Simulation-Based Stacking

Yuling Yao* **Bruno Régaldo-Saint Blancard*** **Justin Domke**
Flatiron Institute, New York* University of Massachusetts Amherst

Abstract

Simulation-based inference has been popular for amortized Bayesian computation. It is typical to have more than one posterior approximation, from different inference algorithms, different architectures, or simply the randomness of initialization and stochastic gradients. With a consistency guarantee, we present a general posterior stacking framework to make use of all available approximations. Our stacking method is able to combine densities, simulation draws, confidence intervals, and moments, and address the overall precision, calibration, coverage, and bias of the posterior approximation at the same time. We illustrate our method on several benchmark simulations and a challenging cosmological inference task.

1 INTRODUCTION

Simulation-based inference (SBI) has been widely used in scientific computing including biology, astronomy, and cosmology (e.g., Cranmer et al., 2020; Gonçalves et al., 2020; Dax et al., 2021; Hahn et al., 2023, see Appx. A for background). Instead of an explicit likelihood function, SBI only requires a forward model that generates simulated observations given parameters. Despite its popularity, there has been a growing concern about the sampling quality of SBI: how accurate the inference is compared with the true posterior. Simulation-based calibration (SBC, Talts et al., 2018) diagnoses posterior miscalibration. Given sufficient data, it will typically reject the null hypothesis because all computations are approximations. But the goal of computation calibration is not to reject. Given some imperfect inferences, what is next? This paper develops a stacking approach to aggregate these miscalibrated SBI outcomes, such that the aggregated inference is closer to the true posterior.

Moreover, non-mixing computation is prevalent in SBI. For one fixed inference task, practitioners often obtain many different posterior inference results because of different neural network architectures and hyperparameters, because the posterior itself can be multimodal and it is hard for one inference run to traverse across all isolated modes, or because it is cheaper for modern hardware to run many short-and-crude simulation-based inferences in parallel. For illustration, Fig. 1 visualizes the divergent computing results in a challenging cosmology problem (SimBIG, Section 4). After varying hyperparameters of neural posterior estimators (normalizing flows) on seemingly reasonable ranges, we obtain up to 1,000 posterior inferences. The rank statistics of one parameter across four inference runs display various miscalibration types, indicating biases, over- and under-confidence. The expected log densities (the log data likelihood, or the negative loss functions) across 1,000 inferences vary by a range of 1.7 nats.

To handle non-mixing posterior inferences, a remedy is to pick one inference, but the selection procedure itself is noisy. Even if we correctly pick the best inference, not exploiting suboptimal inferences wastes computation, inflates the Monte Carlo variance, and reduces estimation efficiency. Another option is to

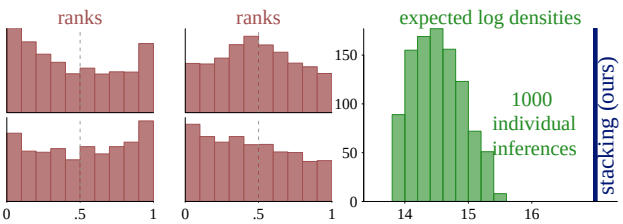

Figure 1: We run 1,000 neural posterior inferences in a challenging cosmology model. The rank histograms of one parameter reveal different types of miscalibration in four runs. The expected log densities of the 1,000 inferences vary by 1.7 nats, while stacking from this paper improves the best approximation by 1.4 nats on holdout simulations.

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. * Joint first authors.

"average over" all inferences. But uniform weighting is generally not optimal and could be a bad idea when there are many bad inferences. Moreover, there are various ways to aggregate inferences, such as taking a linear combination of posterior densities (mixture), a combination of posterior samples, or a combination of confidence intervals. At the same time, posterior approximation can have various goals, such as that the approximate posterior density be close to the truth in some divergence metric, that posterior ranks should be calibrated, that the posterior mean is unbiased, or that the posterior confidence interval attains nominal coverage. If the inference is exact, all of these goals will match. But in reality, most computations are approximate, leading to tension between these goals.

This paper develops a stacking approach to combine multiple simulation-based inferences to improve distributional approximation. As a meta-learning procedure, instances of this framework take many individual inferences as input and output a "stacked" distribution to better approximate the true posterior in a given metric. To make the stacked distribution more flexible, we design three *aggregation forms*: density mixture, sample aggregation, and interval aggregation. To facilitate various user-specific utility of distribution approximating, we design *objective functions* on Kullback–Leibler divergence, rank-based calibration, coverage of posterior intervals, and mean-squared error of moments. Any product of an aggregation form and an objective function renders a stacking method. We develop five practical posterior stacking methods in Section 2. We organize them in a general framework in Section 3, where we further prove that the stacked SBI posterior is asymptotically guaranteed to be the closest to the true posterior distribution in the assigned divergence metric. We recommend hybrid stacking to balance different perspectives of distribution approximation. In Section 4, we illustrate the implementation of our methods in simulated and real-data examples, which involves a cosmology problem regarding galaxy clustering. We discuss related methods and further directions in Section 5.

2 À LA CARTE STACKING

Throughout the paper we work with the general SBI setting where the goal is to sample from a posterior density $p(\theta|y)$ with a potentially intractable likelihood. The parameter space Θ is a subset of $\mathbb{R}^d$, and no assumption is needed for the data space. We create a size-N joint simulation table $\{(\theta_n, y_n)\}_{n=1}^N$ by first drawing parameters θ_n from the prior distribution $p(\theta)$ and one realization of data y_n from the data-forward model $p(y|\theta_n)$. From this simulation table, we run K simulation-based inferences coming from various algorithms or architectures. Given data y_n, the k-th inference, $k = 1, \ldots, K$, returns a learned posterior density $q_k(\theta|y_n)$, and S posterior samples $\tilde{\theta}_{kns}$ from $q_k(\theta|y_n)$, $s = 1, \ldots, S$. The goal of stacking is to construct an ensemble of SBI posteriors, making use of the inferred approximation densities $q_k(\cdot|y)$ or/and the sample draws $\tilde{\theta}$, such that the aggregated approximation is as close to the true posterior $p(\theta|y)$ as possible. See Figure 2 for an illustration.

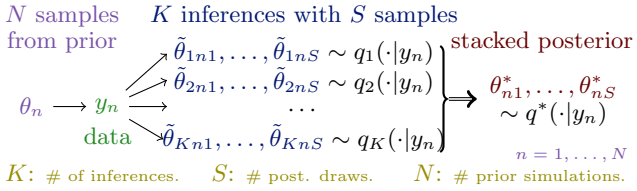

Figure 2: Stacking has two stages. The first is to sample N prior simulations and run K inferences methods, $q_k(\cdot|y), 1 \le k \le K$. The second stage learns a stacked posterior $q^*(\cdot|y)$ to better approximate the true posterior $p(\theta|y)$. We design stacking to facilitate different distribution combination forms and learning objectives.

This section develops five practical posterior stacking algorithms. We defer the related propositions and proofs in Appx. B.

2.1 Density mixture for KL divergence

Perhaps the most natural form to combine posterior densities is to take a linear density mixture

$$q^{\text{mix}}_{\mathbf{w}}(\theta|y) \coloneqq \sum_{k=1}^{K} w_k q_k(\theta|y), \tag{1}$$

where the weight $\mathbf{w} = (w_1, \ldots, w_K)$ lies in a simplex:

$$\mathbf{w} \in \mathbb{S}_K \coloneqq \{0 \le w_k \le 1, \ \sum_{k=1}^{K} w_k = 1\}.$$

To find the optimal weights $\mathbf{w}$, we seek to maximize the log predictive density of $q^{\text{mix}}_w(\theta|y)$, averaged over simulations $\{\theta_1 \ldots, \theta_N\}$,

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w} \in \mathbb{S}_K} \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} w_k q_k(\theta_n|y_n) \right). \tag{2}$$

The expected log density is connected to the Kullback–Leibler (KL) divergence. If the size of the simulation table N is big enough, then up to a constant, the objective function in (2) divided by N converges to the negative conditional KL divergence[1], $\text{KL}(p(\theta|y), q^{\text{mix}}_{\mathbf{w}}(\theta|y))$; see Prop. 2 in Appendix.

[1]Standard notation (Cover and Thomas, 1991) for conditional divergence is $\text{KL}(p(\theta|y) \| q(\theta|y)) \coloneqq \mathbb{E}_{p(y,\theta)} \log(p(\theta|y)/q(\theta|y))$, *not* divergence of conditionals.

Local mixture. All computations are wrong, but some are useful *somewhere*. It is easy to locally adapt the weight $\mathbf{w}(y)$ as a function of data y and output a simplex. In practice, let $\alpha_1(y) = 0$ and for $k > 1$, let $\alpha_k(y)$ be the output of a neural network with its own parameters, and set $w_k(y) = \exp(\alpha_k(y))/\sum_{k=1}^K \exp(\alpha_k(y))$. The locally combined posterior is $q^{\text{mix}}_{\mathbf{w}}(\theta|y) \coloneqq \sum_{k=1}^K w_k(y) q_k(\theta|y)$. Stacking maximizes its log predictive density average over simulations $\max_{\mathbf{w}:y\to\mathbb{S}_K} U(\mathbf{w}) \coloneqq \sum_{n=1}^N \log q^{\text{mix}}_{\mathbf{w}}(\theta_n|y_n)$.

Despite the simplicity, there are two reasons to extend this mixture-stacking-to-max-log-density. First, the mixture has a limited degree of freedom $\mathbf{w} \in \mathbb{S}_K$, which limits the flexibility of the stacked posterior. Second, even in the mixture form, the log-density-based learning (2) does not make use of the existing simulation draws $\{\tilde{\theta}_{kns}\}$, which intuitively can offer more information. The next subsection uses simulation draws.

2.2 Density mixture for rank calibration

From rank-based calibration to rank-based divergence. The rank statistic gives an alternate measurement of posterior approximation quality. For simplicity, first assume the parameter space Θ is one-dimensional. In the k-th inference, we compute the rank statistic (or the p-value) of the prior draw θ_n among its paired posterior $q_k(\theta|y_n)$, i.e.,

$$r_{kn} \coloneqq \frac{1}{S}\sum_{s=1}^S \mathbb{1}\{\tilde{\theta}_{kns} \le \theta_n\}. \tag{3}$$

If the inference is calibrated, $q_k(\cdot|y) = p(\cdot|y)$, then r_{kn} is uniformly distributed over the grid $\{0, 1/S, \cdots, 1\}$. Talts et al. (2018) used this fact to design a rank-based hypothesis testing to test whether the posterior is exact. Taking one step further, we quantify the degree of miscalibration. Given any two conditional distributions $p(\theta|y)$ and $q(\theta|y)$, we define $D_{\text{rank}}(p, q)$, a rank-based generalized divergence metric as follows. Let $D_0(X_1, X_2) \coloneqq \int_0^1 |\Pr(X_1 \le z) - \Pr(X_2 \le z)|^2 dz$ be a distance between two random variables X_1 and X_2 on $[0, 1]$. D_0 compares distributions in cumulative distribution functions (CDFs), which is of the Cramér–von Mises type (Cramér, 1928), and coincides with the continuous ranked probability score (Matheson and Winkler, 1976). Consider CDF transformations: $F_p(\theta|y) \coloneqq \int_{-\infty}^{\theta} p(\theta'|y)d\theta'$ and $F_q(\theta|y) \coloneqq \int_{-\infty}^{\theta} q(\theta'|y)d\theta'$. When (θ, y) are distributed from $p(\theta, y)$, $F_p(\theta|y)$ and $F_q(\theta|y)$ are two random variables on [0,1]. Then define

$$D_{\text{rank}}(p, q) \coloneqq D_0\left(F_p(\theta|y), F_q(\theta|y)\right), \quad (\theta, y) \sim p. \tag{4}$$

This metric $D_{\text{rank}}(p, q)$ is non-negative and its zero is attained when $q(\theta|y) = p(\theta|y)$ almost everywhere (Prop. 3) and hence a generalized divergence. Our defined $D_{\text{rank}}(p, q)$ is appealing since it admits a straightforward empirical estimate, $D_{\text{rank}}(p(\cdot|y), q_k(\cdot|y)) \approx \int_0^1 \left(\hat{F}_{r_k}(t) - t\right)^2 dt$, where $\hat{F}_{r_k}(t) = \frac{1}{N}\sum_{n=1}^N \mathbb{1}(r_{kn} \le t)$ is the empirical CDF of ranks. This integral is computed in a closed form (Appx. C). Moreover, this sample estimate is differentiable on r_{kn} almost everywhere, in contrast to the familiar Kolmogorov–Smirnov test which takes the supremum norm or the Chi-squared test which requires binning.

Rank in the mixture stacking is linear. With K approximate inferences, we still study a mixture posterior $q^{\text{mix}}_{\mathbf{w}}(\theta|y_n) = \sum_{k=1}^K w_k q_k(\theta|y_n)$ and want it to be as correct as possible under rank-based calibration. Conveniently, the rank of θ_n in any $\mathbf{w}$-weighted mixture has an explicit expression using individual ranks,

$$r^{\text{mix}}_n \coloneqq \sum_{k=1}^K w_k r_{kn}. \tag{5}$$

Let θ^{mix} denotes a random variable with the law $q^{\text{mix}}_{\mathbf{w}}(\cdot|y_n) = \sum_{k=1}^K w_k q_k(\theta|y_n)$. For any fixed $\mathbf{w}$ and θ_n, as $S \to \infty$, this r^{mix}_n is a consistent estimate of the mixture CDF, i.e., $r^{\text{mix}}_n|\mathbf{w}, \theta_n \to \Pr(\theta^{\text{mix}} \le \theta_n)$; see Prop. 4. The linear-additivity (5) of the rank statistics can be extended to the local mixture, where the rank of the local mixture posterior is $r^{\text{mix}}_n \coloneqq \sum_{k=1}^K w_k(y_n) r_{kn}$.

Stacking for rank calibration. With the rank-based divergence D_{rank} and the closed form mixture rank r^{mix}_n in (5), we are now ready to run a calibration-aware stacking. We seek to minimize the rank-based divergence $D_{\text{rank}}(p(\cdot|y), q^{\text{mix}}_{\mathbf{w}}(\cdot|y))$ by

$$\min_{\mathbf{w}\in\mathbb{S}_K} \int_0^1 \left(\hat{F}_{r^{\text{mix}}}(t) - t\right)^2 dt, \tag{6}$$

where $\hat{F}_{r^{\text{mix}}}(t) = \frac{1}{N}\sum_{n=1}^N \mathbb{1}\left(\sum_{k=1}^K w_k r_{kn} \le t\right)$. The integral has a closed-form expression and hence the optimization is straightforward (Appx. C).

In addition to D_0 that matches the CDFs of the stacked ranks r^{mix} and the uniform distribution, we can also match their moments, such as to minimize the squared errors, $(\sum_{n=1}^N \sum_{k=1}^K w_k r_{kn}/N - 1/2)^2$ and $(\sum_{n=1}^N \sum_{k=1}^K \log(w_k r_{kn})/N + 1)^2$. Along with (6), these rank-based stacking objectives encourage uniform ranks. As an orthogonal complement to log density stacking (2), rank-based stacking depends on the approximate inferences $\{q_k(\cdot|y)\}$ through and only through their *sample draws*, not *densities*, which is especially suitable when we cannot evaluate the inferences densities such as in short MCMC and GAN.

In reality Θ is not one dimensional. Similar to the practice of SBC, either we pick a one-dimensional

summary statistic $f(\theta, y)$, compute its rank $r_{kn} := \sum_{s=1}^{S} \mathbb{1}\{f(\tilde{\theta}_{kns}, y_n) \leq f(\theta_n, y_n)\}/S$, and run one-dimensional stacking (6) for targeted calibration, or we compute ranks for each dimension separately and sum up the objective function (6) on every dimension.

2.3 Sample stacking for discriminative calibration

So far we only consider density mixtures. It is also natural to work with samples directly. For a given n and any s, $(\theta_{1ns}, \theta_{2ns}, \ldots, \theta_{Kns})$ are posterior draws from K inferences for the same inference task $p(\theta|y_n)$. For example, a linear additive model stacks K approximate samples into one aggregated draw:

$$\theta^*_{ns} = \mathbf{w}_0 + \mathbf{w}_1\tilde{\theta}_{1ns} + \cdots + \mathbf{w}_K\tilde{\theta}_{Kns}, \tag{7}$$

where the parameter $\mathbf{w}_0 \in \mathbb{R}^d, \mathbf{w}_k \in \mathbb{R}^{d\times d}, k \geq 1$.

We want the aggregated sample $\{\theta^*_{n1}, \cdots, \theta^*_{nS}\}$ to be a better sample approximation of the posterior $p(\theta|y_n)$. To measure the sampling quality in SBI, we adopt discriminative calibration (Yao and Domke, 2023): if no classifier can distinguish between $\{(\theta_n, y_n)\}$ and $\{(\tilde{\theta}^*_{n1}, y_n), \cdots, (\tilde{\theta}^*_{nS}, y_n)\}$, then the stacked inference is accurate. Formally, for the n-th simulation run, we create $S+1$ binary classification examples. The first example is $\phi = (\theta_n, y_n)$ with label $z = 1$, and the $(s+1)-$th example is $\phi = (\tilde{\theta}^*_{ns}, y_n)$ with label $z = 0$. Looping over $1 \leq n \leq N$ yields $N(S+1)$ examples $\{(\phi, z)\}$, in which ϕ depends on stacking weights $\mathbf{w}$ via $\tilde{\theta}$. Denote $P(z|\phi)$ to be a probabilistic classifier that predicts label z using feature ϕ, where we reweight the classification loss function to balance two classes. Sample-based stacking solves a minimax optimization:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \max_{P} \sum_{i=1}^{N(S+1)} \log P(z_i|\phi_i). \tag{8}$$

Let $q^*_{\mathbf{w}}(\theta|y)$ be the distribution of the stacked samples (7). As $N \to \infty$, this stacked $q^*_{\hat{w}}(\theta|y)$ minimizes the Jensen-Shannon (JS) divergence between any $q^*_{\mathbf{w}}(\theta|y)$ and true posterior $p(\theta|y)$. See Prop. 5.

2.4 Interval stacking for conformality

Often the focus of Bayesian inference is to correctly quantify the uncertainty in one of a few parameters for downstream tasks such as hypothesis tests or decision theory tasks. For simplicity, again assume a one-dimensional parameter space of interest Θ (otherwise, stack each dimension separately). Given any y_n, in the k-th inference, let l_{kn} and r_{kn} be the left and right interval endpoint of the $(1-\alpha)$ central confidence interval in $q_k(\theta|y_n)$, which typically is computed via the $\alpha/2$ and $1-\alpha/2$ sample quantiles in $\{\tilde{\theta}_{kns} : 1 \leq s \leq S\}$. If the inference is calibrated or conformal, the coverage probability of this interval should be at least $(1-\alpha)$ under the true posterior $p(\theta|y_n)$.

To achieve appropriate coverage, we stack K individual posterior intervals $\{(l_{kn}, r_{kn}) : k \leq K\}$ to produce an aggregated interval (l^*_n, r^*_n). We adopt a simple linear form with the stacking parameter $\mathbf{w} \in \mathbb{R}^{2K}$,

$$l^*_n = \sum_{k=1}^{K} w_k l_{kn}, \quad r^*_n = \sum_{k=1}^{K} w_{k+K} r_{kn}. \tag{9}$$

Besides the correct coverage, we also want the posterior interval to be as narrow as possible to enhance estimation efficiency. The trade-off between coverage and efficiency has been studied in the prediction literature (Gneiting and Raftery, 2007). We design the following interval score stacking, which encourages the coverage and penalizes the length:

$$\begin{aligned} &\min_{\mathbf{w}} \sum_{i=1}^{N} U(r^*_n, l^*_n, \theta_n), \\ \text{where } U(r, l, \theta) :=& (r-l) + \frac{2}{\alpha}(l-\theta)\mathbb{1}(\theta < l) \\ &+ \frac{2}{\alpha}(\theta - r)\mathbb{1}(\theta > r). \end{aligned} \tag{10}$$

The stacked interval (r^*_n, l^*_n) asymptotically provides the optimal posterior quantile estimation—As N approaches ∞, the unique minimizer to the loss function above is when the stacked interval $(r^*(y), l^*(y))$ is identical to the exact pair of the true $\alpha/2$ and $1-\alpha/2$ quantiles in $p(\theta|y)$ for almost every y (Prop. 6).

Unlike the density mixture or sample addition, the stacked interval (9) is reduced-form: we do not specify how to sample from the stacked distribution. Our interval stacking (10) is a semiparametric approach in which any aspect of the posterior distribution other than the quantile is treated as a nuisance parameter.

2.5 Moment stacking for unbiasedness/MSE

Perhaps the posterior mean and covariance remain the two most important summaries of the posterior distribution. We can directly stack these summaries from K approximations. In the k-th individual approximation, the sample mean of $q_k(\cdot|y_n)$ is $\mu_{kn} := \sum_{s=1} \tilde{\theta}_{kns}/S$. In the mixture stacking, the posterior mean of $q^{\text{mix}}_{\mathbf{w}}(\cdot|y_n) = \sum_k w_k q_k(\cdot|y_n)$ is $\mu^*_n(\mathbf{w}) = \sum_k w_k \mu_{kn}$. For sample-based stacking (7), the posterior mean is similar $\mu^{\text{mix}}_n(\mathbf{w}) = \mathbf{w}_0 + \sum_{k=1} \mathbf{w}_k\, \mu_{kn}$. In either case, we can optimize stacking weights to match the first moment,

$$\min_{\mathbf{w}} \sum_{n=1}^{N} ||\mu^*_n(\mathbf{w}) - \theta_n||^2. \tag{11}$$

Likewise, the sample covariance of the k-th posterior inference given y_n is $V_{kn} \coloneqq \sum_{s=1}^{S} ||\tilde{\theta}_{kns} - \mu_{kn}||^2/S$. Using the law of total variation, the covariance of the mixture $q^{\text{mix}}_{\mathbf{w}}(y_n)$ is $V^{\text{mix}}_n(\mathbf{w}) = \sum_k \mathbf{w}_k V_{kn} + \sum_k \mathbf{w}_k ||\mu_{kn} - \bar{\mu}_n||^2$, where $\bar{\mu}_n = \sum_k w_k \mu_{kn}$. We design moment stacking to minimize the following negative-oriented objective function which matches the two moments at the same time:

$$\min_{\mathbf{w}} \sum_{n=1}^{N} U(q^{\text{mix}}_w(\cdot|y_n), \theta_n), \tag{12}$$

where

$$U(q^{\text{mix}}_w(\cdot|y_n), \theta_n) \coloneqq \log \det V^{\text{mix}}_n(\mathbf{w}) + ||\mu^{\text{mix}}_n(\mathbf{w}) - \theta_n||^2_{(V^{\text{mix}}_n(\mathbf{w}))^{-1}} \tag{13}$$

where $||u||^2_\Gamma \coloneqq u^T \Gamma u$ is the weighted norm. As $N \to \infty$, the minimal of the loss function is achieved if and only for almost sure y, the stacked mean $\mu^{\text{mix}}_{\mathbf{w}}$ and covariance $V^{\text{mix}}_{\mathbf{w}}$ exactly matches the $\mathbb{E}(\theta|y)$ and $\text{Var}(\theta|y)$ in the the true posterior (Prop. 7).

3 UNIFIED POSTERIOR STACKING

In this section, we give a unified presentation of the previous five stacking methods in Sec. 2. We observe that they principally vary along two dimensions:

I. What is the combination form? Consider K conditional distributions $q_1(\cdot|y), q_2(\cdot|y), \cdots, q_K(\cdot|y)$ that have support on Θ and represent approximations of the posterior distribution given the same y. We define a *combination form* Φ that maps K conditional distributions into one stacked conditional distribution:

$$\Phi: \quad \{q_1(\cdot|y), q_2(\cdot|y), \cdots, q_K(\cdot|y)\} \to q^*_{\mathbf{w}}(\cdot|y). \tag{14}$$

where $\mathbf{w}$ is the stacking parameter. The output $q^*_{\mathbf{w}}(\cdot|y)$ should be understood as a conditional distribution, which does not necessarily require an explicit density.

II. What is the objective function? To evaluate how well the stacked approximate inference $q^*_{\mathbf{w}}(\cdot|y)$ approximates the true posterior distribution, we need a utility function. We formulate this sampling valuation into a conditional prediction evaluation task. The simulation table gives paired simulations (y, θ) from their joint distribution $p(y, \theta)$, such that for any y, the paired θ can be viewed as an independent draw from the unknown posterior $p(\theta|y)$. The stacked inference $q^*_{\mathbf{w}}(\cdot|y)$ is a conditional distribution of θ given y. A *scoring rule* (Gneiting and Raftery, 2007) is a bivariate function that compares any Θ-supported distribution $q(\cdot)$ and a realization θ,

$$U: (q, \theta) \to \mathbb{R}, \quad \theta \in \Theta. \tag{15}$$

	log score	rank	JS div.	interval	moment
mixture	2.1	2.2			2.5
sample			2.3		2.5
interval				2.4	

Table 1: Table of five stacking methods in relation to combination forms and utility functions. We have used three combination forms: (a) density mixture, (b) sample aggregation, and (c) interval aggregation, and five utility functions (goals): (i) log score (2), (ii) rank based calibration (4), (iii) negative Jensen-Shannon divergence, (iv) interval coverage (10), and (v) moment score (13). Adding up utility functions from one row forms a hybrid stacking.

Table 1 summarizes where our developed methods fit along the combination forms and utility functions. The table is sparse: it is challenging to fill the remaining entries. For example, the confidence interval of the mixture or the density of sample aggregation is intractable. We now explore general posterior stacking with an arbitrary combination form and utility.

Learning and consistency. We need two conditions to produce a valid posterior stacking method. First, we need to evaluate the score of the stacked distribution, $U(q^*_{\mathbf{w}}(\cdot|y_n), \theta_n)$. Second, the scoring rule U in (15) needs to be proper, i.e., for any two θ-distributions p and q,

$$\int_\Theta U(q, \theta) p(\theta) d\theta \le \int_\Theta U(p, \theta) p(\theta) d\theta. \tag{16}$$

These two conditions produces a stacking method. We combine K posterior inferences with the form (14), and fit stacking parameters $\mathbf{w}$ via a sample M-estimate,

$$\hat{\mathbf{w}} = \arg\max \sum_{n=1}^{N} U(q^*_{\mathbf{w}}(\cdot|y_n), \theta_n). \tag{17}$$

The expectation $\mathbb{E}_{p(y,\theta)} U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta)$ is the average utility function of the stacked posterior. Unlike a typical Bayesian prediction evaluation task where there are a large number of observations from one fixed data generating process, here for each fixed y_n we only have one draw θ_n from the true posterior $p(\theta|y)$. As reassurance, the next proposition shows that stacking estimate $\hat{\mathbf{w}}$ from (17) is asymptotically optimal.

Proposition 1. *If the score U is proper, then for any $\epsilon > 0$ and any given $\mathbf{w}'$, as $N \to \infty$,* $\Pr\left(\mathbb{E}_{p(y,\theta)} U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta) \le \mathbb{E}_{p(y,\theta)} U(q^*_{\hat{\mathbf{w}}'}(\cdot|y), \theta) + \epsilon\right) \to 1$.

This M-estimator (17) covers all stacking procerures in this paper except for the rank stacking (6). In a companion paper (Yao et al., 2024), we derive more

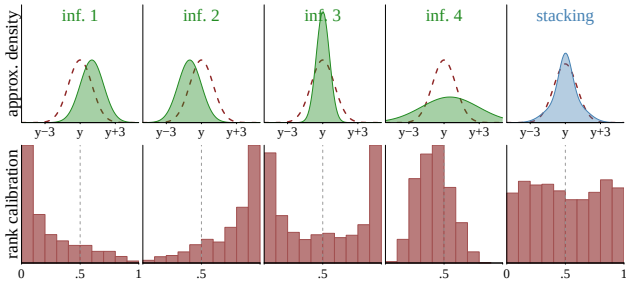

Figure 3: Tension among objectives: Four approximate inferences have the same KL divergence to the true posterior, but differ a lot in the bias, coverage, and rank calibration.

theories and prove the convergence rate and asymptotic normality of the rank-stacking estimator (6) and the M-estimator (17).

A proper scoring rule U produces a generalized divergence by defining $D_U(p,q) = \int_\Theta U(p,\theta)p(\theta)d\theta - \int_\Theta U(q,\theta)p(\theta)d\theta$. Proposition 1 implies that the stacked approximation $q^*_{\hat{\mathbf{w}}}(\cdot|y)$ is asymptotically optimal as its divergence D_U from true posterior $p(\cdot|y)$ is minimized among all possible combinations of the given form (14).

Hybrid stacking. The five stacking methods developed in Sec. 2 cannot exhaust all plausibility. In particular, given a combination operators $q^*_{\mathbf{w}}(\cdot|y)$, if multiple scores $U_1, \ldots, U_m$ satisfy the proper condition (16), then the augmented score $U_1 + \cdots + U_m$ is still valid, and hence existing stacking methods are building blocks toward other stacking approaches. For example, when using the density mixtures, $q^*_{\mathbf{w}}(\cdot|y) = w_k q_k(\theta|y)$, to maximize the hybrid objective $\sum_n \log \sum_k w_k q_k(\theta_n|y_n) - \lambda_2 \sum_n ||\sum_k w_k \mu_{kn} - \theta_n||^2 - \lambda_3(\frac{1}{N}\sum_n \sum_k \log(w_k r_{kn}) + 1)^2$ combines needs for KL-closeness, unbiasedness, and rank calibration.

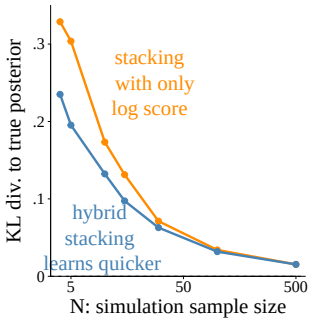

Figure 4: The KL divergence between true posterior to the stacked posterior as training size N varies.

Probabilistic distributions on $\mathbb{R}^d$ are infinite dimensional objects. In contrast to all L^p norms that are equivalent in a finite-dimensional Euclidean space, here these scoring rules gauge different projections of the distribution and can provide nearly orthogonal signals. For instance, suppose the true posterior is $\theta|y \sim \text{normal}(y, 1)$, the green curves in Fig. 3 represent four approximate inferences indistinguishable under the log score as they have the same KL divergence to the true posterior. However, their bias varies from 1 to -1, the real coverage of their nominal 95% confidence varies from 73% to 100%, and their rank distribution can display severe under- or over-confidence. Hybrid stacking, shown as the blue curve, makes use of all signals and improves both density-fitting and calibration. Indeed, even when the KL divergence of the posterior is of interest, adding more information such as rank calibration into stacking objectives boosts efficiency. Fig. 4 shows that hybrid stacking has a quicker learning rate, and its posterior inference is more accurate than the plain log score stacking (2) when the training size N is small. Details of the example are in Appx. D.

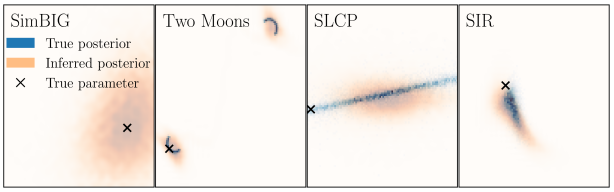

Figure 5: Examples of true and one inferred posteriors in four models. We visualize two margins of the parameter.

General recommendations. *Training-validation split:* To avoid overfitting in stacking, we split the simulation table $\{(\theta, y, \tilde{\theta})\}$ into training and validation parts. We train individual inferences using the training data and train the stacking weights (17) on the validation data. We use extra holdout test data to evaluate the final stacked posterior quality. *Fast optimization:* All objective functions we derived in Section 2 are (almost everywhere) smooth and straightforward to deploy any (stochastic) gradient optimization recipe. The weights in mixture stacking needs a simplex constraint, for which the multiplicative gradient optimization (Zhao, 2023) is suitable. Appx.C.1 discusses smooth approximation of indicator functions. *Quasi Monte Carlo sampling:* When sampling from the stacked inference $\sum_k w_k q_k(\cdot|y)$, the quasi Monte Carlo strategy reduces the variance (Appx.C.2).

4 EXPERIMENTS

We conduct stacking experiments on a variety of inference tasks taken from the SBI benchmark (`sbibm`, Lueckmann et al., 2021) and a practical cosmology problem. These tasks are selected to showcase different computational challenges relating to the geometrical complexity of the posterior, the high dimensionality of the parameters/data, or the limited amount of training examples. Table 2 reports the dimensions and number of training examples involved for each task, and Fig. 5

Task	Settings			Mixture for KL [Log Pred. Density] ↑			Interval Stacking [Coverage Error %] ↓			Moments Stacking [Moments Error] ↓		
	dim($\boldsymbol{\theta}$)	dim($\boldsymbol{y}$)	N	Best	Unif.	Stacked	Best	Unif.	Stacked	Best	Unif.	Stacked
Two Moons	2	2	10k	2.84	2.07	**2.88**	3.25	5.25	**2.75**	-1.72	-1.41	**-1.73**
SLCP	5	8	10k	-5.60	-6.15	**-5.24**	3.76	5.18	**1.16**	1.03	1.25	**0.93**
SIR	2	10	10k	7.57	7.01	**7.65**	2.90	9.35	**1.55**	-8.86	-5.19	**-8.92**
SimBIG	14	3,677	18k	15.6	16.7	**17.0**	6.08	5.85	**4.26**	-3.64	-3.71	**-3.79**

Table 2: Settings for each task and results for best inferred posterior, uniform ensemble, and stacked posterior. Settings: dimensions of parameters $\boldsymbol{\theta}$, data $\boldsymbol{y}$, and number of training examples N. Mixture for KL: log predictive density on holdout test set (higher is better). Interval stacking: parameter-averaged coverage error for the 90% credible intervals on holdout test set (lower is better). Moments stacking: error of posterior moments (13) on holdout test set (lower is better).

visualizes posterior examples.

SBI benchmark: We consider the **Two Moons** (Greenberg et al., 2019), the simple likelihood and complex posterior (**SLCP**, Papamakarios et al., 2019), and an ODE-based **SIR** model. *Practical cosmology model:* We consider a cosmological inference task pertaining to the analysis of galaxy clustering: **SimBIG** (Hahn et al., 2023; Régaldo-Saint Blancard et al., 2023). The SimBIG model involves 14 key physical parameters to describe the evolution of the Universe. We aim to infer from a vector of 3,677 statistical measurements derived from a large galaxy survey.

For each task, we run $K = 50$ (`sbibm`) / $K = 100$ (SimBIG) independent amortized posterior inferences using the Python package `sbi` (Tejero-Cantero et al., 2020). We focus on neural posterior estimators made of conditional normalizing flows. These build on a masked autoregressive flow (**MAP**, Papamakarios et al., 2017) architecture, and a multilayer perceptron (MLP) conditioner. Training consists of maximizing the log predictive density using Adam (Kingma and Ba, 2015) over a fixed number of epochs for the `sbibm` tasks or using an early-stopping procedure for SimBIG: run until the validation loss stops increasing over 20 consecutive epochs. For each inference, we randomly select the number of MAF autoregressive layers, number of MLP hidden layers and units, MLP dropout rate, learning rate, and batch size.

In Appx. D, we include additional experiments that runs stacking on neural spline flow (**NLP**, Durkan et al., 2019) and other SBI benchmark tasks, and we see the same conclusion holds. We give experiment details in Appx. D. We also provide our Python/PyTorch code on GitHub [2].

Stacking reduces KL gap. For each task with K inferences, we run mixture density stacking (1) to maximize the log score (2) trained on a validation set of 1,000 simulations. We compare in Table 2 the expected log predictive density of the posterior approximation computed on a holdout set. This is the negative KL divergence from the true posterior up to a constant. We evaluate three meta posterior approximations: (a) the best single approximation, (b) a uniform weighting of all approximations, and (c) stacking. Stacking has the biggest expected log predictive densities in all cases, indicating a closer posterior inference. The same conclusion holds when we run stacking on neural spline flows in the Appx. D.

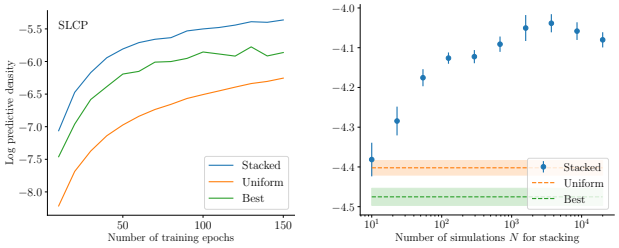

Figure 6: Mixture for KL: (Left) expected log predictive density as a function of training epochs for the SLCP task. Stacking inferences individually trained over $\sim$ 50 epochs performs better than the best single inference trained over 150 epochs. (Right) evolution of the expected log predictive density with varying stacking simulation size N on the SLCP task.

Stacking performs better with less computation. The stacked posterior $q_{\mathbf{w}}^{\text{mix}}$ of K inferences each trained for a fixed number of epochs can reach a better approximation than the best single approximation among a series of K inferences trained after a larger number of training epochs. In the left panel of Fig. 6, we show an illustration of this phenomenon for the SLCP task. In 50 epochs, the stacked posterior already performs better than the best single approximation obtained after 150 epochs, which illustrates the interest in stacking for a limited wall time budget. The right panel of Fig. 6 shows how the performance of stacking changes as the simulation size N varies. We plot the hold-out-data log predictive density of the stacked posterior with neural spline flows, constructed from

[2] https://github.com/bregaldo/simulation_based_stacking.

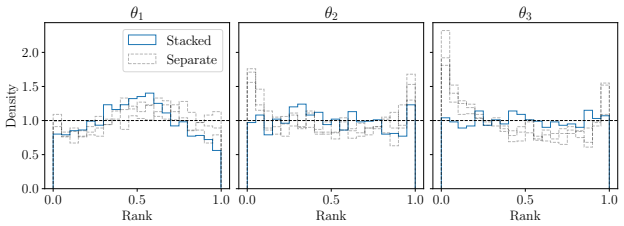

Figure 7: Ranks statistics before and after rank stacking for the SimBIG task. Stacked rank statistics are closer to the uniform distribution (black dashed lines), thus indicating better calibration than individual posteriors.

varying simulation size N. With a very small sample size, stacking can suffer from overfitting and regularization can help.

Stacking calibrates rank statistics. We run stacking for rank calibration (6) on the SimBIG task. Fig. 7 shows the rank statistics obtained from the optimal stacked posterior for the first three cosmological parameters, and compares them to the same rank statistics obtained from three individual posteriors. We constrain the rank statistics of each dimension simultaneously. There is a clear improvement on ranks of parameters θ_2 and θ_3, approaching uniformity. However, for parameter θ_1 the stacked rank statistics display the same kind of underconfidence patterns as individual posteriors. It happens that all individual neural posterior approximations are underconfident for this dimension θ_1. Because any mixture of underconfident approximations remains underconfident, rank-stacking cannot help on the θ_1 margin.

Stacking calibrates confidence intervals. For each task, we perform interval stacking as described in Sec. 2.4 for all parameters simultaneously and focusing on 90% central confidence intervals (i.e. $\alpha = 0.1$). For any scalar parameter θ, we compute the coverage under the true posterior on holdout test data $C_\alpha = \mathbb{E}_{p(y,\theta)}\left[\mathbb{1}\{\theta \in I_{\alpha,q(\cdot|y)}\}\right]$ where $I_{\alpha,q(\cdot|y)}$ is the central confidence interval in approximation $q(\cdot|y)$. If the approximation is perfect, then C_α should be $1-\alpha$. Table 2 reports coverage error $|C_\alpha - (1-\alpha)|$, averaged over all parameters, for best single approximation, uniform ensemble, and stacking. In every task, interval stacking clearly improves coverage.

Stacking calibrates moments. We run moment stacking to calibrate the posterior means and variances by optimizing the moments objective (13). We compare in Table 2 the expected error (13) of posterior means and variances of best single approximation, uniform ensemble, and moment stacking. All errors are computed on holdout test set. Our moment stacking methods outperforms for all tasks.

5 DISCUSSIONS

Stacking/model averaging. Stacking (Wolpert, 1992; Breiman, 1996) is an old idea to combine learning algorithms. Classic stacking combines point predictions only. Recently stacking is advocated to combine Bayesian outcome-predictive-distributions (Clyde and Iversen, 2013; Le and Clarke, 2017; Yao et al., 2018), since it is more robust against model misspecification than Bayesian model averaging (e.g., Hoeting et al., 1999). Traditional stacking aims at the prediction of data y and uses the exchangeability therein. Suppose $\{y_i\}_{i=1}^N$ are IID observations, and $p_k(y_i)$ is the predictive density of y_i in the k-th model, which is only available in likelihood-tractable settings, then stacking seeks to maximize $\sum_{i=1}^n (\log \sum_{k=1}^K w_k p_k(y_i))$, so that the combined predictive density is close to the true data-generating process. In contrast, our paper aims at Bayesian computation and uses the exchangeability of amortized simulations. We do not need a tractable likelihood, nor any structure of data y. Although scoring rules is not a new idea to Bayesian model *evaluation* (Gneiting and Raftery, 2007; Vehtari and Ojanen, 2012), as far as we know, traditional Bayesian stacking is not beyond the log scores and mixture until this paper, while the present paper introduces new combination forms and distributional scoring rules to *learn* stacking weights. Our stacking approach is useful not only to combine different posterior approximations in Bayesian computation, but also to combine different probabilities models to fit observations. In a companion paper (Yao et al., 2024), we establish a general stacking framework to combine probabilistic predictive distributions and provide more theory discussions.

Meta-learning for multi-run Bayesian computation. Modern hardware have attracted the development of parallel Bayesian inferences. One strategy is to tailor MCMC tuning criterion for parallel runs to boost mixing (Hoffman et al., 2021). Another strategy is to run inference methods on subsamples of the dataset and combine the subsampled posteriors to be an unnormalized product $\prod_k q_k(\theta|y)$ (e.g., Nemeth and Sherlock, 2018; Mesquita et al., 2020; Agrawal and Domke, 2021). More generally, it is appealing to run many shorter, and potentially biased inferences and combine them. In this direction, the most related approach to our paper is to use stacking in non-mixing Bayesian computations (Yao et al., 2022). Despite a similar title, the existing stacking-for-computation approach aims to improve how good the statistical model predicts future outcomes. It mixes posterior

predictive distributions to optimize the data score, $\mathbb{E}\log p(\tilde{y}) = \int_\Theta p(\tilde{y}|\theta)q(\theta|y)d\theta$, which is only tractable with a known likelihood, and arguably less relevant to scientific computing where parameters have physical meanings. Our paper has a fundamentally different goal on the inference accuracy.

Simulation-based inference and calibration. Many individual objective functions of our stacking have been used as part of simulation-based inference or calibration. Maximizing the stacked log predictive density shares the same goal of minimizing $\mathrm{KL}(p, q)$ as in the traditional neural posteriors. Simulation-based calibration (Cook et al., 2006; Talts et al., 2018; Modrák et al., 2023) has examined the marginal rank statistics for testing, while we use it for training. Under the repeated prior sampling and correct computation, Bayesian models are calibrated (Dawid, 1982). Our computation calibration should not be confused with the frequentist calibration (repeated data sampling under one true parameter, Masserano et al., 2023). The sample-based stacking is relevant to the discriminative calibration (Yao and Domke, 2023) and the adversarial training (Ramesh et al., 2022). The moment matching shares a similar objective with the moment calibration (Yu et al., 2021) and moment network (Jeffrey and Wandelt, 2020), while our new objective combines two moments. Our paper differs from these existing tools in that we combine multiple inferences.

Limitations and future directions This paper develops a stacking strategy to combine multiple simulation-based inferences for the same task. We design stacking to incorporate various combination forms and utility functions for distributional approximation.

Our stacking utility function is averaged over y, which computes the averaged approximation quality. We have discussed the possibility of local weights, but more evaluation is needed in the future.

Including stacking in the inference pipeline provides double robustness. If individual inferences are accurate enough, there is no need for stacking; If the posterior stacking model is flexible enough, individual inferences can be arbitrarily off. In the experiments we tested the individual inferences are well-constructed, while the stacking model is relatively simple with a relatively negotiable computation cost. Looking forward, with advances in multiple-data processors such as GPUs, it is faster to run a large number of crude approximations in parallel than to optimize one single run to full precision, making it appealing to use a comprehensive stacking model to combine many cheaper inferences, which we leave for future work.

Acknowledgements

The authors thank Andrew Gelman and Andreas Buja for helpful discussions.

References

Agrawal, A. and Domke, J. (2021). Amortized variational inference for simple hierarchical models. *Advances in Neural Information Processing Systems*, 34.

Breiman, L. (1996). Stacked regressions. *Machine Learning*, 24:49–64.

Clyde, M. and Iversen, E. S. (2013). Bayesian model averaging in the M-open framework. In *Bayesian Theory and Applications*, pages 483–498. Oxford University Press.

Cook, S. R., Gelman, A., and Rubin, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons, 2rd edition.

Cramér, H. (1928). On the composition of elementary errors. *Scandinavian Actuarial Journal*, 1:13–74.

Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of National Academy of Sciences*, 117(48):30055–30062.

Dawid, A. P. (1982). The well-calibrated Bayesian. *Journal of American Statistical Association*, 77(379):605–610.

Dax, M., Green, S. R., Gair, J., Macke, J. H., Buonanno, A., and Schölkopf, B. (2021). Real-time gravitational wave science with neural posterior estimation. *Physical Review Letters*, 127(24):241103.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. *Advances in Neural Information Processing Systems*, 32.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of American Statistical Association*, 102(477):359–378.

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

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

Hahn, C., Eickenberg, M., Ho, S., Hou, J., Lemos, P., Massara, E., Modi, C., Moradinezhad Dizgah, A., Régaldo-Saint Blancard, B., and Abidi, M. M. (2023). A forward modeling approach to analyzing galaxy clustering with SIMBIG. *Proceedings of National Academy of Sciences*, 120(42):e2218810120.

Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial. *Statistical Science*, 14(4):382–417.

Hoffman, M., Radul, A., and Sountsov, P. (2021). An adaptive-MCMC scheme for setting trajectory lengths in Hamiltonian Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*.

Jeffrey, N. and Wandelt, B. D. (2020). Solving high-dimensional parameter inference: marginal posterior densities & moment networks. In *NeurIPS Workshop on Machine Learning and the Physical Sciences*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Le, T. and Clarke, B. (2017). A Bayes interpretation of stacking for $\mathcal{M}$-complete and $\mathcal{M}$-open settings. *Bayesian Analysis*, 12:807–829.

Lueckmann, J.-M., Boelts, J., Greenberg, D., Goncalves, P., and Macke, J. (2021). Benchmarking simulation-based inference. In *International Conference on Artificial Intelligence and Statistics*.

Masserano, L., Dorigo, T., Izbicki, R., Kuusela, M., and Lee, A. B. (2023). Simulation-based inference with waldo: Perfectly calibrated confidence regions using any prediction or posterior estimation algorithm. In *International Conference on Artificial Intelligence and Statistics*.

Matheson, J. E. and Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096.

Mesquita, D., Blomstedt, P., and Kaski, S. (2020). Embarrassingly parallel MCMC using deep invertible transformations. In *Uncertainty in Artificial Intelligence*, pages 1244–1252.

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. *Bayesian Analysis*, 1(1):1–28.

Nemeth, C. and Sherlock, C. (2018). Merging MCMC subposteriors through gaussian-process approximations. *Bayesian Analysis*.

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

Papamakarios, G., Sterratt, D., and Murray, I. (2019). Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *International Conference on Artificial Intelligence and Statistics*.

Ramesh, P., Lueckmann, J.-M., Boelts, J., Tejero-Cantero, Á., Greenberg, D. S., Gonçalves, P. J., and Macke, J. H. (2022). GATSBI: Generative adversarial training for simulation-based inference. In *International Conference on Learning Representations*.

Régaldo-Saint Blancard, B., Hahn, C., Ho, S., Hou, J., Lemos, P., Massara, E., Modi, C., Moradinezhad Dizgah, A., Parker, L., Yao, Y., and Eickenberg, M. (2023). SimBIG: Galaxy clustering analysis with the wavelet scattering transform. *arXiv:2310.15250*.

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., and Gelman, A. (2018). Validating Bayesian inference algorithms with simulation-based calibration. *arXiv:1804.06788*.

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., and Macke, J. H. (2020). SBI: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505.

van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press, Cambridge.

Vehtari, A. and Ojanen, J. (2012). A survey of bayesian predictive methods for model assessment, selection and comparison. *Statistical Survey*.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241–259.

Yao, Y. and Domke, J. (2023). Discriminative calibration. *arXiv:2305.14593*.

Yao, Y., Ouyang, J., and Buja, A. (2024). Stacking as a way of life: A general framework for combining predictive distributions. Technical report.

Yao, Y., Vehtari, A., and Gelman, A. (2022). Stacking for non-mixing Bayesian computations: The curse and blessing of multimodal posteriors. *Journal of Machine Learning Research*, 23(1):3426–3471.

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

Yu, X., Nott, D. J., Tran, M.-N., and Klein, N. (2021). Assessment and adjustment of approximate inference algorithms using the law of total variance. *Journal of Computational and Graphical Statistics*, 30(4):977–990.

Zhao, R. (2023). The generalized multiplicative gradient method and its convergence rate analysis. *arXiv: 2207.13198*.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [NA]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets [NA], check if you include:
   (a) Citations of the creator If your work uses existing assets. [NA]
   (b) The license information of the assets, if applicable. [NA]
   (c) New assets either in the supplemental material or as a URL, if applicable. [NA]
   (d) Information about consent from data providers/curators. [NA]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [NA]

5. If you used crowdsourcing or conducted research with human subjects [NA], check if you include:
   (a) The full text of instructions given to participants and screenshots. [NA]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [NA]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [NA]

Appendices to "Simulation Based Stacking"

A Background

Simulation-based inference (SBI). In a typical Bayesian inference setting, we are interested in the posterior inference of parameter $\theta \in \mathbb{R}_d$ given observed data y. The ultimate goal is to infer a version of the condition distribution $p(\theta|y)$ and/or sample from it. In simulation-based inference, we cannot evaluate the likelihood, but we can easily simulate outcomes from the data model $y|\theta$. We create a size-N joint simulation table $\{(\theta_n, y_n)\}_{n=1}^N$ by first drawing parameters θ_n from the prior distribution $p(\theta)$ and one realization of data y_n from the data-forward model $p(y|\theta_n)$.

Normalizing flow. The posterior inference task becomes a conditional density estimation task. In one neural posterior estimation, we parameterize the posterior density as a normalizing flow $q_\beta(\theta|y)$, where $\beta \in \mathbb{R}_m$ is the normalizing flow parameter. More concretely, let z be a multivariate standard Gaussian random variable in $\mathbb{R}_d$, consider $\theta = f_{\beta,y}(z)$, a bijective mapping from $z \in \mathbb{R}_d$ to $\theta \in \mathbb{R}_d$. Let $z = F_{\beta,y}(\theta)$ be the inverse of this mapping. From change-of-variable, the implied distribution of θ is $q_\beta(\theta|y) = p_{\text{MVN}}(z = F_{\beta,y}(\theta))|\det(\frac{d}{d\theta}F_{\beta,y}(\theta))|$, where p_{MVN} denotes the density of standard Gaussian. When the bijective $f_{\beta,y}$ is flexible enough, in principle, the family of derived densities $\{q_\beta(\theta|y) : \beta \in \mathbb{R}_m\}$ covers all smooth conditional distributions on $\mathbb{R}_d$.

Neural posterior estimation. Given any β, $q_\beta(\theta|y)$ is ensured to be a normalized conditional density on Θ by design: $q_\beta(\theta|y) \geq 0$ and $\int_\Theta q_\beta(\theta|y)d\theta = 1$. Since we have simulations from the joint density $p(\theta, y)$, we fit this normalizing flow to minimize the KL divergence:

$$\hat{\beta} = \arg\max_{\beta \in \mathbb{R}_m} \sum_{i=1}^N \log q_\beta(\theta_i|y_i).$$

This inferred $q_{\hat{\beta}}(\theta|y)$ is one neural posterior estimation. We are able to (a) evaluate the density $q_{\hat{\beta}}(\theta|y)$ for any (θ, y) pair, and (b) given any y, sample IID draws $\tilde{\theta}_1, \ldots, \tilde{\theta}_S$ from $q_{\hat{\beta}}(\theta|y)$.

By varying the normalizing flow architecture or hyperparameters, we obtain multiple neural posterior estimations $\{q_1(\theta|y), q_2(\theta|y), \ldots, q_K(\theta|y)\}$. This present paper aims to aggregate them to provide better inference.

B Additional Theory and Proof

Recap of the general SBI stacking setting. Consider a parameter space $\Theta = \mathbb{R}_d$ with the usual Borel measure, and any measurable data space $\mathcal{Y}$. We are given a sample of N IID draws $\{(\theta_i, y_i) : 1 \leq i \leq N\}$ from a joint distribution $p(\theta, y)$ on the product space $\Theta \times \mathcal{Y}$. Let $p(\theta|y)$ be a version of the true conditional density. We are given $K \geq 2$ conditional densities $q_k(\theta|y)$. Besides, for each inference index $k \leq K$ and simulation index $n \leq K$, we have obtained an IID sample of S draws: $\tilde{\theta}_{kn1}, \ldots, \tilde{\theta}_{knS} \sim q_k(\theta|y_n)$. We always denote k the index of inference, n the index of simulations, and s the index of the posterior draw. See Figure 3 for visualization.

Typically we have a training-validation spilt to avoid over-fitting. For the theory part, it suffices to assume that the K conditional densities are fixed, and will not change as N increases.

B.1 General posterior stacking (Proposition 1)

We give a formal definition of Proposition 1 in the paper.

In the general posterior stacking, we specify a combination form, and an objective function of distributional approximation. Suppose we can evaluate the score of the stacked distribution, $U(q^*_{\mathbf{w}}(\cdot|y_n), \theta_n)$. Further if the scoring rule U is proper (16). We combine K posterior inferences with the form (14), and fit stacking parameters $\mathbf{w}$ via a sample M-estimate,

$$\hat{\mathbf{w}} = \arg\max \sum_{n=1}^N U(q^*_{\mathbf{w}}(\cdot|y_n), \theta_n). \tag{18}$$

The expectation $\mathbb{E}_{p(y,\theta)} U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta)$ is the average utility function of the stacked posterior.

Proposition 1. *Clause I (population utility). For any given* $\mathbf{w}$*, as* $N \to \infty$*,*

$$\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\mathbf{w}}(\cdot|y_n), \theta_n) = \mathbb{E}_{p(y,\theta)}\, U(q^*_{\mathbf{w}}(\cdot|y), \theta) + o_p(1).$$

Clause II (asymptotic optimality). For any $\epsilon > 0$*, any given* $\mathbf{w}'$*, as* $N \to \infty$*,*

$$\Pr\left(\mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta) \le \mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}'}(\cdot|y), \theta) + \epsilon\right) \to 1.$$

Clause III (convergence). Further assume that (a) the support of $\mathbf{w}$ *is compact, (b) there is a true* $\mathbf{w}_0$*, such that* $q^*_{\mathbf{w}_0}(\cdot|y) = p(\theta|y)$ *almost everywhere, (c) the combination is locally identifiable at the truth, i.e., if* $q^*_{\mathbf{w}}(\cdot|y) = p(\theta|y)$ *then* $\mathbf{w} = \mathbf{w}_0$*, and (d) the scoring rule* U *is strictly proper, then the stacking weight estimate is consistent as* $N \to \infty$*,*

$$\hat{w} = w_o + o_p(1).$$

Proof. We briefly sketch the proof since similar proofs have appeared in previous propositions.

Clause I is from the weak law of large numbers.

Clause II addresses the optimality of the resulting divergence metric rather than the estimated $\hat{\mathbf{w}}$. From WLLN, for any $\epsilon > 0$, as $N \to \infty$,

$$\Pr\left(\left|\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\hat{w}}(\cdot|y_n), \theta_n) - \mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta)\right| \ge 1/2\epsilon\right) = o(1).$$

$$\Pr\left(\left|\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\mathbf{w}'}(\cdot|y_n), \theta_n) - \mathbb{E}_{p(y,\theta)}\, U(q^*_{\mathbf{w}'}(\cdot|y), \theta)\right| \ge 1/2\epsilon\right) = o(1).$$

Furthermore, by definition,

$$\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\mathbf{w}'}(\cdot|y_n), \theta_n) \le \frac{1}{N}\sum_{n=1}^{N} U(q^*_{\hat{\mathbf{w}}}(\cdot|y_n), \theta_n).$$

Combine these three lines, we have

$$\begin{aligned}
&\Pr\left(\mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta) \le \mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}'}(\cdot|y), \theta) + \epsilon\right) \\
&= \Pr\left(\mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta) \le \mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}'}(\cdot|y), \theta) + \epsilon, \quad \frac{1}{N}\sum_{n=1}^{N} U(q^*_{\mathbf{w}'}(\cdot|y_n), \theta_n) \le \frac{1}{N}\sum_{n=1}^{N} U(q^*_{\hat{\mathbf{w}}}(\cdot|y_n), \theta_n)\right) \\
&\le \Pr\left(\left|\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\hat{w}}(\cdot|y_n), \theta_n) - \mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta)\right| + \left|\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\mathbf{w}'}(\cdot|y_n), \theta_n) - \mathbb{E}_{p(y,\theta)}\, U(q^*_{\mathbf{w}'}(\cdot|y), \theta)\right| \ge \epsilon\right) \\
&\le \Pr\left(\left|\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\hat{w}}(\cdot|y_n), \theta_n) - \mathbb{E}_{p(y,\theta)}\, U(q^*_{\hat{\mathbf{w}}}(\cdot|y), \theta)\right| \ge 1/2\epsilon\right) \\
&+ \Pr\left(\left|\frac{1}{N}\sum_{n=1}^{N} U(q^*_{\mathbf{w}'}(\cdot|y_n), \theta_n) - \mathbb{E}_{p(y,\theta)}\, U(q^*_{\mathbf{w}'}(\cdot|y), \theta)\right| \ge 1/2\epsilon\right) \\
&= o(1),
\end{aligned}$$

which proves Clause II.

Clause III is a direct application of Lemma 1. Here we assume a compact $\mathbf{w}$ space to ensure uniform convergence. □

B.2 Convergence in mixture stacking (Prop. 2)

First, we consider mixture stacking with the log density objective. Given a simplex weight $\mathbf{w} \in \mathbb{S}_K$, The mixed posterior density (1) is $q^{\text{mix}}_{\mathbf{w}}(\theta|y) \coloneqq \sum_{k=1}^K w_k q_k(\theta|y)$. It is straightforward to derive the well-known correspondence between the log density and the Kullback–Leibler (KL) divergence.

Proposition 2. *Clause (I). For any $\mathbf{w} \in \mathbb{S}_K$, as $N \to \infty$, the stacking objective converges in distribution to the negative conditional KL divergence between the true posterior and stacked approximation, i.e.,*

$$\frac{1}{N}\sum_{i=1}^N \log q^{\text{mix}}_{\mathbf{w}}(\theta_i|y_i) + \text{KL}(p(\theta|y), q^{\text{mix}}_{\mathbf{w}}(\theta|y)) - C = o_p(1),$$

where $C = \log p(\theta|y)p(\theta, y)$ is a constant that does not depends on $\mathbf{w}$ or q.

Clause (II). If (a) there exists a $\mathbf{w}_0 \in \mathbb{S}^K$, such that $p(\theta|y) = q^{\text{mix}}_{\mathbf{w}_0}(\theta|y)$ almost everywhere, and (b) the mixture form (1) is locally identifiable at the truth, i.e., if there is there is a $\mathbf{w}'$ such that $q^{\text{mix}}_{\mathbf{w}'}(\theta|y) = q^{\text{mix}}_{\mathbf{w}_0}(\theta|y)$ almost everywhere, then $\mathbf{w}' = \mathbf{w}_0$. Then the optimal stacking weight

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}\in\mathbb{S}_K} \frac{1}{N}\sum_{i=1}^N \log q^{\text{mix}}_{\mathbf{w}}(\theta_i|y_i)$$

converges to the true $\mathbf{w}_0$ in probability, i.e., for any $\epsilon > 0$,

$$\Pr_p(|\hat{\mathbf{w}} - \mathbf{w}_0\,| \geq \epsilon) \to 0, \quad \text{as } N \to \infty.$$

Proof. Clause (I) is a direct application of the weak law of large numbers. Since $\{(\theta_i, y_i)\}$ are IID draws from the joint, in probability we have

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^N \log q^{\text{mix}}_{\mathbf{w}}(\theta_i|y_i) &\xrightarrow{p} \int_{\Theta\times\mathcal{Y}} \log q^{\text{mix}}_{\mathbf{w}}(\theta|y)p(\theta, y)d\theta dy \\
&= \int_{\Theta\times\mathcal{Y}} q^{\text{mix}}_{\mathbf{w}}(\theta|y)p(\theta|y)p(y)d\theta dy \\
&= -\int_{\Theta\times\mathcal{Y}} \left(\log p^{\text{mix}}_{\mathbf{w}}(\theta|y) - \log q^{\text{mix}}_{\mathbf{w}}(\theta|y)\right) p(\theta|y)p(y)d\theta dy + \int \log p^{\text{mix}}_{\mathbf{w}}(\theta|y)p(\theta, y)d\theta dy \\
&= -\,\text{KL}(p(\theta|y), q^{\text{mix}}_{\mathbf{w}}(\theta|y)) + C.
\end{aligned}$$

□

Clause (II) is a consequence of the convergence of the maximum likelihood estimation (MLE). In order to apply to other proportions, we state the general M-estimation theory. For example, the following lemma is from van der Vaart (1998).

Lemma 1. *Assuming $y_1, \dots y_N$ are IID data from $p_{\beta_0}(y)$. Let M_n be random functions and let M be a fixed function of the parameter β such that for every $\epsilon > 0$,*

$$\sup_\beta |M_n(\beta) - M(\beta)| \xrightarrow{\text{P}} 0, \tag{19}$$

$$\sup_{\beta: d(\beta,\beta_0)\geq\epsilon} M(\beta) < M(\beta_0). \tag{20}$$

Then any sequence of estimators $\hat{\beta}_n$ with

$$M_n(\hat{\beta}) \geq M_n(\beta_0) - o_p(1)$$

converges in probability to β_0.

In the context of Clause (II), the stacking parameter $\mathbf{w}$ is on a compact space $\mathbb{S}_K$. Because (a) the identifiable assuming and (b) the unique minimizer to $\mathbb{E}_p \log q(x)$ is $p = q$, the true weight $\mathbf{w}_0$ is the unique minimize of $\int_{\Theta\times\mathcal{Y}} q^{\text{mix}}_{\mathbf{w}}(\theta|y)p(\theta|y)p(y)d\theta dy$, i.e., for any $\epsilon > 0$, and any $\mathbf{w}'$ such that $|| \mathbf{w}_0 - \mathbf{w}' || > \epsilon$, we have

$$\int_{\Theta\times\mathcal{Y}} q^{\text{mix}}_{\mathbf{w}_0}(\theta|y)p(\theta|y)p(y)d\theta dy > \int_{\Theta\times\mathcal{Y}} q^{\text{mix}}_{\mathbf{w}'}(\theta|y)p(\theta|y)p(y)d\theta dy,$$

which verifies condition (20), while the WLLN ensures the uniform convergence condition (19). Applying Lemma 1 proves Clause II.

B.3 Rank-based calibration and stacking (Prop. 3 & 4)

We now deal with rank-based divergence and stacking. We assume $\Theta = \mathbb{R}$ for the next two propositions since we only compute marginal ranks.

Proposition 3. *The rank-based metric $D_{\text{rank}}(p, q)$ defined in (4) is a generalized divergence: (I) $D_{\text{rank}}(p, q) \geq 0$ for any p and q. (II) When $q(\theta|y) = p(\theta|y)$ almost everywhere, $D_{\text{rank}}(p, q) = 0$.*

Proof. From the definition of $D_{\text{rank}}(p, q)$, let $(\theta, y) \sim p(\theta, y)$ be a pair of random variables from p, and let $x_1 = F_p(\theta|y)$ and $x_2 = F_q(\theta|y)$ be two transformed random variables, then $D_{\text{rank}}(p, q) = D_0(x1, x2)$. Because D_0 is a well-defined divergence, $D_{\text{rank}}(p, q) = D_0(x1, x2) \geq 0$ for any p and q. If $p = q$, then x_1 and x_2 have the same distribution and hence $D_{\text{rank}}(p, q) = D_0(x1, x2) = 0$. □

$D_{\text{rank}}(p, q)$ is CDF-based. Its empirical estimates use rank only. It might be unsatisfactory that $D_{\text{rank}}(p, q)$ is not a strict divergence, meaning that there can be a distinct pair of joint distributions $p \neq q$, such that $D_{\text{rank}}(p, q) = 0$. Indeed, for any $p(\theta, y)$, let $q(\theta|y) \coloneqq p(\theta)$, then $D_{\text{rank}}(p, q) = 0$. This edge case is a well-known example where the traditional rank-based calibration is not sufficient and can lead to false-negative testing. That said, the rank-based divergence has the advantage that it is rank/sample only; no density evaluation of q is needed. We find that the rank-based divergence is particularly powerful when augmented with other density-based divergences.

Let us briefly recap the rank-related definitions. We still consider the mixture stacking form (1): given a simplex weight $\mathbf{w} \in \mathbb{S}_K$, the mixed posterior density is $q^{\text{mix}}_{\mathbf{w}}(\theta|y) \coloneqq \sum_{k=1}^K w_k q_k(\theta|y)$. We assume all conditional densities $q_k(\theta|y)$ are continuous on Θ. Let r_{kn} is the rank statistics defined in (3): $r_{kn} \coloneqq \frac{1}{S}\sum_{s=1}^S \mathbb{1}\{\theta_n \geq \tilde{\theta}_{kns}\}$.

Proposition 4. *Clause (I). For any fixed value $\mu \in \Theta$, and any weight $\mathbf{w} \in \mathbb{S}_K$, let*

$$F_{k,y}(\mu) \coloneqq \Pr_{\theta\sim q_k(\theta|y)}(\theta < \mu), \quad \text{and} \quad F^*_y(\mu) \coloneqq \Pr_{\theta\sim q^{\text{mix}}_{\mathbf{w}}}(\theta < \mu)$$

be the CDFs in the individual conditional distributions and the mixture, then the CDF remains linear:

$$F^*_y(\mu) = \sum_{k=1}^K w_k F_{k,y}(\mu). \tag{21}$$

Clause (II). For any simulation index n, define

$$r^{\text{mix}}_n \coloneqq \sum_{k=1}^K w_k r_{kn}$$

be the stacked rank, then for any weight $\mathbf{w} \in \mathbb{S}_K$, this linearly additive rank converges to the mixture CDF:

$$r^{\text{mix}}_n \to \Pr_{\theta\sim q^{\text{mix}}_{\mathbf{w}}(\theta|y_n)}(\theta \leq \theta_n), \quad \text{almost surely}, \quad \text{as} \quad S \to \infty.$$

Proof. Clause I is due to the integral linearity. The CDFs are integrals on a fixed half interval:

$$F_{k,y}(\mu) = \Pr_{\theta\sim q_k(\theta|y)}(\theta < \mu) = \int_{-\infty}^{\mu} q_k(\theta|y)d\theta,$$

Likewise,

$$\begin{aligned} F_y^*(\mu) &= \int_{-\infty}^{\mu} q_k(\theta|y) q_{\mathbf{w}}^{\text{mix}}(\theta|y) d\theta \\ &= \int_{-\infty}^{\mu} \left(q_k(\theta|y) \sum_{k=1}^{K} w_k q_k(\theta|y) \right) d\theta \\ &= \sum_{k=1}^{K} w_k \left(\int_{-\infty}^{\mu} q_k(\theta|y) d\theta \right) \\ &= \sum_{k=1}^{K} w_k F_{k,y}(\mu), \end{aligned}$$

which proves Clause 1.

Clause 2 states the convergence of the mixed ranks. For any fixed k and n, because $\tilde{\theta}_{kns}$ are IID draws from $q_k(\theta|y_n)$, the strong law of large numbers applies:

$$r_{kn} = \frac{1}{S} \sum_{s=1}^{S} \mathbb{1}\{\theta_n \geq \tilde{\theta}_{kns}\} \to \Pr_{\theta \sim q_k(\theta|y_n)}(\theta \leq \theta_n), \quad \text{almost surely.}$$

By an elementary probability lemma of the sum of almost surely convergent (Lemma 2), the mixed rank converges almost surely as well,

$$\begin{aligned} r_n^{\text{mix}} = \sum_{k=1}^{K} w_k r_{kn} &\to \sum_{k=1}^{K} w_k \Pr_{\theta \sim q_k(\theta|y_n)}(\theta \leq \theta_n), \quad \text{almost surely,} \\ &= \Pr_{\theta \sim q_{\mathbf{w}}^{\text{mix}}(\theta|y_n)}(\theta \leq \theta_n), \end{aligned}$$

where the last equality is from Equation (21). □

Lemma 2. *Let $\{X_n\}$ and $\{Y_n\}$ be two sequences of random variables. If X_n converges almost surely to X and Y_n converges almost surely to Y, then the sum $X_n + Y_n$ converges almost surely to $X + Y$.*

B.4 Sample-based stacking for JS divergence (Prop. 5)

In sample-stacking, we aggregate individual inferences in (7). $\theta_{ns}^* = \mathbf{w}_0 + \mathbf{w}_1 \tilde{\theta}_{1ns} + \cdots + \mathbf{w}_K \tilde{\theta}_{Kns}$ is the aggregated inference sample. For any given $\mathbf{w}$ and any y, we define the distribution of the sample-aggregated inference as follows. Let $\tilde{\theta}_k$ be a random variable draws from $q_k(\theta|y)$, these $\tilde{\theta}_k$ are mutually independent, and denote $q_{\mathbf{w}}^*(\theta|y)$ be law of the random variable $\mathbf{w}_0 + \mathbf{w}_1 \tilde{\theta}_1 + \cdots + \mathbf{w}_K \tilde{\theta}_{Kns}$.

From each simulation draw

$$(\theta_n, y_n, \theta_{n1}^*, \theta_{n2}^* \ldots, \theta_{nS}^*),$$

we generate $(S+1)$ classification examples with feature ϕ and label z:

$$\begin{aligned} z = 1, \ \phi = (\theta_n, y_n) \\ z = 0, \ \phi = (\theta_{n1}^*, y_n) \\ \cdots \\ z = 0, \ \phi = (\theta_{nS}^*, y_n) \end{aligned}$$

Let $f(z = 1|\phi) = \Pr(z = 1|\phi, \text{some classifier})$ be any classification probability prediction that uses ϕ to predict z. Let $\mathcal{F}$ be the space of all such binary classifiers. To balance the two classes, we typically use a weighted classification utility function

$$U(z, \phi, f) = \frac{C}{S+1} \mathbb{1}(z = 1) \log f(z = 1|\phi) + \frac{CS}{S+1} \mathbb{1}(t = 0) \log f(z = 0|\phi),$$

where $C = \frac{(S+1)^2}{2S}$ is a normalizing constant.

Sample stacking solves a mini-max optimization.

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \max_{f \in \mathcal{F}} \sum_{i=1}^{N(S+1)} U(z_i, \phi_i, f). \tag{22}$$

Proposition 5. *Clause (I). For any fixed* $\mathbf{w}$ *and fixed* $S \geq 1$*, as* $N \to \infty$*, the best classifier utility corresponds to the conditional Jensen Shannon divergence between the aggregated inference and the truth,*

$$\max_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N(S+1)} U(z_i, \phi_i, f) = \mathbb{E}_y \left[\frac{1}{2} p(\theta|y) \log \frac{p(\theta|y)}{r(\theta|y)} + \frac{1}{2} q^*_{\mathbf{w}}(\theta|y) \log \frac{p(\theta|y)}{r(\theta|y)} \right] - \log 2 + o_p(1),$$

where

$$r(\theta|y) \coloneqq \frac{1}{2} \left(p(\theta|y) + q^*_{\mathbf{w}}(\theta|y) \right).$$

Clause (II). Let $\hat{w}$ *be the solution from sample stacking (22). Assuming (a) there exists one true* $\mathbf{w}_0$ *such that* $q^*_{\mathbf{w}}(\theta|y) = p(\theta|y)$*, and (b) the sample model is locally identifiable at the truth, i.e., if* $q^*_{\mathbf{w}'}(\theta|y) = p(\theta|y)$*, then* $\mathbf{w}' = \mathbf{w}_0$*. Then for any fixed* S*, as* $N \to \infty$*, the stacking weight estimate is consistent in probability,*

$$\hat{\mathbf{w}} = \mathbf{w}_0 + o_p(1).$$

Clause (I) is similar to the standard adversarial learning, (e.g., Theorem 8 in Yao and Domke, 2023). Clause (II) can be proved from Lemma 1. It is a direct consequence of the main proposition.

B.5 Interval stacking (Prop. 6)

In interval stacking, we run stacking to solve

$$\min_{\mathbf{w}} \sum_{i=1}^{N} U((r^*_n, l^*_n), \theta_n), \qquad U((r^*_n, l^*_n), \theta_n) \coloneqq (r^*_n - l^*_n) + \frac{2}{\alpha}(l^*_n - \theta_n)\mathbb{1}(\theta_n < l^*_n) + \frac{2}{\alpha}(\theta_n - r^*_n)\mathbb{1}(\theta_n > r^*_n).$$

Proposition 6. *Clause (I). For any fixed confidence-level* $\alpha \in (0,1)$*, and for any* y*, let* $r_{k,y}$ *and* $l_{k,y}$ *be the* $\frac{\alpha}{2}$ *and* $(1 - \frac{\alpha}{2})$ *quantile of the distribution* $q_k(\theta|y)$*. Given a weight* $\mathbf{w}$*,* $l^*_{\mathbf{w},y} = \sum_{k=1}^{K} w_k l_{k,y}$ *and* $r^*_{\mathbf{w},y} = \sum_{k=1}^{K} \mathbf{w}_{k+K}\, r_{k,y}$ *are the stacked left and right confidence interval endpoint. For any fixed* $\mathbf{w}$ *and as* $N \to \infty$*,*

$$\frac{1}{N} \sum_{i=1}^{N} U((r^*_n, l^*_n), \theta_n) - \mathbb{E}_{p(\theta,y)}\, U(r^*_{\mathbf{w},y}, l^*_{\mathbf{w},y}, \theta) = o_p(1). \tag{23}$$

Clause (II). Assuming that $p(\theta|y) > 0$ *for* $\theta \in \Theta$ *and* y *almost everywhere. For any fixed confidence-level* $\alpha \in (0,1)$*, let* $l_p(y), r_p(y)$ *be the* $\frac{\alpha}{2}$ *and* $(1 - \frac{\alpha}{2})$ *quantile of the true posterior distribution* $p(\theta|y)$*, they are the unique minimize to the population limit in (23). That is, for any two functions* $l(y), r(y)$ *that maps* y *to the* Θ *space,*

$$\mathbb{E}_{p(\theta,y)}\, U(r_p(y), l_p(y), \theta) \leq \mathbb{E}_{p(\theta,y)}\, U(r(y), l(y), \theta),$$

and the equality holds if and only if $r_p(y) = r(y), l_p(y) = l(y)$*, almost everywhere.*

Clause I is from the weak law of large numbers. To prove Clause II, we first state the following Lemma (adapted from Theorem 6 in Gneiting and Raftery, 2007), which addresses a single distribution (no dependence on y).

Lemma 3. *Let* p *be a continuous distribution on* $\Theta = \mathbb{R}$*, and* $p(\theta) > 0$ *for any* θ*. If* s *is a strictly increasing function, and given a fixed confidence-level* $\alpha \in (0,1)$*, then the scoring rule*

$$S(r; \theta) = \alpha s(r) + (s(\theta) - s(r))\mathbb{1}(\theta \leq r).$$

is proper for predicting the quantile of p *at level* α*.*

Proof. The proof of the lemma is also adapted from Gneiting and Raftery (2007). Let μ be the unique α-quantile of p. For any $r < \mu$,

$$\begin{aligned}\mathbb{E}_{\theta\sim p(\theta)}\, S(\mu;\theta) - \mathbb{E}_{\theta\sim p(\theta)}\, S(r;\theta) &= \int_r^{\mu} s(\theta)p(\theta)d\theta + s(r)P(r) - \alpha s(r)\\ &> s(r)(p(\mu) - p(r)) + s(r)p(r) - \alpha s(r)\\ &= 0.\end{aligned}$$

Likewise, for any $r < \mu$, $\mathbb{E}_{\theta\sim p(\theta)}\, S(q;\theta) - \mathbb{E}_{\theta\sim p(\theta)}\, S(r;\theta) > 0$. □

Let $s(x) = x$ and apply this lemma twice, then for any distribution p on $\Theta = \mathbb{R}$, and a fixed confidence level α, suppose μ_l, μ_r are the $\alpha/2$ and $1 - \alpha/2$ quantile of p, it is clear that

$$\mathbb{E}_{\theta\sim p(\theta)}\, U(\mu_r, \mu_l\theta) - \mathbb{E}_{\theta\sim p(\theta)}\, S(\mu_r', \mu_l';\theta) \le 0.$$

The equality holds if and only $\mu_r' = \mu_r, \mu_l' = \mu_l$.

To prove the second clause of Prop. 6, note that $\mathbb{E}_{p(\theta,y)}\, U(r_p(y), l_p(y), \theta) - \mathbb{E}_{p(\theta,y)}\, U(r(y), l(y), \theta) = \mathbb{E}_y\left(\mathbb{E}_{p(\theta|y)}\, U(r_p(y), l_p(y), \theta) - \mathbb{E}_{p(\theta|y)}\, U(r(y), l(y), \theta)\right)$. Because for any given y, $\mathbb{E}_{p(\theta|y)}\, U(r_p(y), l_p(y), \theta) - \mathbb{E}_{p(\theta|y)}\, U(r(y), l(y), \theta) \le 0$ due to the previous lemma, then $\mathbb{E}_{p(\theta,y)}\, U(r_p(y), l_p(y), \theta) - \mathbb{E}_{p(\theta,y)}\, U(r(y), l(y), \theta) \le 0$, and the equality holds if any only if $r_p(y) = r(y), l_p(y) = l(y)$ almost everywhere with respect to $p(y)$.

B.6 Moment stacking (Prop. 7)

Proposition 7. *For any y, let $\mu_k(y)$ and $V_k(y)$ be the mean and covariance of the k-th approximate posterior $q_k(\theta|y)$. Given a weight $\mathbf{w}$, let $V_y^*(\mathbf{w})$ and $\mu_y^*(\mathbf{w})$ be the covariance and mean in the $\mathbf{w}$-mixed posterior $\sum_{k=1}^K q_k(\theta|y)$, as defined in subsection 2.5. Then for any fixed $\mathbf{w}$ and $N \to \infty$,*

$$\frac{1}{N}\sum_{n=1}^{N}\left(\log\det V_n^*(\mathbf{w}) + ||\mu_n^*(\mathbf{w}) - \theta_n||^2_{(V_n^*(\mathbf{w}))^{-1}}\right) = \mathbb{E}_{p(\theta,y)}\left(\log\det V_y^*(\mathbf{w}) + ||\mu_y^*(\mathbf{w}) - \theta||^2_{(V_y^*(\mathbf{w}))^{-1}}\right) + o_p(1).$$

Let $\mu(y)$ and $V(y)$ be the true mean and variance of the posterior $p(\theta|y)$, then for any $\mathbf{w}$,

$$\mathbb{E}_{p(\theta,y)}\left(\log\det V(y) + ||\mu(y) - \theta||^2_{V^{-1}(y)}\right) \le \mathbb{E}_{p(\theta,y)}\left(\log\det V_y^*(\mathbf{w}) + ||\mu_y^*(\mathbf{w}) - \theta||^2_{(V_y^*(\mathbf{w}))^{-1}}\right),$$

and the equality holds if and only if $V_y^*(\mathbf{w}) = V(y)$ and $V_y^*(\mathbf{w}) = \mu(y)$ almost everywhere with respect to $p(y)$, if attainable.

The proof is very similar to Prop. 6, requiring one application of the WLLN, and to verify that the underlying score is proper. We omit the details here.

The next proposition states that any mixture of a list of pointwisely under-confident approximations will remain under-confident.

Proposition 8. *For a fixed y, if for any k,*

$$\mathrm{Var}_{q_k}(\theta|y) \ge \mathrm{Var}_p(\theta|y),$$

then for any weight $\mathbf{w} \in \mathbb{S}_K$, the variance in the mixture is always under-confident: k,

$$\mathrm{Var}_{p_{\mathbf{w}}^{\mathrm{mix}}}(\theta|y) \ge \mathrm{Var}_p(\theta|y).$$

Proof. Use the law of total variance, for any fixed $\mathbf{w}$,

$$\begin{aligned}
\mathrm{Var}_{p^{\mathrm{mix}}_{\mathbf{w}}}(\theta|y) &= \mathrm{Var}\,\mathbb{E}_{q_k}(\theta|y) + \mathbb{E}\,\mathrm{Var}_{q_k}(\theta|y) \\
&\geq \mathbb{E}\,\mathrm{Var}_{q_k}(\theta|y) \\
&= \sum_{k=1}^{K} w_k \mathrm{Var}_{q_k}(\theta|y) \\
&\geq \min_k \mathrm{Var}_{q_k}(\theta|y) \\
&\geq \mathrm{Var}_p(\theta|y).
\end{aligned}$$

□

C Practical Implementation

C.1 Smooth approximation of indicator functions

We approximate indicators functions using an infinitely differentiable approximation of the Heaviside step function $H_\varepsilon(x) = (1 + \exp(-2\varepsilon x))^{-1}$ for a given ε value. In practice, we select an ε value that is sufficiently small compared to typical evaluation points of H_ε.

In particular, in the context of Sect. 2.2, we choose $\varepsilon = 1/100$. In the context of Sect. 2.4, we choose $\varepsilon = (\min_n(r_n - l_n)/1,000$ where $\min_n(r_n - l_n)$ is the minimum interval length over the training set.

C.2 Quasi Monte Carlo sampling from the stacked posterior

Given weights w_k, and K simulation draws $\{\theta_{ks}\} \sim q_k(\cdot)$, $k = 1, \cdot, K$, the goal is to draw samples from the stacked inference $\sum_{k=1}^{K} w_k q_k(\cdot)$. We first draw a fixed-sized $S_k^\star = \lfloor S w_k \rfloor$ sample randomly without replacement from the k-th inference, and then sample the remaining $S - \sum_{k=1}^{K} S_k^\star$ samples without replacement with the probability proportional to $w_k - S_k^\star/S$ from inference k.

C.3 Closed-form expression for the rank-based integral in Eq. (6)

In the context of Sect. 2.2, we consider N i.i.d. rank samples $r_1, \ldots, r_N$ and their corresponding empirical CDF $\hat{F}_{r,N}(t) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}(r_n \leq t)$. We derive a closed-form expression of a Cramér–von Mises-type distance between $\hat{F}_{r,N}$ and the CDF of a uniform distribution $F_{\mathcal{U}(0,1)}(t) = t1_{t\in[0,1]}$:

$$\begin{aligned}
\int_0^1 |\hat{F}_{r,N}(t) - t|^2\, dt &= \int_0^1 \hat{F}_{r,N}(t)^2\, dt - 2\int_0^1 t\hat{F}_{r,N}(t)\, dt + \frac{1}{3}, \\
&= \frac{1}{N^2}\sum_{i,j=1}^{N}\int_0^1 \mathbb{1}(r_i \leq t)\mathbb{1}(r_j \leq t)\, dt - \frac{2}{N}\sum_{i=1}^{N}\int_0^1 t\,\mathbb{1}(r_i \leq t)\, dt + \frac{1}{3}, \\
&= \frac{1}{N^2}\sum_{i,j=1}^{N}(1 - \max(r_i, r_j)) - \frac{1}{N}\sum_{i=1}^{N}(1 - r_i^2) + \frac{1}{3}, \\
&= \frac{1}{N}\sum_{i=1}^{N} r_i^2 - \frac{1}{N^2}\sum_{i,j=1}^{N}\max(r_i, r_j) + \frac{1}{3}.
\end{aligned}$$

D Experiment Details

D.1 Toy example: hybrid stacking in Gaussian posteriors

In Section 3, we design a true posterior inference $\theta|y \sim \mathrm{normal}(y, 1)$. We consider four manually corrupted posterior inferences,

Hyperparameter	Minimum value		Maximum value		Distribution
	`sbibm`	SimBIG	`sbibm`	SimBIG	
Nb. of MAF layers	3	5	8	11	uniform
Nb. of MLP hidden units	32	256	256	1024	log-uniform
Nb. of MLP layers	2	2	4	4	uniform
MLP Dropout prob.	0.0	0.1	0.2	0.2	uniform
Batch size	20	20	100	100	uniform
Learning rate	1e-5	5e-6	1e-3	5e-5	log-uniform

Table 3: Constraints for the random selection of the hyperparameters of the neural posterior estimators.

Inference 1: $\theta|y \sim \text{normal}(y+1, 1)$,
Inference 2: $\theta|y \sim \text{normal}(y-1, 1)$,
Inference 3: $\theta|y \sim \text{normal}(y, 0.56)$,
Inference 4: $\theta|y \sim \text{normal}(y+0.5, 2.45)$.
They are designed in such a way that the KL divergence between the true posterior and each of the four approximate inferences is roughly the same.

In hybrid stacking, we maximize the sum of log density and rank-calibration:

$$\max_{\mathbf{w}\in\mathbb{S}_K} \left(\sum_{n=1}^{N} \log \sum_{k=1}^{K} w_k q_k(\theta_n|y_n) - \lambda \left(\left(\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} \log(w_k r_{kn}) + 1 \right)^2 + \left(\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} (w_k r_{kn}) - \frac{1}{2} \right)^2 \right) \right).$$

We use $\lambda = 100$, because the calibration error is of a smaller scale. Though it is straightforward to further tune λ via standard cross-validation, we keep a default value without tuning.

D.2 Hyperparameters in SBI Benchmark and SimBIG

In Section 4, we have demonstrated our stacking on three models from the SBI Benchmark: "Two Moons", "simple likelihood and complex posterior", and SIR model. In addition, we applied our method to a cosmological inference task "SimBIG".

For each inference task, we run a large number of neural posterior inferences, constructed by varying the hyperparameters in the normalizing flow on a grid. Table 3 summarizes the range of hyperparameters we used.

D.3 Additional experiments using neural spline flows

In the main paper we have tested stacking on Masked Autoregressive Flow. Here we run stacking on five SBI benchmark tasks (two moons, SLCP, SIR, Gaussian mixture and Lotka-Volterra). In each task, we run $K = 50$ neural spline flows to approximate the Bayesian posterior, and run posterior stacking these flows. Table 4 demonstrates the gain of the mixture stacking and the moment stacking where we evaluate the log predictive density or the moment errors in each task. Our stacking approach outperforms uniform weighting and selection.

Task	Mixture for KL [Log Pred. Density] ↑			Moments Stacking [Moments Error] ↓		
	Best	Unif.	Stacked	Best	Unif.	Stacked
Two Moons	$3.57_{\pm.01}$	$3.54_{\pm.01}$	$\mathbf{3.66}_{\pm.01}$	$-1.72_{\pm.02}$	$-1.72_{\pm.02}$	$\mathbf{-1.73}_{\pm.02}$
SLCP	$-4.44_{\pm.03}$	$-4.38_{\pm.02}$	$\mathbf{-4.06}_{\pm.03}$	$0.86_{\pm.02}$	$0.99_{\pm.01}$	$\mathbf{0.75}_{\pm.01}$
SIR	$7.78_{\pm.02}$	$7.77_{\pm.02}$	$\mathbf{7.88}_{\pm.02}$	$-7.02_{\pm.02}$	$-6.99_{\pm.01}$	$\mathbf{-8.49}_{\pm.02}$
Gauss. Mixture	$-1.24_{\pm.02}$	$-1.16_{\pm.02}$	$\mathbf{-1.13}_{\pm.02}$	$0.28_{\pm.01}$	$0.30_{\pm0.01}$	$\mathbf{0.26}_{\pm.01}$
Lotka-Volterra	$12.94_{\pm.02}$	$12.69_{\pm.02}$	$\mathbf{13.47}_{\pm.02}$	$-6.58_{\pm.03}$	$-5.52_{\pm.01}$	$\mathbf{-6.91}_{\pm.01}$

Table 4: We run stacking on five benchmark tasks, each with the $K = 50$ neural spline flows. The gray number indicates the standard error.