weak informativity and the information in one prior relative to another. *Statistical Science* **26** 423–439. MR2917964

[87] Evans, M. and Moshonov, H. (2006). Checking for prior-data conflict. *Bayesian Analysis* **1** 893–914. MR2282210

[88] Fedorov, V. (2010). Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics* **2** 581–589.

[89] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C. and Venkatasubramanian, S. (2015). Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 259–268. ACM, Sydney NSW Australia. https://doi.org/10.1145/2783258.2783311

[90] Fengler, A., Govindarajan, L. N., Chen, T. and Frank, M. J. (2021). Likelihood approximation networks (LANs) for fast inference of simulation models in cognitive neuroscience. *Elife* **10** e65074.

[91] Flegal, J. M., Haran, M. and Jones, G. L. (2008). Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science* **23**. https://doi.org/10.1214/08-STS257. MR2516823

[92] Floridi, L. and Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* **30** 681–694. https://doi.org/10.1007/s11023-020-09548-1

[93] Foong, A., Burt, D., Li, Y. and Turner, R. (2020). On the expressiveness of approximate inference in bayesian neural networks. *Advances in Neural Information Processing Systems* **33** 15897–15908.

[94] Fox, C. W. and Roberts, S. J. (2012). A tutorial on variational Bayesian inference. *Artificial Intelligence Review* **38** 85–95. https://doi.org/10.1007/s10462-011-9236-8

[95] Frazier, D. T. and Drovandi, C. (2021). Robust approximate Bayesian inference with synthetic likelihood. *Journal of Computational and Graphical Statistics* **30** 958–976. MR4356598

[96] Frazier, D. T., Robert, C. P. and Rousseau, J. (2020). Model mis-

specification in approximate Bayesian computation: consequences and diagnostics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **82** 421–444. MR4084170

[97] Freedman, D. A. (2010). *Statistical Models and Causal Inference: A Dialogue with the Social Sciences*. Cambridge University Press. MR2668307

[98] Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* **19** 1–67. MR1091842

[99] Frome, E. L. (1983). The analysis of rates using Poisson regression models. *Biometrics* **39** 665–674. https://doi.org/10.2307/2531094

[100] Fuglstad, G.-A., Simpson, D., Lindgren, F. and Rue, H. (2019). Constructing priors that penalize the complexity of Gaussian random fields. *Journal of the American Statistical Association* **114** 445–452. MR3941267

[101] Gabaix, X. and Laibson, D. (2008). The seven properties of good models. *The Foundations of Positive and Normative Economics: A Handbook* 292–319.

[102] Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. and Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **182** 389–402. MR3902665

[103] Gao, Y., Kennedy, L., Simpson, D. and Gelman, A. (2021). Improving multilevel regression and poststratification with structured priors. *Bayesian Analysis* **16** 719–744. MR4303866

[104] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P. and Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems* **31**.

[105] Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American statistical Association* **95** 1300–1304. MR1825281

[106] Gelfand, A. E. and Vounatsou, P. (2003). Proper multivariate conditional autoregressive models for spatial data analysis. *Biostatistics* **4** 11–15.

[107] Gelman, A. (2004). Parameterization and Bayesian Modeling. *Journal of the American Statistical Association* **99** 537–545. https://doi.org/10.1198/016214504000000458

[108] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013). *Bayesian Data Analysis (3rd edition)*. Chapman and Hall/CRC. https://doi.org/10.1201/b16018. MR3235677

[109] Gelman, A., Goodrich, B., Gabry, J. and Vehtari, A. (2019). R-squared for Bayesian regression models. *The American Statistician* **73** 307–309. https://doi.org/10.1080/00031305.2018.1549100. MR3989374

[110] Gelman, A. and Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

[111] Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation

using multiple sequences. *Statistical Science* **7** 457–472. MR1294072

[112] Gelman, A., Simpson, D. and Betancourt, M. (2017). The prior can often only be understood in the context of the likelihood. *Entropy* **19** 555–567.

[113] Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C. and Modrák, M. (2020). Bayesian workflow. *arXiv preprint*.

[114] George, E. I., Makov, U. and Smith, A. (1993). Conjugate likelihood distributions. *Scandinavian Journal of Statistics* 147–156. MR1229290

[115] Gertheiss, J. and Tutz, G. (2009). Penalized regression with ordinal predictors. *International Statistical Review* **77** 345–365.

[116] Gesmundo, A. and Dean, J. (2022). An evolutionary approach to dynamic introduction of tasks in large-scale multitask learning systems. *arXiv preprint*.

[117] Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science* 473–483.

[118] Ghosh, S., Yao, J. and Doshi-Velez, F. (2019). Model selection in Bayesian neural networks via horseshoe priors. *Journal of Machine Learning Research* **20** 1–46. MR4048993

[119] Gilmore, R. and McCallum, J. (1995). Structure in the bifurcation diagram of the Duffing oscillator. *Physical Review E* **51** 935. MR1383543

[120] Glymour, C., Zhang, K. and Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in Genetics* **10** 524.

[121] Gneiting, T., Balabdaoui, F. and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69** 243–268. MR2325275

[122] Goel, P. K. and Degroot, M. H. (1981). Information about hyperparameters in hierarchical models. *Journal of the American Statistical Association* **76** 140–147. MR0608185

[123] Gonçalves, P. J., Lueckmann, J.-M., Deistler, M., Nonnenmacher, M., Öcal, K., Bassetto, G., Chintaluri, C., Podlaski, W. F., Haddad, S. A., Vogels, T. P. et al. (2020). Training deep neural density estimators to identify mechanistic models of neural dynamics. *Elife* **9** e56261.

[124] Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. MIT Press. MR3617773

[125] Grazzini, J., Richiardi, M. G. and Tsionas, M. (2017). Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control* **77** 26–47. MR3626045

[126] Greenberg, D., Nonnenmacher, M. and Macke, J. (2019). Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning* 2404–2414.

[127] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research* **13** 723–773. MR2913716

[128] Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., Leslie, D. S., Forster, J. J., Wagenmakers, E.-J. and Steingroever, H. (2017). A tutorial on bridge sampling. *Journal of Mathematical Psychology* **81** 80–97. https://doi.org/10.1016/j.jmp.2017.09.005. MR3722819

[129] Gronau, Q. F., Singmann, H. and Wagenmakers, E.-J. (2020). bridgesampling: An R package for estimating normalizing constants. *Journal of Statistical Software* **92** 1–29. https://doi.org/10.18637/jss.v092.i10

[130] Gu, X., Mulder, J. and Hoijtink, H. (2018). Approximated adjusted fractional Bayes factors: A general method for testing informative hypotheses. *British Journal of Mathematical and Statistical Psychology* **71** 229–261.

[131] Guala, F. (2002). Models, simulations, and experiments. In *Model-based reasoning: Science, technology, values* 59–74. Springer.

[132] Haldane, J. B. S. (1932). A note on inverse probability. In *Mathematical Proceedings of the Cambridge Philosophical Society* **28** 55–61. Cambridge University Press.

[133] Hansen, M. H. and Yu, B. (2001). Model selection and the principle of minimum description length. *Journal of the American Statistical Association* **96** 746–774. MR1939352

[134] Hastie, T., Tibshirani, R., Friedman, J. H. and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* **2**. Springer. MR2722294

[135] Hastings, W. K. (1970). *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*. Oxford University Press. MR3363437

[136] Hermans, J., Begy, V. and Louppe, G. (2020). Likelihood-free mcmc with amortized approximate ratio estimators. In *International Conference on Machine Learning* 4239–4248. PMLR.

[137] Hoban, S., Bertorelle, G. and Gaggiotti, O. E. (2012). Computer simulations: Tools for population and evolutionary genetics. *Nature Reviews Genetics* **13** 110–122.

[138] Hodges, J. S. and Sargent, D. J. (2001). Counting degrees of freedom in hierarchical and other richly-parameterised models. *Biometrika* **88** 367–379. MR1844837

[139] Hoffman, M. and Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*. MR3214779

[140] Holland, P. W. and Wainer, H. (1993). *Differential Item Functioning*. Routledge.

[141] Hossenfelder, S. (2018). *Lost in Math: How Beauty Leads Physics Astray*. Hachette, UK.

[142] Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning* **110** 457–506. MR4234924

[143] Hyttinen, A., Eberhardt, F. and Järvisalo, M. (2015). Do-calculus

when the true graph is unknown. In *UAI* 395–404. Citeseer.

[144] Ibrahim, J. G. and Chen, M.-H. (2000). Power prior distributions for regression models. *Statistical Science* **15** 46–60. MR1842236

[145] Imbens, G. W. and Rubin, D. B. (2015). *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press. MR3309951

[146] Ivanova, D. R., Foster, A., Kleinegesse, S., Gutmann, M. U. and Rainforth, T. (2021). Implicit deep adaptive design: policy-based experimental design without likelihoods. *Advances in Neural Information Processing Systems* **34** 25785–25798.

[147] Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks* **14** 1569–1572.

[148] Izmailov, P., Vikram, S., Hoffman, M. D. and Wilson, A. G. G. (2021). What are Bayesian neural network posteriors really like? In *Proceedings of the 38th International Conference on Machine Learning* 4629–4640. PMLR.

[149] Janitza, S., Strobl, C. and Boulesteix, A.-L. (2013). An AUC-based permutation variable importance measure for random forests. *BMC Bioinformatics* **14** 1–11.

[150] Janson, L., Fithian, W. and Hastie, T. J. (2015). Effective degrees of freedom: a flawed metaphor. *Biometrika* **102** 479–485. MR3371017

[151] Jiang, B., Wu, T.-y., Zheng, C. and Wong, W. H. (2017). Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica* 1595–1618. MR3701500

[152] Jiang, D., Yu, J., Ji, C. and Shi, N. (2011). Asymptotic behavior of global positive solution to a stochastic SIR model. *Mathematical and Computer Modelling* **54** 221–232. MR2801881

[153] Jospin, L. V., Laga, H., Boussaid, F., Buntine, W. and Bennamoun, M. (2022). Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine* **17** 29–48.

[154] Kallioinen, N., Paananen, T., Bürkner, P.-C. and Vehtari, A. (2021). Detecting and diagnosing prior and likelihood sensitivity with power-scaling. *arXiv preprint*.

[155] Kantz, H. (1994). A robust method to estimate the maximal Lyapunov exponent of a time series. *Physics Letters A* **185** 77–87.

[156] Kaplan, D. (2008). *Structural Equation Modeling: Foundations and Extensions* **10**. Los Angelos: Sage.

[157] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S. and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics* **3** 422–440.

[158] Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association* **90** 773–795. MR3363402

[159] Kim, B., Khanna, R. and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! Criticism for interpretability. In *Advances in Neural Information Processing Systems* **29**.

[160] Kim, S., Ma, R., Mesa, D. and Coleman, T. P. (2013). Efficient Bayesian inference methods via convex optimization and optimal trans-

port. In *2013 IEEE International Symposium on Information Theory* 2259–2263. IEEE.

[161] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. *arXiv preprint*.

[162] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1 × 1 convolutions. *Advances in Neural Information Processing Systems* **31**.

[163] Kleijn, B. and van der Vaart, A. (2006). Misspecification in infinite-dimensional Bayesian statistics. *The Annals of Statistics* 837–877. MR2283395

[164] Kleijn, B. and van der Vaart, A. (2012). The Bernstein-Von-Mises theorem under misspecification. *Electronic Journal of Statistics* **6** 354–381. MR2988412

[165] Klinger, E., Rickert, D. and Hasenauer, J. (2018). pyABC: distributed, likelihood-free inference. *Bioinformatics* **34** 3591–3593.

[166] Kobyzev, I., Prince, S. J. and Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43** 3964–3979.

[167] Kocaoglu, M., Snyder, C., Dimakis, A. G. and Vishwanath, S. (2017). Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint*.

[168] Kolczynska, M. and Bürkner, P.-C. (2021). Modeling public opinion over time: A simulation study of latent trend models. *Journal of Survey Statistics and Methodology*.

[169] Kong, A., Liu, J. S. and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* **89** 278–288. https://doi.org/10.1080/01621459.1994.10476469

[170] Koren, I., Tziperman, E. and Feingold, G. (2017). Exploring the nonlinear cloud and rain equation. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27** 013107. MR3594279

[171] Krueger, J. (2001). Null hypothesis significance testing: On the survival of a flawed method. *American Psychologist* **56** 16–26. https://doi.org/10.1037/0003-066X.56.1.16

[172] Lambert, B. and Vehtari, A. (2022). R^*: A robust MCMC convergence diagnostic with uncertainty using decision tree classifiers. *Bayesian Analysis* **17** 353–379. Publisher: International Society for Bayesian Analysis. MR4483223

[173] Lavin, A., Zenil, H., Paige, B., Krakauer, D., Gottschlich, J., Mattson, T., Anandkumar, A., Choudry, S., Rocki, K., Baydin, A. G., Prunkl, C., Paige, B., Isayev, O., Peterson, E., McMahon, P. L., Macke, J., Cranmer, K., Zhang, J., Wainwright, H., Hanuka, A., Veloso, M., Assefa, S., Zheng, S. and Pfeffer, A. (2021). Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint*.

[174] Le, T. A., Baydin, A. G. and Wood, F. (2017). Inference compila-

tion and universal probabilistic programming. In *Artificial Intelligence and Statistics* 1338–1348. PMLR.

[175] Lee, A. and Whiteley, N. (2018). Variance estimation in the particle filter. *Biometrika* **105** 609–625. MR3842888

[176] Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J. and Sohl-Dickstein, J. (2018). Deep neural networks as Gaussian processes. *arXiv preprint*. `https://doi.org/10.48550/arXiv.1711.00165`

[177] Lee, Y., Oh, S. H. and Kim, M. W. (1991). The effect of initial weights on premature saturation in back-propagation learning. In *IJCNN-91-Seattle International Joint Conference on Neural Networks* **i** 765–770 vol.1. `https://doi.org/10.1109/IJCNN.1991.155275`

[178] Lehmann, E. L. and Casella, G. (2006). *Theory of Point Estimation*. Springer Science & Business Media. MR1639875

[179] Liddell, T. M. and Kruschke, J. K. (2018). Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology* **79** 328–348.

[180] Lindgren, F. and Rue, H. (2015). Bayesian spatial modelling with R-INLA. *Journal of Statistical Software* **63** 1–25.

[181] Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics* 986–1005. MR0083936

[182] Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of Atmospheric Sciences* **20** 130–141. MR4021434

[183] Lotfi, S., Izmailov, P., Benton, G., Goldblum, M. and Wilson, A. G. (2022). Bayesian model selection, the marginal likelihood, and generalization. *arXiv preprint*.

[184] Lueckmann, J.-M., Bassetto, G., Karaletsos, T. and Macke, J. H. (2019). Likelihood-free inference with emulator networks. In *Symposium on Advances in Approximate Bayesian Inference* 32–53. PMLR. MR3980813

[185] Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M. and Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems* **30**.

[186] Lunderman, S., Morzfeld, M., Glassmeier, F. and Feingold, G. (2020). Estimating parameters of the nonlinear cloud and rain equation from a large-eddy simulation. *Physica D: Nonlinear Phenomena* **410** 132500. MR4091346

[187] Lurie, A. I. (2002). *Analytical Mechanics*. Springer Science & Business Media. MR1950333

[188] MacEachern, S. N. (2016). Nonparametric Bayesian methods: A gentle introduction and overview. *Communications for Statistical Applications and Methods* **23** 445–466.

[189] MacKay, D. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press. MR2012999

[190] MacKay, D. J. (1995). Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accel-*

erators, Spectrometers, Detectors and Associated Equipment **354** 73–80.

[191] MacKay, D. J. et al. (1998). Introduction to Gaussian processes. *NATO ASI Series F Computer and Systems Sciences* **168** 133–166.

[192] Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint*.

[193] Marin, J.-M., Pudlo, P., Estoup, A. and Robert, C. (2018). *Likelihood-Free Model Choice*. Chapman and Hall/CRC Press. MR3889283

[194] Marjoram, P., Molitor, J., Plagnol, V. and Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* **100** 15324–15328.

[195] Mark, C., Metzner, C., Lautscham, L., Strissel, P. L., Strick, R. and Fabry, B. (2018). Bayesian model selection for complex dynamic systems. *Nature Communications* **9** 1803. https://doi.org/10.1038/s41467-018-04241-5

[196] Martin, G. M., Frazier, D. T. and Robert, C. P. (2021). Approximating Bayes in the 21st century. *arXiv preprint*.

[197] Masegosa, A. (2020). Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems* **33** 5479–5491.

[198] May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature* **261** 459.

[199] Mayo-Wilson, C. and Zollman, K. J. (2021). The computational philosophy: simulation as a core philosophical method. *Synthese* 1–27. MR4341870

[200] McCallum, R. S. (2003). *Handbook of Nonverbal Assessment* **30**. Springer.

[201] McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)* **42** 109–127. MR0583347

[202] McElreath, R. (2020). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Chapman and Hall/CRC.

[203] Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica* **6** 831–860. MR1422406

[204] Mestdagh, M., Verdonck, S., Meers, K., Loossens, T. and Tuerlinckx, F. (2019). Prepaid parameter estimation without likelihoods. *PLoS Computational Biology* **15** e1007181.

[205] Mikkola, P., Martin, O. A., Chandramouli, S., Hartmann, M., Pla, O. A., Thomas, O., Pesonen, H., Corander, J., Vehtari, A., Kaski, S., Bürkner, Paul-Christian and Klami, Arto (2021). Prior knowledge elicitation: The past, present, and future. *arXiv preprint*.

[206] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* **267** 1–38. https://doi.org/10.1016/j.artint.2018.07.007. MR3874511

[207] Minka, T. P. (2013). Expectation propagation for approximate Bayesian inference. *arXiv preprint*.

[208] Modrák, M., Moon, A. H., Kim, S., Bürkner, P., Huurre, N., Faltejsková, K., Gelman, A. and Vehtari, A. (2023). Simulation-based calibration checking for Bayesian computation: The choice of test quantities shapes sensitivity. arXiv:2211.02383 [stat]. https://doi.org/10.48550/arXiv.2211.02383

[209] Molnar, C. (2020). *Interpretable Machine Learning*. Lulu.com.

[210] Morgan, S. L. and Winship, C. (2015). *Counterfactuals and Causal Inference*. Cambridge University Press.

[211] Morris, M., Wheeler-Martin, K., Simpson, D., Mooney, S. J., Gelman, A. and DiMaggio, C. (2019). Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in Stan. *Spatial and Spatio-Temporal Epidemiology* **31** 1–18.

[212] Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability* **29** 429–443. MR1450938

[213] Nalborczyk, L., Bürkner, P.-C. and Williams, D. R. (2019). Pragmatism should not be a substitute for statistical literacy, a commentary on Albers, Kiers, and van Ravenzwaaij (2018). *Collabra: Psychology* **5**.

[214] Neal, R. M. (2011). MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo* 139–188. Chapman and Hall/CRC. MR2858447

[215] Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* **135** 370–384. MR0375592

[216] Noble, D. (2012). A theory of biological relativity: no privileged level of causation. *Interface Focus* **2** 55–64.

[217] Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer. MR1713114

[218] Nussbaumer, A., Pope, A. and Neville, K. (2021). A framework for applying ethics-by-design to decision support systems for emergency management. *Information Systems Journal*.

[219] O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B (Methodological)* **57** 99–118. MR1325379

[220] O'Hagan, A. (2019). Expert knowledge elicitation: Subjective but scientific. *The American Statistician*. MR3925710

[221] Opper, M. and Winther, O. (2000). Gaussian processes for classification: Mean-field algorithms. *Neural Computation* **12** 2655–2684.

[222] Osterlind, S. J. and Everson, H. T. (2009). *Differential Item Functioning* **161**. Sage.

[223] Paananen, T., Piironen, J., Bürkner, P.-C. and Vehtari, A. (2021). Implicitly adaptive importance sampling. *Statistics and Computing* **31** 16. https://doi.org/10.1007/s11222-020-09982-2. MR4216405

[224] Pacchiardi, L. and Dutta, R. (2021). Generalized Bayesian likelihood-free inference using scoring rules estimators. *arXiv preprint*.

[225] Pacchiardi, L. and Dutta, R. (2022). Likelihood-free inference with generative neural networks via scoring rule minimization. *arXiv preprint arXiv:2205.15784*.

[226] Paige, B. and Wood, F. (2016). Inference networks for sequential Monte Carlo in graphical models. *International Conference on Machine Learning* **48** 3040–3049.

[227] Palminteri, S., Wyart, V. and Koechlin, E. (2017). The importance of falsification in computational cognitive modeling. *Trends in cognitive sciences* **21** 425–433.

[228] Papamakarios, G. and Murray, I. (2016). Fast ε-free inference of simulation models with Bayesian conditional density estimation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* 1036–1044.

[229] Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S. and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.* **22** 1–64. MR4253750

[230] Papamakarios, G., Pavlakou, T. and Murray, I. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems* **30**.

[231] Papamakarios, G., Sterratt, D. and Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics* 837–848. PMLR.

[232] Parliament and of the European Union, C. (2016). General data protection regulation.

[233] Parno, M., Moselhy, T. and Marzouk, Y. (2016). A multiscale strategy for Bayesian inference using transport maps. *SIAM/ASA Journal on Uncertainty Quantification* **4** 1160–1190. MR3556075

[234] Pavone, F., Piironen, J., Bürkner, P.-C. and Vehtari, A. (2022). Using reference models in variable selection. *Computational Statistics*. `https://doi.org/10.1007/s00180-022-01231-6`. MR4556024

[235] Pearl, J. (2009). Causal inference in statistics: An overview. *Statistics Surveys* **3** 96–146. MR2545291

[236] Pearl, J. (2009). *Causality*. Cambridge University Press. MR2548166

[237] Pearl, J. (2012). The do-calculus revisited. *arXiv preprint*.

[238] Pearl, J. (2019). The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM* **62** 54–60.

[239] Piantadosi, S. T. (2018). One parameter is always enough. *AIP Advances* **8** 095118.

[240] Piironen, J. and Vehtari, A. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics* **11** 5018–5051. MR3738204

[241] Piironen, J. and Vehtari, A. (2017). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing* **27** 711–735.

https://doi.org/10.1007/s11222-016-9649-y. MR3613594

[242] Plummer, M. et al. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing* **124** 1–10. Vienna, Austria.

[243] Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A. and Feldman, M. W. (1999). Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Molecular Biology and Evolution* **16** 1791–1798.

[244] Pérez, C., Martín, J. and Rufo, M. J. (2006). MCMC-based local parametric sensitivity estimations. *Computational Statistics & Data Analysis* **51** 823–835. MR2297491

[245] Radev, S. T., D'Alessandro, M., Mertens, U. K., Voss, A., Köthe, U. and Bürkner, P.-C. (2021). Amortized Bayesian model comparison with evidential deep learning. *IEEE Transactions on Neural Networks and Learning Systems.*

[246] Radev, S. T., Graw, F., Chen, S., Mutters, N. T., Eichel, V. M., Bärnighausen, T. and Köthe, U. (2021). OutbreakFlow: Model-based Bayesian inference of disease outbreak dynamics with invertible neural networks and its application to the COVID-19 pandemics in Germany. *PLoS Computational Biology* **17** e1009472.

[247] Radev, S. T., Mertens, U. K., Voss, A., Ardizzone, L. and Köthe, U. (2020). BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems.* MR4516681

[248] Radev, S. T., Schmitt, M., Pratz, V., Picchini, U., Köthe, U. and Bürkner, P.-C. (2023). JANA: Jointly Amortized Neural Approximation of complex Bayesian models. *Uncertainty in Artificial Intelligence (UAI) Conference Proceedings.* arXiv:2302.09125 [cs, stat]. https://doi.org/10.48550/arXiv.2302.09125

[249] Radev, S. T., Voss, A., Wieschen, E. M. and Bürkner, P.-C. (2020). Amortized Bayesian inference for models of cognition. In *International Conference on Cognitive Modelling (ICCM).*

[250] Raginsky, M., Rakhlin, A. and Telgarsky, M. (2017). Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory* 1674–1703. PMLR.

[251] Raissi, M. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 22. MR3881695

[252] Ranganath, R., Gerrish, S. and Blei, D. (2014). Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* 814–822. PMLR.

[253] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning* 63–71. Springer.

[254] Raynal, L., Marin, J.-M., Pudlo, P., Ribatet, M., Robert, C. P.

and Estoup, A. (2019). ABC random forests for Bayesian parameter inference. *Bioinformatics* **35** 1720–1728.

[255] Rissanen, J. (1978). Modeling by shortest data description. *Automatica* **14** 465–471.

[256] Robert, C. P., Casella, G. and Casella, G. (1999). *Monte Carlo Statistical Methods* **2**. Springer. MR1707311

[257] Roos, M., Martins, T. G., Held, L. and Rue, H. (2015). Sensitivity analysis for Bayesian hierarchical models. *Bayesian Analysis* **10** 321–349. MR3420885

[258] Rothwell, J. (2014). How the war on drugs damages black social mobility. *The Brookings Institution*.

[259] Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics* **12** 1151–1172. MR0760681

[260] Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC. https://doi.org/10.1201/9780203492024. MR2130347

[261] Rue, H., Martino, S. and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71** 319–392. MR2649602

[262] Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P. and Lindgren, F. K. (2017). Bayesian computing with INLA: A review. *Annual Review of Statistics and Its Application* **4** 395–421. MR3634300

[263] Rust, J. and Golombok, S. (2014). *Modern Psychometrics: The Science of Psychological Assessment (3rd edition)*. Routledge. https://doi.org/10.4324/9781315787527

[264] San Martín, E. (2018). Identifiability of structural characteristics: How relevant is it for the Bayesian approach? *Brazilian Journal of Probability and Statistics* **32** 346–373. MR3787758

[265] San Martin, E. and González, J. (2010). Bayesian identifiability: Contributions to an inconclusive debate. *Chilean Journal of Statistics* **1** 69–91. MR2756120

[266] Schad, D. J., Betancourt, M. and Vasishth, S. (2021). Toward a principled Bayesian workflow in cognitive science. *Psychological Methods* **26** 103.

[267] Schad, D. J., Nicenboim, B., Bürkner, P.-C., Betancourt, M. and Vasishth, S. (2021). Workflow techniques for the robust use of Bayes factors. *Psychological Methods*.

[268] Schafer, T. L. J. and Matteson, D. S. (2023). Locally adaptive shrinkage priors for trends and breaks in count time series. *arXiv preprint*. arXiv:2309.00080 [stat].

[269] Schmitt, M., Bürkner, P.-C., Köthe, U. and Radev, S. T. (2023). Detecting model misspecification in amortized Bayesian inference with neural networks. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*.

[270] Scholz, M. and Bürkner, P.-C. (2022). Prediction can be safely used as a proxy for explanation in causally consistent Bayesian generalized linear models. *arXiv preprint arXiv:2210.06927*.

[271] Schuster, H. G. and Just, W. (2006). *Deterministic Chaos: An Introduction*. John Wiley & Sons. MR2190040

[272] Sharma, S., Sharma, S. and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Science* **6** 310–316.

[273] Shmueli, G. (2010). To explain or to predict? *Statistical Science* **25** 289–310. MR2791669

[274] Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press.

[275] Sisson, S. A., Fan, Y. and Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* **104** 1760–1765. MR2301870

[276] Sornette, D. (2009). Why stock markets crash. In *Why Stock Markets Crash* Princeton University Press.

[277] Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and Van der Linde, A. (1998). Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models Technical Report, Citeseer.

[278] Spirtes, P. and Zhang, K. (2016). Causal discovery and inference: concepts and recent methodological advances. In *Applied Informatics* **3** 1–28. SpringerOpen.

[279] Springer, S., Haario, H., Susiluoto, J., Bibov, A., Davis, A. and Marzouk, Y. (2021). Efficient Bayesian inference for large chaotic dynamical systems. *Geoscientific Model Development* **14** 4319–4333.

[280] Stone, M. (1978). Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics* **9** 127–139. MR0494581

[281] Storch, L. S., Pringle, J. M., Alexander, K. E. and Jones, D. O. (2017). Revisiting the logistic map: A closer look at the dynamics of a classic chaotic population model with ecologically realistic spatial structure and dispersal. *Theoretical Population Biology* **114** 10–18.

[282] Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M. and Dessimoz, C. (2013). Approximate Bayesian computation. *PLOS Computational Biology* **9** e1002803. `https://doi.org/10.1371/journal.pcbi.1002803`. MR3032718

[283] Säilynoja, T., Bürkner, P.-C. and Vehtari, A. (2022). Graphical test for discrete uniformity and its applications in goodness-of-fit evaluation and multiple sample comparison. *Statistics and Computing* **32** 32. `https://doi.org/10.1007/s11222-022-10090-6`. MR4402179

[284] Talts, S., Betancourt, M., Simpson, D., Vehtari, A. and Gelman, A. (2018). Validating Bayesian inference algorithms with simulation-based calibration. *arXiv preprint*.

[285] Tavaré, S., Balding, D. J., Griffiths, R. C. and Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics* **145** 505–518.

[286] Stan Development Team (2022). Stan Modeling Language Users

Guide and Reference Manual, Version 2.30.

[287] Thall, P. F. and Vail, S. C. (1990). Some covariance models for longitudinal count data with overdispersion. *Biometrics* 657–671. MR1085814

[288] Thompson, E. and Varela, F. J. (2001). Radical embodiment: Neural dynamics and consciousness. *Trends in Cognitive Sciences* **5** 418–425.

[289] Turner, B. M. and Sederberg, P. B. (2014). A generalized, likelihood-free method for posterior estimation. *Psychonomic Bulletin & Review* **21** 227–250.

[290] van der Linden, W. J. and Hambleton, R. K. (1997). *Handbook of Modern Item Response Theory*. Springer-Verlag. MR1601043

[291] van der Pas, S. (2021). Theoretical guarantees for the horseshoe and other global-local shrinkage priors. In *Handbook of Bayesian Variable Selection* 133–160. Chapman and Hall/CRC.

[292] van der Schaft, A. (2007). Port-Hamiltonian systems: an introductory survey. In *Proceedings of the International Congress of Mathematicians Madrid, August 22–30, 2006* (M. Sanz-Solé, J. Soria, J. L. Varona and J. Verdera, eds.) 1339–1365. European Mathematical Society Publishing House, Zuerich, Switzerland. https://doi.org/10.4171/022-3/65

[293] Van Erp, S., Oberski, D. L. and Mulder, J. (2019). Shrinkage priors for Bayesian penalized regression. *Journal of Mathematical Psychology* **89** 31–50. Publisher: Elsevier. MR3903921

[294] VanderWeele, T. (2015). *Explanation in Causal Inference: Methods for Mediation and Interaction*. Oxford University Press.

[295] Vehtari, A. (2021). Comparison of MCMC effective sample size estimators. https://avehtari.github.io/rhat_ess/ess_comparison.html.

[296] Vehtari, A., Gelman, A. and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing* **27** 1413–1432. https://doi.org/10.1007/s11222-016-9696-4. MR3647105

[297] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. and Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *Bayesian Analysis* **16**. https://doi.org/10.1214/20-BA1221. MR4298989

[298] Vehtari, A., Gelman, A., Sivula, T., Jylänki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D. and Robert, C. P. (2020). Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. *Jorunal of Machine Learning Research* **21** 1–53. MR4071200

[299] Vehtari, A. and Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys* **6** 142–228. MR3011074

[300] Vehtari, A., Simpson, D., Gelman, A., Yao, Y. and Gabry, J. (2021). Pareto smoothed importance sampling. *arXiv preprint*.

[301] Victoria, A. H. and Maragatham, G. (2021). Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems* **12**

217–223.

[302] Vives, J., Losilla, J.-M. and Rodrigo, M.-F. (2006). Count data in psychological applied research. *Psychological Reports* **98** 821–835.

[303] von Krause, M., Radev, S. T. and Voss, A. (2022). Mental speed is high until age 60 as revealed by analysis of over a million participants. *Nature Human Behaviour* 1–9.

[304] Wagenaar, W. A. and Sagaria, S. D. (1975). Misperception of exponential growth. *Perception & Psychophysics* **18** 416–422.

[305] Wagenmakers, E.-J., Sarafoglou, A. and Aczel, B. (2022). One statistical analysis must not rule them all.

[306] Wainberg, M., Merico, D., Keller, M. C., Fauman, E. B. and Tripathy, S. J. (2022). Predicting causal genes from psychiatric genome-wide association studies using high-level etiological knowledge. *Molecular Psychiatry* **27** 3095–3106. https://doi.org/10.1038/s41380-022-01542-6

[307] Ward, D., Cannon, P., Beaumont, M., Fasiolo, M. and Schmon, S. (2022). Robust neural posterior estimation and statistical model criticism. *Advances in Neural Information Processing Systems* **35** 33845–33859.

[308] Watanabe, S. (2009). *Algebraic Geometry and Statistical Learning Theory*. Cambridge University Press. MR2847990

[309] Watanabe, S. and Opper, M. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research* **11**. MR2756194

[310] Welandawe, M., Andersen, M. R., Vehtari, A. and Huggins, J. H. (2022). Robust, automated, and accurate black-box variational inference. *arXiv preprint*.

[311] Williams, C. K. and Rasmussen, C. E. (1996). Gaussian processes for regression. In *Advances in Neural Information Processing Systems* 514–520.

[312] Williams, D. R., Carlsson, R. and Bürkner, P.-C. (2017). Between-litter variation in developmental studies of hormones and behavior: Inflated false positives and diminished power. *Frontiers in Neuroendocrinology* **47** 154–166.

[313] Winter, B. and Bürkner, P.-C. (2021). Poisson regression for linguists: A tutorial introduction to modelling count data with brms. *Language and Linguistics Compass* **15** e12439.

[314] Wiqvist, S., Frellsen, J. and Picchini, U. (2021). Sequential neural posterior and likelihood approximation. *arXiv preprint*.

[315] Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65** 95–114. MR1959095

[316] Wrinch, D. and Jeffreys, H. (1919). On some aspects of the theory of probability. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **38** 715–731.

[317] Yang, J.-B., Shen, K.-Q., Ong, C.-J. and Li, X.-P. (2009). Feature selection for MLP neural network: The use of random permutation of prob-

abilistic outputs. *IEEE Transactions on Neural Networks* **20** 1911–1922.

[318] Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (2018). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis* **13** 917–1007. MR3853125

[319] Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (2018). Yes, but did it work?: Evaluating variational inference. *Proceedings of Machine Learning Research* **80** 5581–5590.

[320] Yarkoni, T. and Westfall, J. (2017). Choosing prediction over explanation in psychology: Lessons from machine learning. *Perspectives on Psychological Science* **12** 1100–1122.

[321] Zhang, A. Y. and Zhou, H. H. (2020). Theoretical and computational guarantees of mean field variational inference for community detection. *The Annals of Statistics* **48** 2575–2598. MR4152113

[322] Zhang, F. and Gao, C. (2020). Convergence rates of variational posterior distributions. *The Annals of Statistics* **48** 2180–2207. MR4134791

[323] Zhang, Q., Wu, Y. N. and Zhu, S.-C. (2018). Interpretable convolutional neural networks. In *Computer Vision and Pattern Recognition Conference Proceedings* 8827–8836.

[324] Zhang, Q.-s. and Zhu, S.-c. (2018). Visual interpretability for deep learning: A survey. *Frontiers of Information Technology & Electronic Engineering* **19** 27–39. https://doi.org/10.1631/FITEE.1700808

[325] Zhang, Y. D., Naughton, B. P., Bondell, H. D. and Reich, B. J. (2020). Bayesian regression using a prior on the model fit: The R2-D2 shrinkage prior. *Journal of the American Statistical Association* 1–13. MR4436318

[326] Zhou, Y., Johansen, A. M. and Aston, J. A. D. (2016). Toward automatic model comparison: An adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics* **25** 701–726. https://doi.org/10.1080/10618600.2015.1060885. MR3533634