A Divergence-Based Method for Weighting and Averaging Model Predictions

Olav Benjamin Vassend
University of Inland Norway

Abstract

This paper uses a minimum divergence framework to introduce a new way of calculating model weights that can be used to average probabilistic predictions from statistical and machine learning models. The method is general and can be applied regardless of whether the models under consideration are fit to data using frequentist, Bayesian, or some other fitting method. The proposed method is motivated in two different ways and is shown empirically to perform better than or on a par with standard model averaging methods, including model stacking and model averaging that relies on Akaike-style negative exponentiated model weighting, especially when the sample size is small. Our theoretical analysis explains why the method has a small-sample advantage.

Keywords: Model averaging, model stacking, PAC Bayes

1 INTRODUCTION

It is well known that averaging predictions from multiple models can improve predictive accuracy. Averaging model predictions generally requires that we assign a weight to each model that represents how accurate the model is likely to be on future observations. In addition to their role in model averaging, model weights are often used to compare the relative merits of competing models (Wagenmakers and Farrell, 2004), or for assessing the importance of predictors (Burnham and Anderson, 1998, p. 167). This paper considers the problem of assigning weights to models within a minimum-divergence framework similar to those employed in Williams (1980); Diaconis and Zabell (1982) and Bissiri et al. (2016), and proposes a simple method for doing so. Our perspective throughout is predictive: the objective is to construct a probability distribution over future outcomes that achieves high predictive accuracy, where accuracy is typically quantified using a strictly proper scoring rule (Gneiting and Raftery, 2007).

Although our main aim in this paper is to show that the proposed method often has a predictive advantage over competing approaches, especially in the small-sample case, the method has other strengths as well. First, in contrast to many competitors, it is not intrinsically computationally intensive. Second, the weights it produces tend to be more stable than those produced by other approaches in our experiments (see Section 4.1). More broadly, another aim of this article is to bridge several literatures that are not often connected and to shed light on when and why different model-weighting approaches work.

In brief, the method that will be proposed works as follows. Given K models $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_K$, and n data points $y_1, y_2, \ldots, y_n$, let $p_k^p(y_i)$ be the probability (density) that model k assigns to data point y_i after the model has been fit to all of the data using some fitting method. Let $\tilde{y_1}, \tilde{y_2}, \ldots, \tilde{y_n}$ be a draw of n new data points. We then define the "optimism" of each model as follows:

$$op_k = -\sum_i \log p_k^p(\tilde{y}_i) + \sum_i \log p_k^p(y_i) \qquad (1)$$

op_k is the degree to which the in-sample accuracy overestimates the predictive accuracy of the model on future data. Note that it is a random variable since it depends on the future (unseen) data. We now define the following set of optimism-penalizing "prior" model weights:

$$w_k^{op} = \frac{e^{-op_k}}{\sum_{i=1}^K e^{-op_i}} \qquad (2)$$

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

These prior weights summarize the *relative* optimism of all the models under consideration in the sense that models that are comparatively optimistic receive a lower probability and models that are comparatively pessimistic receive a higher probability. Finally, we consider the goal of selecting optimal "posterior" model weights w_k^p. Formally, if S^K is the set of probability mass distributions with K components, we consider the following optimization problem:

$$\min_{w^p \in S^K} \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} - \sum_i \log \sum_k w_k^p p_k^p(y_i) \qquad (3)$$

The left-hand sum in (3) is the KL divergence from the posterior weights to the optimism-penalizing prior model weights whereas the right-hand sum is a measure of the predictive accuracy of the posterior weights on the data, so that (3) may be regarded as trading off divergence from the prior against divergence from the data. For lack of a better name, we therefore call posterior model weights that optimize the above expression "divergence-based weights." (3) is a convex optimization problem in w^p (the KL term is strictly convex and -log of an affine positive form is convex), hence it has a unique solution and can be solved efficiently with general-purpose optimizers, e.g., Solnp (Ghalanos and Theussl, 2015).

In a Bayesian context, related optimization problems have been proposed by Masegosa (2020), Futami et al. (2021), Futami et al. (2022), and Morningstar et al. (2022). However, these papers are concerned with Bayesian inference within a single model, whereas our goal is to average predictions from multiple (complex and often non-Bayesian) models, which is why we need a prior that penalizes model optimism. Crucially, the divergence-based model weighting optimization problem 3 does not fall under the general "rule of three" format proposed by Knoblauch et al. (2022), because the fit-to-data term averages over the posterior inside rather than outside the loss function. Very recently, and independently of us, McLatchie et al. (2025) propose a variant of (3) that does average over the posterior inside rather than outside the loss function, but—in contrast to us—McLatchie et al. (2025) focus on predictive accuracy given a Bayesian model rather than averaging over multiple models.

It is important to emphasize that the optimism-penalizing "prior" is not a Bayesian prior distribution reflecting a belief that each model is true. As we explain in the appendix, we think it is sensible to interpret the prior and posterior weights as reflecting a comparative degree of trust in the predictive accuracy of each model under consideration.

Because model optimism as defined in (1) involves future data, it must itself be estimated. A variety of approaches are available for this purpose, including cross-validation, the bootstrap, and information criteria such as AIC or WAIC. In order to make comparisons with competing methods as transparent and fair as possible, all experiments in this paper use 5-fold cross-validation (CV) to estimate optimism. That is, let $F_1, F_2, \ldots, F_5$ be a subdivision of the data into 5 equally sized folds, let $-F_j$ consist of the data not in F_j, and let $p_k^{-F_j}$ denote model k when fit to all data not in F_k. We then estimate model k's optimism by: $\hat{op}_k = -\sum_{j=1}^5 \log p_k^{-F_j}(F_j) + \sum_j \log p_k^p(F_j)$.

Although we consistently use 5-fold CV in this paper, we note that a significant advantage of our method is that we can avail ourselves of any estimate of model optimism. For example, the AIC penalty term is a good alternative if all the models are parametric and fit using maximum likelihood. It is significantly more stable and less computationally expensive than CV, and (somewhat surprisingly perhaps) in our experience it often yields better predictive accuracy with small sample sizes than CV.

For reasons of numerical stability, we solve the following equivalent version of (3) in practice:

$$\min_{w^p \in S^K} \sum_k w_k^p \log w_k^p + \sum_k w_k^p op_k - \sum_i \log \sum_k w_k^p p_k^p(y_i) \qquad (4)$$

The implementation of divergence-based model-weighting can then be summarized as follows: (i) use cross validation (or some other method) to estimate the optimism op_k of each of the K models under consideration. (ii) Fit each model to all the data $y_1, \ldots, y_n$ and form the matrix of pointwise predictions $p_k^p(y_i)$. (iii) Plug the estimates of op_k and $p_k^p(y_i)$ into (4) and solve for w^p.

The plan for the rest of the paper is as follows. Section 2 briefly introduces model stacking and negative exponentiated weighting. Section 3 presents two distinct theoretical justifications of divergence-based model weighting. Section 4 demonstrates the method in a linear regression simulation and on several data sets. Finally, Section 5 discusses various ways divergence-based model weighting may be extended and improved upon. The appendix shows how to implement the method in R (R Core Team, 2020), gives examples of how to use the method in a Bayesian context, gives a third justification of the method that shows how it may be regarded as a modification of Bayesian model averaging. The R code that was used to conduct all simulations and data analyses in the paper is avail-

able on GitHub at `https://github.com/Vassendo/DivergenceBasedModelWeighting`.

2 EXISTING MODEL WEIGHTING AND AVERAGING METHODS

If we look at existing model weighting and averaging methods, we can distinguish between two broad classes of popular approaches. One approach is to separately calculate a predictive score for each model and then transform these scores into model weights. Bayesian model averaging as well as model averaging based on information criteria, such as AIC-based model weighting (Akaike, 1979; Rigollet and Tsybakov, 2012), fall into this category. The other popular approach is to directly estimate the best model weights or the best combination of model predictions by creating a "model ensemble" (Wolpert, 1992; Breiman, 1996; Yao et al., 2018). As we will see, both these approaches have drawbacks that the divergence-based procedure that will be proposed in this paper appears to overcome.

2.1 Negative Exponentiated Model Weighting

A common scheme for model weighting/averaging computes a nonnegative predictive score $s_{\mathcal{M}_k}$ for each model $\mathcal{M}_k$ and converts scores to probabilities via *negative exponentiation*:

$$w_k = \frac{e^{-s_{\mathcal{M}_k}}}{\sum_{i=1}^{K} e^{-s_{\mathcal{M}_i}}}. \tag{5}$$

Bayesian model averaging fits this template when $s_{\mathcal{M}_k}$ is the negative log marginal likelihood. Although alternatives exist (e.g., Vassend (2022)), negative exponentiation is standard and enjoys desirable theoretical properties in learning theory (Vovk, 1990; Cesa-Bianchi and Lugosi, 2006; van der Hoeven et al., 2018). Predictive scores can be defined in many ways. A prominent approach uses estimated optimism as in 1. Let $\hat{op}_k$ denote the estimated (expected) optimism of model k. If the goal is model selection, one chooses

$$\min_k \sum_i -\log p_k^p(y_i) + \hat{op}_k. \tag{6}$$

In parametric settings, AIC (Akaike, 1973, 1974; Sugiura, 1978; Hurvich and Tsai, 1989; Burnham and Anderson, 1998) approximates optimism by the parameter count; more generally, resampling or data splitting (e.g., cross-validation) is used. Optimism-based *weighting* differs from selection by applying (5) to the criterion instead of minimizing it directly. As shown in Section 4 of the Appendix, all such negative-exponentiated weightings implicitly solve

$$\min_{w^p \in S^K} \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} - \sum_i \sum_k w_k^p \log p_k^p(y_i), \tag{7}$$

Despite being widely used, negative exponentiation has a well-known drawback: as the sample size grows, the weights tend to concentrate on the single best model, even when a convex combination would predict better. This issue, noted for Bayesian model averaging by Minka (2002) and detailed in Section 4 of the Appendix, arises for all forms of negative-exponentiated weighting.

2.2 Model Stacking

Another approach treats model averaging as an estimation or optimization problem. A standard method is stacking (Stone, 1974; Wolpert, 1992; Breiman, 1996; Leblanc and Tibshirani, 1996; Yao et al., 2018), which chooses weights via cross-validation to optimize predictive accuracy. Originally for averaging point forecasts, Yao et al. (2018) use the logarithmic score to stack Bayesian predictive distributions; however, Yao et al.'s (2018) method applies beyond the Bayesian setting. We call this general procedure "stacking with the log score."

Let $F_1, \ldots, F_m$ be m equally sized folds and $-F_j$ their complements. Fit each model on $-F_j$ (e.g., ML or Bayesian inference) to obtain leave-F_j-out predictors $p_k^{-F_j}$. Then solve

$$\min_{w \in S^K} \sum_{F_j} \sum_{y_i \in F_j} -\log\Big(\sum_k p_k^{-F_j}(y_i)\, w_k\Big). \tag{8}$$

Unlike negative exponentiated weighting, stacking directly estimates weights for the optimal linear combination of predictions. When no single model clearly dominates, stacking often outperforms methods based on negative exponentiation, including Bayesian model averaging (Clarke, 2003).

In machine learning, stacking often uses a two-level setup: leave-F_j-out predictors form "level-one" learners whose predictions feed a "level-two" meta-learner (or super-learner) (van der Laan et al., 2007), such as e.g., logistic regression, gradient boosting, or random forests. Unlike previous averaging schemes, the meta-learner need not produce probabilistic model weights to yield combined predictions.

Various forms of stacking enjoy good asymptotic properties (Le and Clarke, 2017; Yao et al., 2021). Nonetheless, as we show empirically (simulations and datasets

in Section 4; additional results in Section 2 of the Appendix), stacking can struggle with relatively small datasets, which is an issue that persists even when regularization is incorporated (see Section 4).

3 THEORETICAL PERSPECTIVES ON THE DIVERGENCE-BASED APPROACH TO MODEL AVERAGING

In the next two subsections, we provide two theoretical justifications of divergence-based model weighting. Section 4 of the appendix provides yet another motivation for the method, which also shows how the method may be regarded as a modification of Bayesian model averaging.

3.1 Divergence-Based Model Weighting as an Empirical Approximation of the Ideal Objective Function

Since our goal is to maximize predictive accuracy on future data, we would ideally choose posterior model weights that minimize the following expression, where the expectation is with respect to the true (but unknown) probability distribution that governs the actual distribution of future data y:

$$\min_{w^p \in S^K} \mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] \tag{9}$$

Because this objective depends on unknown quantities, we must rely on a tractable surrogate. A natural first attempt is to minimize the loss of the model average evaluated on the observed data:

$$\min_{w^p \in S^K} \sum_i -\log \sum_k w_k^p p_k^p(y_i) \tag{10}$$

However, (10) is biased, since it uses the same data both to fit the individual models and to assess the performance of the weighted average. This bias favors models that are overly optimistic (as defined in 1). Given the relationship between $p_k^p(\tilde{y}_i)$ and $p_k^p(y_i)$ in (1), a natural idea is to "discount" each weight w_k^p by a factor proportional to $e^{-o\hat{p}_k}$, where $o\hat{p}_k$ is an estimate of model k's optimism. But because the posterior weights are probabilities and must sum to one, absolute optimism levels are not what matter—what matters is each model's optimism relative to the others. To that end we form the vector of *optimism-penalizing prior model weights* as defined in 2 to downweight overly optimistic models and upweight relatively conservative ones. To incorporate this into the objective, we add a penalty term, $F(w^p, w^{op})$ that penalizes posterior weights in proportion to how far they diverge from the prior weights:

$$\min_{w^p \in S^K} cF(w^p, w^{op}) - \sum_i \log \sum_k w_k^p p_k^p(y_i) \tag{11}$$

Here, c is a constant that regulates the influence of the prior weights as compared to the data in determining the optimal posterior weights. For F, a natural and flexible choice is the class of f-divergences, which take the following form, for some convex function f:

$$F(w^p, w^{op}) = \sum_k w_k^p f(\frac{w_k^{op}}{w_k^p}) \tag{12}$$

To rule out pathological counter-examples, we assume that the generator function f is sufficiently well-behaved: in particular, we assume that it is real analytic. As we discuss in the appendix, this assumption is much stronger than necessary, but has the benefit of being simple to state. In any case, most f-divergences used in practice have this property.[1] The key questions now are (i) which f-divergence we ought to use and (ii) what value to assign to the constant c. To make headway on these questions, let's consider the special case where one of the models under consideration is substantially better than the other models, so that the best way of averaging the model predictions is simply to assign all (or almost all) the posterior weight to the single best model. A natural boundary condition is that, in this special case (11) should agree with the standard optimism-based model-selection criterion, i.e., (6). After all, when ordinary model selection is optimal, our weighting method should agree with it.

Or, to state the same boundary condition differently, let V^K be the set of vertices of the simplex S^K, i.e., the set of probability vectors w such that $w_k = 1$ for some k. Then if we optimize (3) over V^K rather than S^K, so that we force the method to select a single method, then the method should agree with the standard optimism-based model selection method expressed in (6). In other words, we require:

[1] Importantly, even if we did not impose this extra condition in Theorem 3.1, the theorem would still rule out all *standard* f-divergences aside from the KL divergence.

$$\begin{aligned}\arg \min_{w^p \in V^K} c \sum_k w_k^p f(\frac{w_k^{op}}{w_k^p}) - \sum_i \log \sum_k w_k^p p_k^p(y_i) \\ = \arg\min_k \sum_i -\log p_k^p(y_i) + op_k\end{aligned} \tag{13}$$

Imposing the boundary condition in (13) is enough to uniquely determine both c and f:

Theorem 3.1. *If we require that the boundary condition (13) be satisfied in all situations and that the generator f be real analytic, then:*

1. *The f-divergence in (11) must be the KL divergence.*
2. *The constant c in (11) must equal* 1.
3. *Consequently, (11) coincides with the divergence-based model weighting problem (3).*

The brief reason why (13) entails Theorem 3.1 is that (13) forces the two methods to impose the same ranking over models, and this in turn forces them to be equal up to a constant offset that does not matter for optimization purposes. A more careful proof is in the appendix.

Theorem 3.1 gives a strong theoretical reason for setting $c = 1$ and for using the KL divergence. In Section 5 of the appendix, we consider what happens if we let c deviate from 1 in a linear regression experiment. The conclusion from this experiment is that $c > 1$ is somewhat better at smaller sample sizes, consistent with our theoretical analysis in Section 3.2.2 below, but that this comes at the price of worse performance at larger sample sizes. We therefore think $c = 1$ is a good default option. In Section 5 of the appendix, we also investigate what happens if we replace the KL divergence with the Brier divergence, $\sum_k (w_k^p - w_k^{op})^2$. The upshot is that the KL divergence performs substantially better. Finally, we also consider replacing the optimism-penalizing prior (2) with the flat distribution $w_k = 1/K$. Again, the conclusion is that the flat distribution results in much worse predictions than the optimism-penalizing prior (2).

3.2 Justifications based on inequality bounds

Readers familiar with PAC-Bayesian theory will notice a resemblance between the divergence-based model weighting optimization problem (3) and standard PAC-Bayesian bounds. This similarity makes it natural to try to analyze divergence-based model weighting from a PAC-Bayesian point of view, especially because PAC-Bayesian justifications of (certain sorts of) Bayesian model averaging have already been provided by Germain et al. (2016) and McAllester (1999).

However, in attempting to construct a PAC Bayesian analysis, we face two problems: (i) standard PAC Bayesian inequalities bound the expected average of the log likelihoods of the model, $\mathrm{E}[-\sum_k w_k^p \log p_k^p(y)]$, but what we are interested in bounding is the expectation of log of the linear mixture of the models, $\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)]$, i.e., (9). In other words, the log operator of PAC-Bayesian bounds is in the "wrong" place. (ii) Divergence-based model weighting (3) uses the same data to both fit the models and then evaluate predictive accuracy, whereas standard PAC-Bayesian bounds assume that the models under consideration are evaluated on new data. In what follows we give two different inequality bounds: the first bound assumes that overfitting is insignificant and shows how we can convert a standard PAC Bayesian bound into a bound on (9) by making assumptions about the tail behavior of the log scores of the models. The second bound allows for substantial overfitting and makes no assumptions about tail behavior, but is loose if overfitting is not substantial. All proofs are in the appendix.

3.2.1 The first inequality bound

Standard PAC-Bayes bounds place the logarithm in the "wrong" location. Still, we can often turn those bounds into ones we care about by assuming mild tail behavior for the models' log scores. The key quantity is the pointwise Jensen gap, which measures how far the log of the mixture is from the mixture of the logs:

$$J_{out}(w^p) = -\sum_k w_k^p \log p_k^p(\tilde{y}_i) + \log \sum_k w_k^p p_k^p(\tilde{y}_i) \tag{14}$$

If the individual log scores are well behaved, this gap will usually be well behaved too. For illustration, assume the individual log losses are sub-Gaussian: there exists $s > 0$ such that for all $\lambda \in R$:

$$\log \mathrm{E}[e^{\lambda(-\log p_k^p(\tilde{y}) - \mathrm{E}[-\log p_k^p(y)])} \leq \frac{\lambda^2 s^2}{2} \tag{15}$$

Because the model space is finite, sub-Gaussian log losses imply a sub-Gaussian Jensen gap. We have:

Theorem 3.2. *If each of $-\log p_k^p(\tilde{y}_i)$ is sub-Gaussian, then there exists a variance s', such that for every set of posterior weights w^p, the Jensen gap $\sum_i J_{out}(w^p)$ is also sub-Gaussian with variance at most ns'.*

Theorem (3.2) shows that $\sum_i J_{out}(w^p)$ is subgaussian for all w^p. We now make the additional uniformity assumption that $\sup_{w^p} \sum_i J_{out}(w^p)$ is also subgaussian

for some variance ns''. This is an extra assumption, but it is plausible in many common scenarios. Applying a subgaussian bound then gives us a general way of turning a PAC-Bayesian bound into a bound on our quantity of interest. For example, combining this assumption with the result with Corollary 4 of Germain et al. (2016) yields the following:

Theorem 3.3. *Suppose each model $-\log p_k^p(\tilde{y}_i)$ is sub-Gaussian with variance parameter s, and let ns'' be defined as above. Then for any prior distribution w_k and any $\delta \in (0, 1]$, with probability at least $1-\delta$ we have, simultaneously for all distributions w_k^p:*

$$
\begin{aligned}
&n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] \\
&\leq \sum_k w_k^p \log \frac{w_k^p}{w_k} - \sum_i \log \sum_k w_k^p p_k^p(\tilde{y}_i) \\
&\quad + n\frac{1}{2}s^2 + \log \frac{1}{\delta} + \sqrt{2ns'' \log \frac{1}{\delta}}
\end{aligned}
$$

The values of s and s'' of course matter to the tightness of the bound, but for optimization purposes, they are irrelevant. If overfitting is insignificant, so that $op_k \approx 0$, we can justifiably substitute the in-sample error $\sum_i -\log \sum_k w_k^p p_k^p(y_i)$ for the (unknown) out-of-sample $\sum_i -\log \sum_k w_k^p p_k^p(\tilde{y}_i)$ and hence optimizing Theorem (3.3) reduces to optimizing the divergence-based model weighting problem (3). Importantly, in this setting the prior weights w can be chosen freely; they need not be the optimism-penalizing weights of (2), since the assumption of negligible overfitting removes the need for explicit optimism correction.

A natural concern with Theorem 3.3 is that the sub-Gaussian assumption may seem overly restrictive. This assumption was adopted for clarity of exposition rather than necessity. Weaker tail conditions—such as sub-exponentiality of the log-likelihood terms—yield bounds of the same general form, albeit looser. In Section 5 of the appendix, we do a linear regression experiment with heavy-tailed models. The result is still that divergence-based model weighting performs better than the alternatives.

3.2.2 The second inequality bound

In this section we wish to provide a bound without imposing any assumptions on the tail behavior of the models. Our goal here is to provide a bound that is instructive when overfitting is substantial. To quantify the degree of overfitting of each model, define:

$$
m_k = \frac{\sum_i -\log p_k^p(\tilde{y}_i)}{\sum_i -\log p_k^p(y_i)} \tag{16}
$$

The ratio m_k is a measure of the extent to which model k overfits on the data. Note that in general $m_k > 1$. Let $m = \min_k m_k$. In cases where all models under consideration overfit, m will be much greater than 1. By applying Jensen's inequality to the left side of the inequality in Theorem 3 in Germain et al. (2016) and using (16) to simplify the right side, we have:

Theorem 3.4. *Let m_k be defined as in (16) and let $m = \min_k m_k$. Assume $\sum_i -\log p_k^p(y_i) > 0$ for each model k, and $m > 1$. For any $\delta \in (0, 1]$, with probability at least $1-\delta$ we have, simultaneously for all distributions w_k^p:*

$$
\begin{aligned}
n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] &\leq \frac{m}{m-1} \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} \\
&\quad - \frac{1}{m-1} \sum_k w_k^p \log w_k^p \\
&\quad + \log K - \frac{m}{m-1} \log \sum_k e^{-op_k} - \log \delta + \phi_{term}.
\end{aligned}
$$

Here, $\phi_{term} = \log \frac{1}{K} \sum_k \mathrm{E}[e^{\sum_i \log p_k^p(\tilde{y}_i) - n\mathrm{E}[\log p_k^p(y)]}]$, and does not depend on w^p. When m is large (i.e., the models substantially overfit), we get, approximately:

$$
n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] \leq \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} + C
$$

Where C collects terms that do not depend on w^p. Hence, when overfitting is substantial, we can estimate the optimal weights for $n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)]$ by simply optimizing $\sum_k w_k^p \log \frac{w_k^p}{w_k^{op}}$. This helps explain why both divergence-based model weighting (3) and negative exponentiated model weighting perform well when the sample size is very small, since both (3) and (7) include, and will tend to be dominated by, $\sum_k w_k^p \log \frac{w_k^p}{w_k^{op}}$ in the small-sample case.

3.2.3 An asymptotic result

A natural concern is that divergence-based model weighting might systematically overfit, since it relies on the same data both to fit the models and to construct the prior model weights. If this were the case, the procedure could fail to converge to the ideal objective (9). The following result shows, however, that if the methods used to fit each of the models individually converge, then the the empirical objective (11) converges to the ideal objective (9) as the sample size grows. The proof relies on the log-sum inequality and is given in the supplementary materials.

Theorem 3.5. *For $k = 1, 2, \ldots, K$, let p_k^p be a predictive distribution that results from fitting model*

$\mathcal{M}_k$ to data $D = \{y_1, y_2, \ldots, y_n\}$ that are distributed i.i.d. according to some distribution Q. Let $\tilde{D} = \{\tilde{y_1}, \tilde{y_2}, \ldots, \tilde{y_n}\}$ also be i.i.d. from Q, let $op_k = -\sum_i \log p_k^p(\tilde{y}_i) + \sum_i \log p_k^p(y_i)$ be the optimism of each model. Let S^K be the set of all probability functions with K components. Suppose each p_k^p converges in probability to a distribution p_k^* in the sense that $\frac{1}{n}\sum_i |[\log p_k^*(y_i) - \log p_k^p(y_i)]| \xrightarrow{P} 0$ and $\frac{1}{n}\sum_i |[\log p_k^*(\tilde{y}_i) - \log p_k^p(\tilde{y}_i)]| \xrightarrow{P} 0$. Suppose also that all relevant integrals are finite. Then the following holds for all $w^p \in S^K$:

$$\frac{1}{n}\Big|\sum_k w_k^p \log w_k^p + \sum_k w_k^p op_k - \sum_i \log \sum_k w_k^p p_k^p(y_i) - \mathrm{E}[-\log \sum_k w_k^p p_k^p(y)]\Big| \xrightarrow{P} 0$$

This result shows that divergence-based model weighting—in contrast to negative exponentiated model weighting, but similarly to model stacking—has the nice asymptotic property of converging towards the ideal (9).

4 EXPERIMENTAL RESULTS

The following two subsections apply divergence-based model averaging in simulations and on data sets. Section B of the appendix applies the method in a Bayesian context and compares it to Bayesian stacking and Pseudo-BMA+ (Bayesian negative exponentiated model weighting) (Yao et al., 2018).

4.1 Linear Regression Simulation Experiment

To show how divergence-based model weighting works in practice, we consider a linear regression simulation experiment, where the true distribution is of the following form:

$$Y = \mathbf{X}\beta + \alpha + \epsilon \tag{17}$$

Here, β is a 20-dimensional vector, α is an intercept, and ϵ is a normally distributed error term. We consider two possible ways of generating a precise data-generating distribution. In the first, non-sparse setting, the true values of the parameters of the data-generating distribution are determined as follows: the components of β are generated independently from $\mathcal{N}(\mu = 0, sd = 0.5)$; α is generated from $\mathcal{N}(\mu = 0, sd = 2)$; and ϵ is generated from $\mathcal{N}(0, 5)$. In the second, sparse setting, the values of the parameters are set in the same way, except that half of the components of the β vector are randomly set to 0. To generate the design matrix, we also consider two possible scenarios. In the first scenario, all the entries of the design matrix $\mathbf{X}$ are generated independently from $\mathcal{N}(\mu = 0, sd = 1)$. In the second scenario, the columns are set to have pairwise correlations of 0.5. In total, then, we consider four different types of data-generating distribution.

Next, we create a set of 10 predictive models, where each model is constructed randomly in the following way: first, we select (with uniform probability) a number m between one and five; next, m predictors are randomly selected from the design matrix and are joined with an intercept term to form a normal, linear regression model. Of course, in practice we would not want to randomly construct our models. However, the purpose of this section is to give a fair assessment of how divergence-based model weighting performs given an arbitrary model space.[2]

We generate a training set with a sample size of n (ranging from 10 to 200) and a test set with 200 data points. We use maximum likelihood estimation to fit each of the models on the training set and then combine the predictions from the 10 fitted models on the test set by using the following methods: (1) divergence-based model weighting, (2) Negative exponentiated model weighting with 5-fold CV, (3) model stacking with the log score.

Finally, we calculate the root mean squared error (RMSE) of each of the model averaging methods on the test set. To get a sense of how the various model averaging methods perform across a range of model spaces and given different data-generating distributions, the entire preceding procedure is repeated 1000 times with different randomly created data-generating distributions and model spaces, and the results averaged. The results are shown in Figure 1.

In this simulation, stacking and divergence-based weighting are asymptotically equal and better than Akaike style negative exponentiated weighting. However, for very small sample sizes, divergence-based weighting and Akaike weighting both perform substantially better than stacking. These results are in line with the claims we made in Section 2.

An important question concerns the stability of the model weights, especially because model optimism is estimated via cross-validation. Weight stability is also important for interpreting comparative predictive accuracy between models. To investigate this issue, we fix a ground-truth data-generating process and a set of models, and then repeat the simulation multiple times to examine how the estimated weights vary across

[2]See the R code in the supplementary material for the precise procedure used to generate the models.

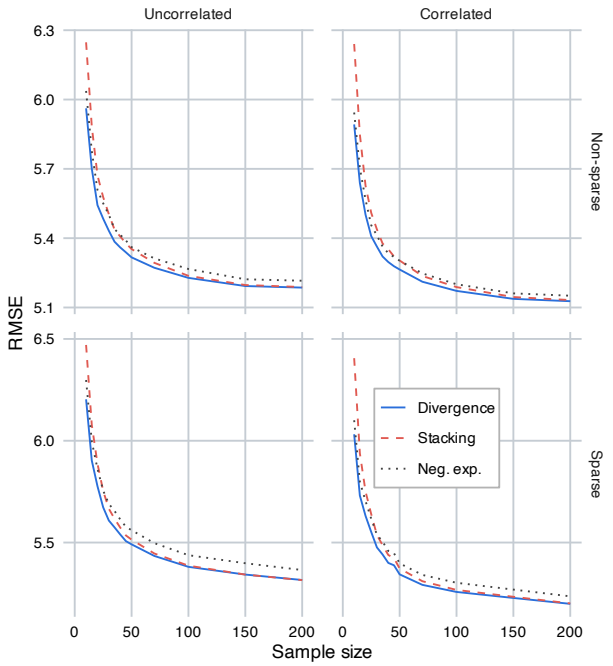

Figure 1: RMSE of model-weighting methods with various kinds of data-generating distribution. The standard error of each point is less than 0.04.

runs. Figure 2 shows, for each sample size, the standard deviation of each weight component (averaged across components) over 1000 runs.

Naturally, the precise values in Figure 2 depend on the particular data-generating process and model space, but the qualitative pattern is representative.[3] As the figure shows, divergence-based model weighting has a systematically lower standard deviation at every sample size. Thus, in this experiment, divergence-based model weighting not only yields more accurate predictions but also produces more stable weights than stacking and negative exponentiated weighting.

4.2 Using Divergence-Based Model Weighting to Average Predictions From Machine Learning Models

We consider twelve data sets of varying sizes and dimensions from the UC Irvine Repository (Dua, Dheeru and Graff, 2019). For reasons of space, a complete description of the data sets is omitted, but a footnote gives the official name of each data set and a reference, if one is available.[4]

[3]Readers are encouraged to experiment with alternative parameter settings in the simulation.

[4]In what follows, we refer to data sets in accordance with each data set's "citation request" in the UC Irvine Data Base. Not every data set has an associated publication. In descending order, the data sets in Table 1 are: "Cervical cancer behavior risk data set" (Sobar et al., 2016). "Fertility data set" (Gil et al., 2012). "Heart failure clinical records data set" (Chicco and Jurman, 2020). "Hungarian heart disease data set" (donated by Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.). "Cleveland heart disease data set" (donated by V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D.). "Early stage diabetes risk prediction dataset" (Islam et al., 2019). "ILPD (Indian Liver Patient Dataset) Data Set". "Breast Cancer Wisconsin (Original) Data Set." "Statlog (Australian Credit Approval) Data Set." "Statlog (German Credit Data) Data Set." "Wine Quality Data Set" (Cortez et al., 2009). "Adult/census income data set."

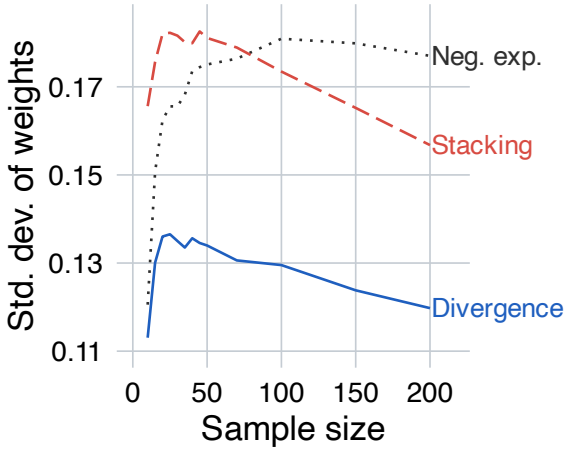

Figure 2: Standard deviation of model weights across 1000 simulation runs for different sample sizes. Lower values indicate more stable weights.

Every data set is minimally cleaned and preprocessed: rows with missing entries are removed, factor predictors are converted to numbers, and finally all the predictors are standardized. Each data set is divided into a training set consisting of 85% of the observations and a test set consisting of the remaining 15% of the observations. For data sets with $n < 100$ observations, the results (and se) are averages of 1000 independent train/test splits; for data sets with $100 < n < 10000$, 500 train/test splits are used; finally, for data sets with $n > 10000$, 100 train/test splits are used.

We consider the following six models: logistic regression, elastic net regression, gradient boosting, support vector machine with a radial basis, random forest, and k-nearest neighbor. We use stratified 5-fold cross validation and the R machine learning interface Caret (Kuhn, 2012) with the default settings (using nested cross validation) to fit the hyper-parameters of all models and to produce probabilistic predictions on the test set, and we also use 5-fold cross validation on the training data set to estimate model

Table 1: Log scores of divergence-based model weighting (DW), stacking with log scores (LS), Akaike style negative exponentiated model weighting with log scores (NEW), stacking with a logistic meta-learner (GLM), stacking with an elastic net meta-learner (NET), and stacking with a gradient boosting meta-learner (GBM). The best score on each row is in bold (lower scores are better). The first two columns contain the number of observations (n) and predictors (p) in each data set. The last column shows the standard error across all repetitions of the mean difference between the best and second best score on each row.

	n	p	DW	LS	NEW	GLM	NET	GBM	se
Cervical cancer	65	19	**0.270**	0.287	0.272	0.705	0.298	0.371	0.0003
Fertility	99	9	**0.275**	0.286	0.283	0.334	0.303	0.310	0.0008
Heart failure	299	11	**0.512**	**0.512**	0.513	0.523	0.520	0.530	0.0004
Heart disease 1	293	13	**0.387**	0.389	0.394	0.401	0.400	0.411	0.0009
Heart disease 2	303	13	**0.390**	0.393	0.400	0.398	0.396	0.412	0.0008
Diabetes	520	16	0.076	0.077	0.075	0.066	**0.065**	0.073	0.0010
Liver disease	578	10	**0.494**	0.498	0.503	0.504	0.504	0.510	0.0006
Breast cancer	682	9	**0.097**	0.104	0.102	0.111	0.106	0.105	0.0009
Australian credit	689	14	**0.322**	0.323	0.330	0.324	0.323	0.328	0.0005
German credit	999	24	**0.481**	0.483	0.495	0.483	0.482	0.486	0.0003
Wine quality	1599	11	0.418	0.415	0.418	**0.409**	0.411	0.418	0.0002
Income prediction	30161	14	0.326	0.321	0.345	0.334	0.335	**0.312**	0.0002
Mean			**0.337**	0.341	0.344	0.383	0.345	0.356	

optimism. To average predictions from the models, we consider—as before—model stacking with the log score, divergence-based model weighting, and Akaike style negative exponentiated model weighting. But in addition, we also use the following standard "meta-learners" to form "stacked" model predictions: logistic regression, elastic net, gradient boosting. Note that all of these meta-learners incorporate regularization. The hyper-parameters of each meta-learner are tuned using 5-fold cross validation and the default settings of the CaretEnsemble (Deane-Mayer and Knowles, 2016) package.

Table 1 gives the results. Divergence-based model weighting has the best log score on nine of the twelve data sets and has the best overall mean score. Moreover, in each pairwise comparison, divergence-based model weighting does better on at least ten of the twelve data sets. Note also that, in the four data sets that have the smallest size, negative exponentiated model weighting (NEW) performs better than all the methods that use various forms of stacking. However, divergence-based model weighting performs even better. And, by contrast with negative exponentiated model weighting, divergence-based model weighting also does well on the larger data sets.

As seen in Table 1, divergence-based model weighting is not always the best-performing method. What might explain this? While we do not have a complete account, our results suggest a tentative explanation. Divergence-based model weighting assigns a separate weight to each model and is therefore linear in the model predictions, similarly to negative exponentiated model weighting and stacking with the log score (including Bayesian stacking). This linearity is advantageous when the sample size is small and is also important for interpretative purposes (especially model comparison). Moreover, Theorem 3.5 shows that divergence-based model weighting is an asymptotically optimal linear way of combining model predictions. However, for certain kinds of data, non-linear combinations can still be predictively superior. This may explain the rows of Table 1 in which divergence-based weighting is outperformed by alternative (non-linear) methods.

5 CONCLUSION

Divergence-based model weighting is a theoretically well-motivated way of calculating model weights that performs well both in simulations and on data. The evidence we have presented suggests that the method performs well compared to the alternatives when sample sizes are small, while also retaining good performance with larger sample sizes. The method can be extended and possibly improved upon in several directions. First, it is possible to reformulate (3) in such a way that the model weights depend on the inputs, similar to how Sill et al. (2009) and Yao et al. (2021) extend stacking to make the stacking weights input-dependent. Second, it might be possible to find prior model weights that would be better than the optimism-penalizing weights we have proposed in this paper. We leave this as work for the future.

Acknowledgments

Writing this paper took a tremendous amount of time and effort. I am greatly indebted to several reviewers who made the paper much stronger than it otherwise would have been. Work on this project has been supported by funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101164097).

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Akaike, H. (1979). A Bayesian extension of the minimum AIC procedure of autoregressive model fitting. *Biometrika 66*(2), 237–242.

Bissiri, P. G., C. Holmes, and S. Walker (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B (Methodological) 78*(5), 1103–1130.

Breiman, L. (1996). Stacked regressions. *Machine Learning 24*(1), 49–64.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review 7*(1), 1–3.

Burnham, K. P. and D. R. Anderson (1998). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach* (Second ed.). New York, NY: Springer-Verlag.

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games*. Cambridge University Press, New York.

Chicco, D. and G. Jurman (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making 20*(1), 16.

Clarke, B. (2003). Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *J. Mach. Learn. Res. 4*(null), 683–712.

Cortez, P., A. Cerdeira, F. Almeida, T. Matos, and J. Reis (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems 47*, 547–553.

Deane-Mayer, Z. A. and J. E. Knowles (2016). caretEnsemble: ensembles of caret models. *R package version 2*(0).

Diaconis, P. and S. L. Zabell (1982). Updating subjective probability. *Journal of the American Statistical Association 77*(380), 822–830.

Dua, Dheeru and Graff, C. (2019). UCI Machine Learning Repository.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016.

Gil, D., J. L. Girela, J. De Juan, M. J. Gomez-Torres, and M. Johnsson (2012). Predicting seminal quality with artificial intelligence methods. *Expert Systems with Applications 39*(16), 12564–12573.

Futami, F., Iwata, T., Ueda, N., Sato, I., and Sugiyama, M. (2021). Loss function based second-order Jensen inequality and its application to particle variational inference. *Advances in Neural Information Processing Systems 34*, 6803 – 6815.

Futami, F., Iwata, T., Ueda, N., Sato, I., and Sugiyama, M. (2022). Predictive variational Bayesian inference as risk-seeking optimization. *International Conference on Artificial Intelligence and Statistics 151*, 5051 – 5083.

Germain, P., Bach, F., Lacoste, A., and Lacoste-Julien, S. (2016). PAC-Bayesian theory meets Bayesian inference. *Advances in Neural Information Processing Systems 29*, 1–9.

Ghalanos, A. and Theussl, S. (2015). Rsolnp: General Non-Linear Optimization.

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

Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika 76*(2), 297–307.

Islam, M. M. F., R. Ferdousi, S. Rahman, and H. Y. Bushra (2019). Likelihood prediction of diabetes at early stage using data mining techniques. *Computer Vision and Machine Intelligence in Medical Image Analysis*.

Knoblauch, J., J. Jewson, and T. Damoulas (2022). An optimization-centric view on Bayes' rule: reviewing and generalizing variational inference. *Journal of Machine Learning Research 23*(132), 1–109.

Kuhn, M. (2012). The caret package. *Journal of Statistical Software 28*.

Kullback, S. and R. Leibler (1951). On information and sufficiency. *Annals of Mathematical Statistics 22*(1), 79–86.

Le, T. and B. Clarke (2017). A Bayes interpretation of stacking for $\mathcal{M}$-Complete and $\mathcal{M}$-Open settings. *Bayesian Analysis 12*(3), 807–829.

Leblanc, M. and R. Tibshirani (1996). Combining estimates in regression and classification. *Journal of the American Statistical Association 91*(436), 1641–1650.

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

McAllester, D. (1999). PAC-Bayesian model averaging. *Proceedings of the 12th Annual Conference on Computational Learning Theory*, 164 – 170.

McLatchie, Y., B.-E. Cherief-Abdellatif, D. T. Frazier, and J. Knoblauch (2025). Predictively oriented posteriors. *arXiv preprint arXiv:2510.01915*.

Minka, A. (2002). Bayesian model averaging is not model combination. *https://tminka.github.io/papers/bma.html*

Morningstar, W. R., A. A. Alemi, and J. V. Dillon (2022). Pacm-Bayes: narrowing the empirical risk gap in the misspecified Bayesian regime. *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics 151*, 8270 – 8298.

R Core Team (2020). R: A Language and Environment for Statistical Computing.

Rigollet, P. and A. B. Tsybakov (2012). Sparse estimation by exponential weighting. *Statistical Science 27*(4), 18,558–575.

Sill, J., G. Takacs, L. Mackey, and D. Lin (2009). Feature-weighted linear stacking. *arXiv:0911.0460v2*.

Sobar, R. Machmud, and A. Wijaya (2016). Behavior determinant based cervical cancer early detection with machine learning algorithm. *Advanced Science Letters 22*(10), 3120–3123.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639.

Stan Development Team (2019). Stan Modeling Language.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological) 36*(2), 111–147.

Sugiura, N. (1978). Further analysis of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics - Theory and Methods 7*(1), 13–26.

van der Hoeven, D., T. van Erven, and W. Kotlowski (2018). The many faces of exponential weights in online learning. *Proceedings of Machine Learning Research 75*, 1–26.

van der Laan, M. J., E. C. Polley, and A. E. Hubbard (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology 6*(1).

Vassend, O. B. (2022). Justifying the norms of inductive inference. *British Journal for the Philosophy of Science 73*(1), 135–160.

Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., Bürkner, P.-C., Paananen, T., and Gelman, A. (2020). LOO: Efficient leave-one-out cross-validation and WAIC for Bayesian models.

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432.

Vershynin, R. (2025). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, New York.

Vovk, V. (1990). Aggregating strategies. In M. Fulk and J. Case (Eds.), *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pp. 371–383. San Mateo, CA: Morgan Kaufmann.

Wagenmakers, E.-J. and S. Farrell (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin & Review 11*(1), 192–196.

Watanabe, S. (2013). A widely applicable Bayesian information criterion. *J. Mach. Learn. Res.*, 14(1):867–897.

Williams, P. M. (1980). Bayesian conditionalisation and the principle of minimum information. *The British Journal for the Philosophy of Science 31*(2), 131–144.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks 5*(2), 241–259.

Yao, Y., V. Aki, S. Daniel, and G. Andrew (2018). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis 13*(3), 917–1007.

Yao, Y., G. Pirs, A. Vehtari, and A. Gelman (2021). Bayesian hierarchical stacking: some models are (somewhere) useful. *arXiv:2101.08954v2*.

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. [Not Applicable] The algorithm involves solving a relatively simple non-linear convex programming problem, which can be done almost instantly using the R package Rsolnp, for example.

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] This is in the supplementary materials.

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] This is provided in the appendix.

(c) Clear explanations of any assumptions. [Yes] In some cases, more complete explanations are provided in the appendix.

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] Some of this information has been relegated to the supplementary material.

(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). [Not Applicable] The algorithm described in this paper can be run on an ordinary laptop from the last 10 years.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Yes]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable] The data sets are from the UC Irvine Machine Learning Repository.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A IMPLEMENTING DIVERGENCE-BASED MODEL WEIGHTING AND STACKING WITH THE LOG SCORE IN R

To calculate the divergence-based model weights for a given set of models, we first need to calculate the pointwise predictive probabilities, $p_k^p(y_i)$, for each of the models and all data points, and gather the results in an n by K matrix. How to do this depends on which models we are fitting as well as what package we use to fit the models. For the sake of concreteness, our running example in this document will be Bayesian inference with a linear model, and we will assume that the models are fit to data using Stan (Stan Development Team, 2019). In a Bayesian context, the pointwise predictive probability function $p_k^p(y_i)$ assumes the form of a pointwise posterior predictive probability (density) function (Gelman et al., 2014), $\int p_k(y_i|\theta)p(\theta|y)d\theta$, i.e., the posterior predictive probability (density) of each data point y_i, after the model has been fit to all the data. Since this latter expression is rather complicated, we will continue to use the generic notation $p_k^p(y_i)$, however.

To calculate the pointwise posterior predictive probabilities for the Bayesian linear regression model using Stan, we include the following code in the "generated quantities" part of the Stan model code:

```
vector[N] log_lik;
for (n in 1:N) log_lik[n] = normal_lpdf(y[n] | X[n, ] * beta + alpha, sigma);
```

We then extract the log-likelihoods from the posterior sample, exponentiate, and take the column averages to calculate the posterior predictive probability densities. (See `https://mc-stan.org/loo/reference/extract_log_lik.html` for more detailed information.)

In addition to a matrix of pointwise posterior predictive probabilities, we also need to estimate the optimism of each model. There are several Bayesian estimates of model optimism, e.g. the penalty term in the Widely Applicable Information Criterion (Watanabe, 2013) or in the Deviance Information Criterion (Spiegelhalter et al., 2002), or the approximate Bayesian leave-one-out cross validation (LOO) estimate in Vehtari et al. (2017). In practice, our experience is that the estimates tend to be quite close to each other. In the next section, we use the LOO-based estimate, which Vehtari et al. (2017) argue is most accurate. This estimate can be calculated easily using the LOO package (Vehtari et al., 2020).

Once we have a vector of model optimism estimates (the unnormalized "prior") and a matrix of pointwise posterior predictive probabilities (which we call "pointwise" in the following), the divergence-based model weights can be calculated in R using the package RSolnp (Ghalanos and Theussl, 2015) with the following function:

```
divergence_weights <- function(pointwise, prior){
  num_of_models <- ncol(pointwise)
  par_start <- rep(1, num_of_models)
  par_start <- par_start/sum(par_start)
  lower_bound <- rep(0, num_of_models)
  upper_bound <- rep(1, num_of_models)
  equal <- function(x){ #Sum to 1 constraint
    sum(x)
  }
  optimizing_fn <- function(x){
    predictions <- pointwise%*%x
    score  <- -sum(log(predictions)) +
      (sum(x*log(x)) + sum(x*prior))
    return(score)
  }
  weights <- solnp(pars = par_start, fun = optimizing_fn,
                   eqfun = equal, eqB = 1, LB = lower_bound,
                   UB = upper_bound)$par
  return(weights)
}
```

In the next section, we compare the Bayesian implementation of divergence-based model weighting to Bayesian stacking and pseudobma+. The latter two methods are introduced in Yao et al. (2018), who demonstrate

that both methods work better than a wide range of Bayesian model averaging methods, including standard Bayesian model averaging. Bayesian stacking is a Bayesian implementation of stacking with the log score, whereas pseudobma+ is a version of negative exponentiated model weighting that uses cross validation and bootstrapping to calculate model scores. Both methods rely on Pareto-smoothed importance sampling to approximate Bayesian leave-one-out cross validation. We refer the reader to Yao et al. (2018) for further details.

To calculate pseudobma+ weights we use the aforementioned LOO package (Vehtari et al., 2020). The LOO package can also be used to calculate stacking weights, but it tends to run into convergence issues, including in the simulations described in the next section.[5] We therefore use the following function instead, where "pointwise" in this case refers to the leave-k-out matrix of model predictions (or the LOO approximation of this matrix):

```
stacking_weights <- function(pointwise){
  num_of_models <- ncol(pointwise)
  par_start <- rep(1, num_of_models)
  par_start <- par_start/sum(par_start)
  lower_bound <- rep(0, num_of_models)
  upper_bound <- rep(1, num_of_models)
  equal <- function(x){
    sum(x)
  }
  optimizing_fn <- function(x){
    predictions <- pointwise%*%x
    score  <- -sum(log(predictions))
    return(score)
  }
  weights <- solnp(pars = par_start, fun = optimizing_fn,
                   eqfun = equal, eqB = 1, LB = lower_bound,
                   UB = upper_bound)$par
  return(weights)
}
```

B BAYESIAN SIMULATION EXPERIMENT

This section replicates the linear subset regression simulations in Yao et al. (2018), who in turn adapt the example from Breiman (1996). The set-up is as follows. First, covariates $X_1, X_2, \ldots, X_{15}$ are independently generated from $\mathcal{N}(5,1)$. Next, α_j coefficients are calculated in the following manner, for $j = 1, 2, \ldots, 15$:

$$\alpha_j = 1_{|j-4|<h}(h - |j-4|)^2 + 1_{|j-8|<h}(h - |j-8|)^2 1_{|j-12|<h}(h - |j-12|)^2 \tag{18}$$

Where h is a parameter that determines how many of the coefficients are "strong". Following Yao et al. (2018), we put $h = 5$, which yields 15 "weak" coefficients. Next, β_j coefficients are defined as follows: $\beta_j = \gamma\alpha_j$, where γ is chosen such that the standard deviation of $\sum_j \beta_j X_j$ is equal to 2. We also generate errors ϵ independently from $\mathcal{N}(0,1)$. Finally, we generate Y values as follows:

$$Y = \beta_1 X_1 + \ldots \beta_{15} X_{15} + \epsilon \tag{19}$$

Again following Yao et al. (2018), we consider two different model spaces. The first model space consists of all models of the form $\mathcal{M}_k : Y \sim \mathcal{N}(\beta_k X_k, \sigma^2)$ (i.e., each model contains a single predictor) and the second model space consists of all models of the form $\mathcal{M}_k : Y \sim \mathcal{N}(\sum_k \beta_j X_j, \sigma^2)$ (the models are increasingly larger subsets of the set of all predictors). For each model in both model spaces we use the following priors: $\beta_k \sim \mathcal{N}(0, 10)$ and $\sigma \sim \text{Gamma}(0.1, 0.1)$. Note that neither of these model spaces (particularly not the first one) would realistically be used in practice. However, the two model spaces are still useful for the purposes of evaluating model averaging

[5] Yuling Yao (personal communication) recommends using the function described in Yao et al. (2021), which uses Stan to perform the optimization. We have tried this and confirm that it gives the same results as our function.

techniques since they represent two extremes: the case where all the models under consideration are radically misspecified, and the case where one of the models is true.

We generate a training data set from the true data distribution 19 consisting of n data points, where n ranges from 5 to 200, and we generate a test data set consisting of 200 data points and calculate the mean log predictive density of each model averaging method. We repeat this simulation 100 times and average the results to estimate the expected mean log predictive densities.

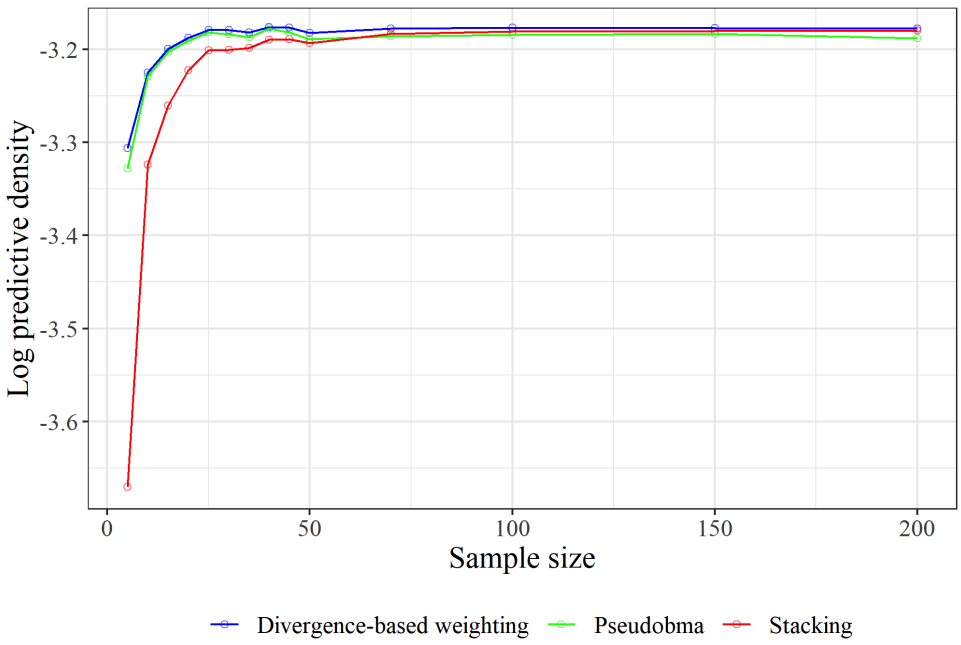

Figure 3: First simulation: each model consists of a single predictor.

Figure 3 compares divergence weighting, Bayesian stacking, and pseudobma+ on the first model space. Pseudobma+ is better than Bayesian stacking for sample sizes smaller than 100, but Bayesian stacking is asymptotically better than pseudobma+.[6] Divergence-based model weighting dominates both pseudobma+ and Bayesian stacking for small sample sizes in this experiment, and asymptotically behaves like Bayesian stacking.

Figure 4 compares the model weighting methods on the second model space. In this case, all three methods are about equally accurate for sample sizes larger than 25 and they all assign a probability of approximately 1 to the true model for large sample sizes. However, for very small sample sizes, divergence-based weighting is slightly more accurate than the alternatives.

C PROOFS OF THEOREMS

C.1 Proof of Theorem 3.1 (the characterization theorem) from the main document

For convenience, let us restate the boundary condition and the theorem. The boundary condition is:

$$\begin{aligned} &\arg \min_{w^p \in V^K} c \sum_k w_k^p f(\frac{w_k^{op}}{w_k^p}) - \sum_i \log \sum_k w_k^p p_k^p(y_i) \\ &= \arg\min_k \sum_i -\log p_k^p(y_i) + op_k \end{aligned} \tag{20}$$

[6] Our results here differ somewhat from those reported in Yao et al. (2018). Comparing Figure 3 in Yao et al. (2018) with our Figure 3, it is clear that the graphs for stacking are very similar, but pseudobma+ performs much better in our simulation than in Yao et al.'s (2018). We have not been able to determine the cause of this discrepancy.

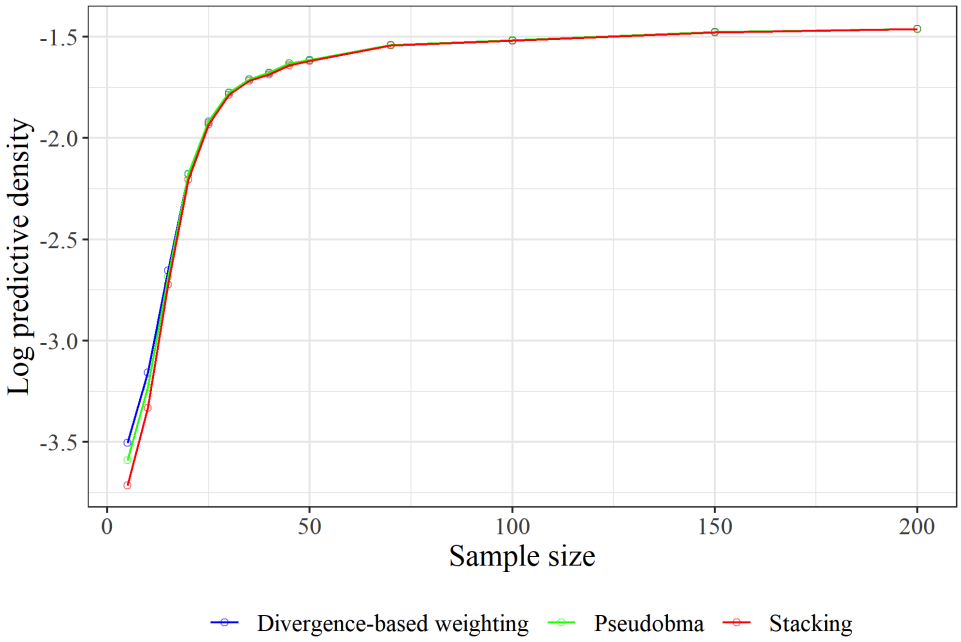

Figure 4: Second simulation: the models are increasingly larger subsets of the set of all predictors.

Where $\sum_k w_k^p f(\frac{w_k^{op}}{w_k^p})$ is an f-divergence with a real analytic generator f. I.e., f is a real analytic convex function such that $f(1) = 0$ and $\lim_{w\to 0} wf(\frac{1}{w}) = 0$.

And here is the theorem:

Theorem 3.1. *If we require that the boundary condition (13) be satisfied in all situations and that the generator f be real analytic, then:*

1. *The f-divergence in (11) must be the KL divergence.*
2. *The constant c in (11) must equal* 1.
3. *Consequently, (11) coincides with the divergence-based model weighting problem (3).*

Proof. If we have $w_k = 1$ for some k, then $c\sum_k w_k^p f(\frac{w_k^{op}}{w_k^p}) - \sum_i \log \sum_k w_k^p p_k^p(y_i)$ reduces to $cf(w_k^{op}) - \sum_i \log p_k^p(y_i)$. Hence, putting $S_k = -\sum_i \log p_k^p(y_i)$, the boundary condition (20) reduces to:

$$\begin{aligned} &\arg\min_k cf(w_k^{op}) + S_k \\ &= \arg\min_k op_k + S_k \end{aligned} \tag{21}$$

By the definition of w_k^{op}, we have: $op_k = -\log w_k^{op} + Z$, where $Z = -\log \sum_j e^{-op_j}$ does not depend on k and is therefore irrelevant in the optimization. Hence, we have:

$$\begin{aligned} &\arg\min_k cf(w_k^{op}) + S_k \\ &= \arg\min_k -\log w_k^{op} + S_k \end{aligned} \tag{22}$$

The fact that this holds in all situations means it holds for all $w^{op} \in S^K$ and all $S \in \mathbb{R}^K$. Now suppose there exist $w'^{op} \in S^K$ and all $S' \in \mathbb{R}^K$ such that there are indices m and n for which $cf({w'}_m^{op}) + {S'}_m \geq cf({w'}_n^{op}) + {S'}_n$

but $-\log {w'_m}^{op} + {S'}_m < -\log {w'_n}^{op} + {S'}_n$. Then, since we are free to vary S, we can create a new S'' with the same m and n indices as S', but with all other entries of S'' very large, such that

$$m = \arg\min_k \big(-\log {w''_k}^{op} + {S''}_k \big) \quad \text{and} \quad n = \arg\min_k \big(cf({w'_k}^{op}) + {S'}_k \big),$$

which contradicts (22). Hence, there cannot exist such indices m and n, and we conclude that (22) entails the following for all indices m and n:

$$cf(w_m^{op}) + S_m \geq cf(w_n^{op}) + S_n \iff -\log w_m^{op} + S_m \geq -\log w_n^{op} + S_n \tag{23}$$

Because $T = S_n - S_m$ can range freely in $\mathbb{R}$, 23 entails that the following holds for all $T \in \mathbb{R}$:

$$cf(w_m^{op}) - cf(w_n^{op}) \geq T \iff \log w_n^{op} - \log w_m^{op} \geq T \tag{24}$$

But this entails that $cf(w_m^{op}) - cf(w_n^{op}) = \log w_n^{op} - \log w_m^{op}$. Since it holds for all w^{op}, it must, in particular, hold when $w_n^{op} = 1$, and thus we get (since $f(1) = 0$ for f divergences:

$$cf(w_m^{op}) = -\log w_m^{op} \tag{25}$$

Since this holds for all possible values of w_m^{op}, and w_m^{op} can be any number in the interval $(0, 1)$, $cf(x)$ is identical to $\log x$ in the interval $(0, 1)$. We now use the fact that f is real analytic to conclude that $cf(x)$ is identical to $\log x$ for all x.[7] We can plug this into the boundary condition (20) and get:

$$\begin{aligned} &\arg\min_{w^p \in V^K} \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} - \sum_i \log \sum_k w_k^p p_k^p(y_i) \\ &= \arg\min_k \sum_i -\log p_k^p(y_i) + op_k \end{aligned} \tag{26}$$

And this establishes what we intended to show, i.e., $c = 1$ and the f-divergence is the KL divergence.

□

C.2 Proof of Theorem 3.2 and Theorem 3.3 from the main document

let us first define the key quantities. First, we define the pointwise "out-of-sample" Jensen gap as follows:

$$J_{out}(w^p) = -\sum_k w_k^p \log p_k^p(\tilde{y}_i) + \log \sum_k w_k^p p_k^p(\tilde{y}_i) \tag{27}$$

Next, we assume that the individual log losses are sub-Gaussian: there exists $s > 0$ such that for all $\lambda \in R$:

$$\log \mathrm{E}[e^{\lambda(-\log p_k^p(\tilde{y}) - \mathrm{E}[-\log p_k^p(y)])} \leq \frac{\lambda^2 s^2}{2} \tag{28}$$

We will now prove Lemma 3.2 from the main document:

Theorem 3.2. *If each of $-\log p_k^p(\tilde{y}_i)$ is sub-Gaussian, then there exists a variance s', such that for every set of posterior weights w^p, the Jensen gap $\sum_i J_{out}(w^p)$ is also sub-Gaussian with variance at most ns'.*

[7] But note that what we technically need is the much weaker condition that identity on $(0, 1)$ entails identity on the whole domain.

Proof. First, to simplify the notation, let:

$$X_k = -\log p_k^p(\tilde{y}) \tag{29}$$

$$\mu_k = \mathrm{E}[-\log p_k^p(\tilde{y})] \tag{30}$$

Then the sub-Gaussian assumption says that, for each k and all λ:

$$\log \mathrm{E}[e^{\lambda(X_k-\mu_k)}] \leq \frac{\lambda^2 s^2}{2} \tag{31}$$

Let

$$L = \sum_k w_k^p X_k \tag{32}$$

and

$$S = \log \sum_k w_k^p e^{X_k} \tag{33}$$

We will first prove that each of L and S is sub-Gaussian. First, since e^x is convex, Jensen's inequality implies:

$$\mathrm{E}[e^{\lambda(L-\mathrm{E}[L])}] = \mathrm{E}[e^{\sum_k w_k^p \lambda(X_k-\mu_k)}] \leq \sum_k w_k^p \mathrm{E}[e^{\lambda(X_k-\mu_k)}] \leq \frac{\lambda^2 s^2}{2} \tag{34}$$

Thus, L is sub-Gaussian with variance s. As for S, we have that:

$$\max_k X_k \leq S \leq \max_k X_k + \log K \tag{35}$$

Hence, S is trapped on both sides by $\max_k -X_k$ and a constant shift of $\max_k -X_k$. It follows that S is sub-Gaussian if $\max_k -X_k$ is sub-Gaussian, which it is. Indeed, we have (see Proposition 2.7.6 of Vershynin (2025)):

$$||\max_k X_k - \mathrm{E}[\max_k X_k]||_{\phi_2} \leq s\sqrt{\log K} \tag{36}$$

More carefully, (35) implies:

$$||(S - \max_k X_k) - E[(S - \max_k X_k)]||_{\phi_2} \leq \log K \tag{37}$$

And therefore, by the triangle inequality:

$$||S - E[S]||_{\phi_2} \leq ||\max_k X_k - \mathrm{E}[\max_k X_k]||_{\phi_2} + \log K = s\sqrt{\log K} + \log K \tag{38}$$

Hence, $S - E[S]$ is sub-Gaussian with some variance t.

Finally, we consider the (per sample) Jensen gap $L + S$:

$$\begin{aligned}\log \mathrm{E}[e^{\lambda((L+S)-\mathrm{E}[(L+S)])}] &= \mathrm{E}[e^{\lambda(L-\mathrm{E}[L])}e^{\lambda(S-\mathrm{E}[S])}] \\ &\leq \sqrt{\mathrm{E}[e^{2\lambda(L-\mathrm{E}[L])}e^{2\lambda(S-\mathrm{E}[S])}]} \\ &\leq 0.5\frac{(2\lambda)^2 s^2}{2} + 0.5\frac{(2\lambda)^2 t^2}{2} \\ &= 2s^2 + 2t^2\end{aligned}$$

Therefore, the pointwise Jensen gap is sub-Gaussian with variance $s'' = 2s^2 + 2t^2$. Finally, summing $L + S$ over i independent $\tilde{y}_i$ yields a sum of n independent subgaussian variables with variance s'', and therefore it has a variance of ns''.

□

We will now prove Theorem 3.3. We need the following slightly reworded version of corollary 4 from Germain et al. (2016) (see the reference for a complete proof):

Theorem C.1. *Suppose each model* $\log p_k^p(\tilde{y}_i)$ *is sub-Gaussian with variance parameter* s*. Then for any prior distribution* w_k *and for any* $\delta \in (0, 1]$*, with probability at least* $1-\delta$ *we have for all distributions* w_k^p *simultaneously:*

$$\begin{aligned}n\mathrm{E}[-\sum_k w_k^p \log p_k^p(y)] \leq &-\sum_i \sum_k w_k^p \log p_k^p(\tilde{y}_i) \\ &+\sum_k w_k^p \log \frac{w_k^p}{w_k} + \log\frac{1}{\delta} + n\frac{1}{2}s^2\end{aligned}$$

We also need the following fact about subgaussian variables:

Lemma C.2. *Suppose* $\sup_{w^p} \sum_i J_{out}(w^p)$ *is subgaussian with variance* ns''*. Then, for all* $\delta \in (0, 1]$*, with probability at least* $1 - \delta$*, the following holds for all* w^p *simultaneously:*

$$n\mathrm{E}[\sum_i J_{out}(w^p)] \leq \sum_i J_{out}(w^p)] + \sqrt{2ns'' \log\frac{1}{\delta}} \tag{39}$$

Here is the result we will prove:

Theorem 3.3. *Suppose each model* $-\log p_k^p(\tilde{y}_i)$ *is sub-Gaussian with variance parameter* s*, and let* ns'' *be defined as above. Then for any prior distribution* w_k *and any* $\delta \in (0, 1]$*, with probability at least* $1 - \delta$ *we have, simultaneously for all distributions* w_k^p*:*

$$\begin{aligned}&n\mathrm{E}[-\log\sum_k w_k^p p_k^p(y)] \\ \leq &\sum_k w_k^p \log\frac{w_k^p}{w_k} - \sum_i \log\sum_k w_k^p p_k^p(\tilde{y}_i) \\ &+n\frac{1}{2}s^2 + \log\frac{1}{\delta} + \sqrt{2ns'' \log\frac{1}{\delta}}\end{aligned}$$

Proof. Lemma C.2 implies that, for any $\delta \in (0, 1]$, with probability at least $1 - \delta$, the following holds:

$$n\mathrm{E}[-\log\sum_k w_k^p p_k^p(y) + \sum_k w_k^p \log p_k^p(y)] \leq -\sum_i \log\sum_k w_k^p p_k^p(\tilde{y}_i) + \sum_i\sum_k w_k^p \log p_k^p(\tilde{y}_i) + \sqrt{2ns'' \log\frac{1}{\delta}} \tag{40}$$

Rearranged, this implies:

$$n\mathrm{E}[-\log\sum_k w_k^p p_k^p(y)] \leq n\mathrm{E}[-\sum_k w_k^p \log p_k^p(y)] + \sum_i\sum_k w_k^p \log p_k^p(\tilde{y}_i) - \sum_i \log\sum_k w_k^p p_k^p(\tilde{y}_i) + \sqrt{2ns' \log\frac{1}{\delta}} \tag{41}$$

Theorem C.1 further implies that for any prior distribution w_k and for any $\delta \in (0,1]$, with probability at least $1-\delta$, the following holds for all distributions w^p simultaneously:

$$n\mathrm{E}[-\sum_k w_k^p \log p_k^p(y)] + \sum_i\sum_k w_k^p \log p_k^p(\tilde{y}_i) \leq \sum_k w_k^p \log\frac{w_k^p}{w_k} + \log\frac{1}{\delta} + n\frac{1}{2}s^2 \tag{42}$$

Putting 41 and 42 together yields that the following holds for any prior distribution w_k and for any $\delta \in (0,1]$, with probability at least $1-\delta$ for all distributions w_k^p simultaneously:

$$n\mathrm{E}[-\log\sum_k w_k^p p_k^p(y)] \leq \sum_k w_k^p \log\frac{w_k^p}{w_k} - \sum_i \log\sum_k w_k^p p_k^p(\tilde{y}_i) + n\frac{1}{2}s^2 + \log\frac{1}{\delta} + \sqrt{2ns'' \log\frac{1}{\delta}} \tag{43}$$

□

C.3 Proof of Theorem 3.4 from the main document

We will use the following theorem (Germain et al., 2016):

Theorem C.3. *For any prior distribution w_k and for any $\delta \in (0,1]$ and $c > 0$, with probability at least $1-\delta$ we have for all distributions w_k^p simultaneously:*

$$\begin{aligned} &n\mathrm{E}[-\sum_k w_k^p \log p_k^p(y)] \\ &\leq -\sum_i\sum_k w_k^p \log p_k^p(\tilde{y}_i) + \frac{1}{c}\sum_k w_k^p \log\frac{w_k^p}{w_k} \\ &\quad -\frac{1}{c}\log\delta + \phi_{term} \end{aligned}$$

Here, $\phi_{term} = \log\frac{1}{K}\sum_k \mathrm{E}[e^{\sum_i \log p_k^p(\tilde{y}_i) - n\mathrm{E}[\log p_k^p(y)]}]$. For simplicity, let $T = -\frac{1}{c}\log\delta + \phi_{term}$

Define the following quantities:

$$J_{IN} = -\sum_i\sum_k w_k^p \log p_k^p(y_i) + \sum_i \log\sum_k w_k^p p_k^p(y_i) \tag{44}$$

$$m_k = \frac{\sum_i -\log p_k^p(\tilde{y}_i)}{\sum_i -\log p_k^p(y_i)} \tag{45}$$

J_{IN} is the "in-sample" Jensen Gap calculated on the same data as was used to fit the models. m_k is a measure of the extent to which model k overfits on the data. Note that in general $m_k > 1$. Let $m = \min_k m_k$. In cases where all models under consideration overfit, m will tend to be much greater than 1. Using (C.3), we have (by applying Jensen's inequality on the first line and setting $c = 1$):

$$
\begin{aligned}
n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] &\leq n\mathrm{E}[-\sum_k w_k^p \log p_k^p(y)] \\
&\leq -\sum_i \sum_k w_k^p \log p_k^p(\tilde{y}_i) + \sum_k w_k^p \log \frac{w_k^p}{w_k} + T \\
&= -\sum_i \sum_k w_k^p \log p_k^p(y_i) + \sum_k w_k^p op_k + \sum_k w_k^p \log \frac{w_k^p}{w_k} + T \\
&= -\sum_i \sum_k w_k^p \log p_k^p(y_i) + \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} \\
&\quad -\sum_k w_k^p \log w_k - \log \sum_k e^{-op_k} + T \\
&= -\sum_i \log \sum_k w_k^p p_k^p(y_i) + \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} \\
&\quad + J_{IN} - \sum_k w_k^p \log w_k - \log \sum_k e^{-op_k} + T
\end{aligned} \tag{46}
$$

We also have:

$$
\begin{aligned}
J_{IN} &= -\sum_i \sum_k w_k^p \log p_k^p(\tilde{y}_i) + \sum_i \log \sum_k w_k^p p_k^p(y_i) \\
&\quad + \sum w_k^p \log w_k^p - \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} + \log \sum_k e^{-op_k} \\
&\geq -m \sum_i \sum_k w_k^p \log p_k^p(y_i) + \sum_i \log \sum_k w_k^p p_k^p(y_i) \\
&\quad + \sum w_k^p \log w_k^p - \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} + \log \sum_k e^{-op_k} \\
&= mJ_{IN} - (m-1) \sum_i \log \sum_k w_k^p p_k^p(y_i) + \sum w_k^p \log w_k^p \\
&\quad - \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} + \log \sum_k e^{-op_k}
\end{aligned} \tag{47}
$$

This implies:

$$
\begin{aligned}
(m-1)J_{IN} &\leq (m-1) \sum_i \log \sum_k w_k^p p_k^p(y_i) - \sum w_k^p \log w_k^p \\
&\quad + \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} - \log \sum_k e^{-op_k}
\end{aligned} \tag{48}
$$

Thus, combining (46) and (48), we have:

$$
\begin{aligned}
& n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] \\
\leq & -\sum_i \log \sum_k w_k^p p_k^p(y_i) + \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} \\
& + J_{IN} - \sum_k w_k^p \log w_k - \log \sum_k e^{-op_k} + T \\
\leq & \frac{m}{m-1} \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} \\
& - \frac{m}{m-1} \log \sum_k e^{-op_k} - \frac{1}{m-1} \sum w_k^p \log w_k^p \\
& - \sum_k w_k^p \log w_k + T
\end{aligned}
\tag{49}
$$

When m is large (i.e., the models overfit), we get, approximately (and setting $w_k = 1/K$):

$$
\begin{aligned}
& n\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] \leq \\
& \sum_k w_k^p \log \frac{w_k^p}{w_k^{op}} + \log K - \log \sum_k e^{-op_k} + T
\end{aligned}
$$

C.4 Proof of Theorem 3.5 From the Main Document

For convenience, we first restate Theorem 3.5 (here renumbered as Theorem 3.5):

Theorem 3.5. *For $k = 1, 2, \ldots, K$, let p_k^p be a predictive distribution that results from fitting model $\mathcal{M}_k$ to data $D = \{y_1, y_2, \ldots, y_n\}$ that are distributed i.i.d. according to some distribution Q. Let $\tilde{D} = \{\tilde{y_1}, \tilde{y_2}, \ldots, \tilde{y_n}\}$ also be i.i.d. from Q, let $op_k = -\sum_i \log p_k^p(\tilde{y}_i) + \sum_i \log p_k^p(y_i)$ be the optimism of each model. Let S^K be the set of all probability functions with K components. Suppose each p_k^p converges in probability to a distribution p_k^* in the sense that $\frac{1}{n}\sum_i |[\log p_k^*(y_i) - \log p_k^p(y_i)]| \xrightarrow{P} 0$ and $\frac{1}{n}\sum_i |[\log p_k^*(\tilde{y}_i) - \log p_k^p(\tilde{y}_i)]| \xrightarrow{P} 0$. Suppose also that all relevant integrals are finite. Then the following holds for all $w^p \in S^K$:*

$$
\begin{aligned}
& \frac{1}{n} | \sum_k w_k^p \log w_k^p + \sum_k w_k^p op_k - \sum_i \log \sum_k w_k^p p_k^p(y_i) \\
& - \mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] | \xrightarrow{P} 0
\end{aligned}
$$

Proof. We need the sum-log inequality:

Lemma C.4. *Suppose $A_1, A_2, \ldots, A_K$ and $B_1, B_2, \ldots, B_K$ are sequences of positive numbers. Then:*

$$
\log \frac{\sum_k A_k}{\sum_k B_k} \leq \sum_k \frac{A_k}{\sum_j A_j} \log \frac{A_k}{B_k}
$$

We can now prove Theorem 3.5.

We start by expanding $|\frac{1}{n}\sum_k w_k^p \log w_k^p + \frac{1}{n}\sum_k w_k^p op_k - \frac{1}{n}\sum_i \log \sum_k w_k^p p_k^p(y_i) - \mathrm{E}[-\log \sum_k w_k^p p_k^p(y)]|$ as follows:

$$
\begin{aligned}
|\frac{1}{n}\sum_k w_k^p \log w_k^p + \frac{1}{n}\sum_k w_k^p op_k - \frac{1}{n}\sum_i \log \sum_k w_k^p p_k^p(y_i) - \mathrm{E}[-\log \sum_k w_k^p p_k^p(y)]| \\
\leq |\frac{1}{n}\sum_k w_k^p \log w_k^p| \\
+|\frac{1}{n}\sum_k w_k^p op_k| \\
+\frac{1}{n}|\sum_i \log \sum_k w_k^p p_k^*(y_i) - \sum_i \log \sum_k w_k^p p_k^*(\tilde{y}_i)| \\
+\frac{1}{n}|\sum_i \log \sum_k w_k^p p_k^*(\tilde{y}_i) - \sum_i \log \sum_k w_k^p p_k^p(\tilde{y}_i)| \\
+\frac{1}{n}|\sum_i \log \sum_k w_k^p p_k^p(y_i) - \sum_i \log \sum_k w_k^p p_k^*(y_i)| \\
+|\mathrm{E}[-\log \sum_k w_k^p p_k^p(y)] - \frac{1}{n}\sum_i [-\log \sum_k w_k^p p_k^p(\tilde{y}_i)|
\end{aligned}
\tag{50}
$$

Every line after the inequality sign in 50 converges (in probability) to 0 as $n \to \infty$. First, $|\sum_k w_k^p \log w_k^p||$ is bounded for all w_k^p by $|\log K|$, so $\frac{1}{n}|\sum_k w_k^p \log w_k^p|$ simply converges to 0.

Second, we have:

$$
\begin{aligned}
\frac{1}{n}|\sum_k w_k^p op_k| = \frac{1}{n}|\sum_k w_k^p (\sum_i \log p_k^p(\tilde{y}_i) - \sum_i \log p_k^p(y_i))| \\
\leq \frac{1}{n}|\sum_i \sum_k w_k^p \log p_k^p(\tilde{y}_i) - \log p_k^p(y_i))| = \frac{1}{n}|\sum_i \sum_k w_k^p (\log p_k^p(\tilde{y}_i) - \log p_k^*(\tilde{y}_i) + \log p_k^*(\tilde{y}_i) \\
- \log p_k^*(y_i) + \log p_k^*(y_i) - \log p_k^p(y_i))| \\
\leq \frac{1}{n}|\sum_i \sum_k w_k^p (\log p_k^p(\tilde{y}_i) - \log p_k^*(\tilde{y}_i))| \\
+\frac{1}{n}|\sum_i \sum_k w_k^p (\log p_k^*(\tilde{y}_i) - \log p_k^*(y_i))| \\
+\frac{1}{n}|\sum_i \sum_k w_k^p (\log p_k^*(y_i) - \log p_k^p(y_i)))|
\end{aligned}
\tag{51}
$$

In the last three lines, the first and third converge (in probability) to 0 by the assumption of the theorem. The second term converges (in probability) to 0 by the Weak Law of Large Numbers. Hence, $\frac{1}{n}|\sum_k w_k^p op_k|$ converges (in probability) to 0.

Next, $|\frac{1}{n}\sum_i \log \sum_k w_k^p p_k^*(y_i) - \sum_i \log \sum_k w_k^p p_k^*(\tilde{y}_i)|$ converges (in probability) to 0 by the Weak Law of Large Numbers.

We next show that $\frac{1}{n}|\sum_i \log \sum_k w_k^p p_k^*(\tilde{y}_i) - \sum_i \log \sum_k w_k^p p_k^p(\tilde{y}_i)|$ converges (in probability) to 0 by using the log-sum probability.

Let $A_{ik} = \frac{w_k^p p_k^*(\tilde{y}_i)}{\sum_j w_j^p p_j^*(\tilde{y}_i)}$. Note that $0 \leq A_{ik} \leq 1$ and $\sum_k A_{ik} = 1$. First, assume $\sum_i \log \sum_k w_k^p p_k^*(\tilde{y}_i) - \sum_i \log \sum_k w_k^p p_k^p(\tilde{y}_i) \geq 0$. By the log-sum inequality, we have:

$$
\begin{aligned}
|\frac{1}{n}\sum_i \log\sum_k w_k^p p_k^*(\tilde{y}_i) - \frac{1}{n}\sum_i \log\sum_k w_k^p p_k^p(\tilde{y}_i)| \\
&= \frac{1}{n}|\sum_i \log\frac{\sum_k w_k^p p_k^*(\tilde{y}_i)}{\sum_k w_k^p p_k^p(\tilde{y}_i)}| \\
&\leq \frac{1}{n}|\sum_k\sum_i A_{ik}\log\frac{w_k^p p_k^*(\tilde{y}_i)}{w_k^p p_k^p(\tilde{y}_i)}| \\
&\leq \frac{1}{n}\sum_k\sum_i A_{ik}|\log\frac{p_k^*(\tilde{y}_i)}{p_k^p(\tilde{y}_i)}| \\
&\leq \frac{1}{n}\sum_k\sum_i |\log p_k^*(\tilde{y}_i) - \log p_k^p(\tilde{y}_i)|
\end{aligned}
\tag{52}
$$

The last line converges (in probability) to 0 by our theorem's assumption. If $\sum_i \log\sum_k w_k^p p_k^*(\tilde{y}_i) - \sum_i \log\sum_k w_k^p p_k^p(\tilde{y}_i) \leq 0$, we can do the same argument with the roles of $p_k^*(\tilde{y}_i)$ and $p_k^p(\tilde{y}_i)$ reversed and arrive at the same conclusion.

By exactly the same argument, we can show that $\frac{1}{n}|\sum_i \log\sum_k w_k^p p_k^p(y_i) - \sum_i \log\sum_k w_k^p p_k^*(y_i)|$ converges (in probability) to 0.

Finally, by the Weak Law of Large Numbers $|\mathrm{E}[-\log\sum_k w_k^p p_k^p(y)] - \frac{1}{n}\sum_i[-\log\sum_k w_k^p p_k^p(\tilde{y}_i)|$ converges (in probability) to 0.

Hence, we conclude that all the terms in (50) converge (in probability) to 0.

□

D A third justification of divergence-based model weighting: divergence-based model weighting as a modification of Bayesian model averaging

Bissiri et al. (2016) show that Bayesian updating may be regarded as the solution to a certain optimization problem, where—speaking loosely—the goal is to have a posterior distribution over the parameter values that achieves the dual goal of being close to the prior distribution (and therefore suitably skeptical and cautious) and also assigns a high posterior probability (density) to parameter values that are predictively accurate on the data. More formally, let $p^p(\theta)$ and $p(\theta)$ be the posterior and prior distributions over θ, respectively, and let S be the set of all probability distributions over θ. Then Bissiri et al. (2016) formulate the following optimization problem, which they show is uniquely minimized by the Bayesian posterior distribution:

$$
\min_{p^p\in S}\int p^p(\theta)\log\frac{p^p(\theta)}{p(\theta)}d\theta - \sum_i\int p^p(\theta)\log p(y_i|\theta)d\theta \tag{53}
$$

In fact, Bissiri et al. (2016) consider a modified version of 53, where $\log p(y_i|\theta)$ is replaced with a generic loss function. They argue that this is a legitimate move from a decision theoretic point of view, since 53 may be regarded as an expected loss that consists of two separate pieces: the loss that results from having a posterior distribution that differs from the prior distribution (quantified in the first term of 53 as the KL divergence (Kullback and Leibler, 1951) from the posterior to the prior); and the loss that results from assigning high posterior probabilities to parameter values that have a poor fit to the data (quantified in the second term of 53 as the (negative) log-likelihood function averaged over the posterior). If there is reason to think that the model is misspecified, then Bissiri et al. (2016) argue that there is no reason why predictive accuracy should necessarily be measured in terms of the log-likelihood. They furthermore show that when the log-likelihood is replaced with a different loss function, Bayesian updating may no longer be optimal.

Here, we also seek to modify the optimization problem in (53), but in a different direction. First, instead of considering the problem of assigning an optimal posterior distribution over a parameter in a model, we are instead interested in the problem of assigning a set of optimal posterior weights w_k^p over a set of statistical or

machine learning models, where—using the notation in (1)—each model has been fit to all the data, thereby producing a predictive distribution $p_k^p(y_i)$. Hence, we first reformulate (53) as follows:

$$\min_{w^p \in S^K} \sum_k w_k^p \log \frac{w_k^p}{w_k} - \sum_i \sum_k w_k^p \log p_k^p(y_i) \tag{54}$$

In (54), posterior model weights are rewarded (in the second term) to the extent that they assign a high probability to models that are individually predictively accurate (as measured by the log score). In fact, different Akaike style model weighting methods may be regarded as providing solutions to optimization problems of the form in (54). For example, let $w_k^{AIC} \propto e^{-(c_k + \frac{c_k(c_k+1)}{n-c_k-1})}$ be the prior model weights and let $p_k^p(y_i) = p_k(y_i|\hat{\theta})$, where $\hat{\theta}$ is the maximum likelihood of the parameter θ of model k. Plugging these terms into (54) yields the following optimization problem:

$$\min_{w^p \in S^K} \sum_k w_k^p \log \frac{w_k^p}{w_k^{AIC}} - \sum_i \sum_k \log p_k(y_i|\hat{\theta}) w_k^p \tag{55}$$

The general results in Bissiri et al. (2016) now imply that the posterior model weights that optimize (55) are proportional to $\prod_i p_k(y_i|\hat{\theta}) e^{-(c + \frac{c(c+1)}{n-c-1})}$, which are the small sample corrected form of the well known Akaike model weights (Akaike, 1979). Thus, Akaike model weighting and negative exponentiated model weighting more generally have a theoretical foundation in the general divergence framework presented in Bissiri et al. (2016). That is, we may regard negative exponentiated model weighting as a process that involves two steps: first, formulate a set of prior model weights that guard against overfitting (e.g., the aforementioned Akaike prior weights); second, choose the posterior model weights that optimize (54). We may therefore call the optimization problem in (54) the "negative exponentiated optimization problem."

Note, however, that in a model weighting and averaging context, the optimization problem in (54) does not really capture what we are after. Indeed, in a Bayesian context, Masegosa (2020) argues that optimizing (53) will often result in suboptimal predictions, and since (54) is a minor modification of (53) there is good reason to think the same will be true for (54). The main problem, which was pointed out by Minka (2002) in the context of Bayesian model averaging, is that negative exponentiated model weighting will tend to increasingly concentrate the probability on the single best model. This is clear from (54), because as the amount of data increases, $\sum_i \sum_k w_k^p \log p_k^p(y_i)$ will increasingly be dominated by the probability model that has the best log score on the data. However, what we would like is to reward posterior weights that produce accurate *combinations* of model predictions. Fortunately, this can be achieved in a simple way by replacing the second term of (54) with $\sum_{i=1}^n - \log \sum_{k=1}^K w_k^p p_k^p(y_i)$, i.e., we simply move the position of the log operator. A similar move is recommended by Morningstar et al. (2022) in a Bayesian context. Moving the log operator yields an objective function that is more sensible—given our aims—than (54), namely the divergence-based model weighting problem (1) from the main document. The divergence-based model weighting problem formally quantifies the dual goal of having a set of posterior model weights that are close to the prior weights (and therefore suitably skeptical) while also yielding a *linear combination* of model predictions that have a good fit to the data. Thus, in this way, divergence-based model weighting arises as a natural modification of negative exponentiated model weighting. Indeed, for small sample sizes, the KL divergence term—which is shared by the divergence-based model weighting optimization problem (1) and the negative exponentiated model weighting problem (54)—will be relatively influential, and hence we might expect the divergence-based model weights and the negative exponentiated model weights to be fairly similar. However, for large sample sizes, the KL divergence term will play a negligible role, and the divergence-based model weighting optimization problem (1) will be more similar to the stacking optimization problem (5) from the main document.

Given the close connection between the divergence-based optimization problem (1) and the Bayesian optimization problem (53) and its modification (54), we think it is natural to interpret the divergence-based model weights in a broadly Bayesian way: w_k^p reflects the (relative) posterior trust we place in model $\mathcal{M}_k$ for predictive purposes. As such, w_k^p should not be regarded as an uncertain estimate of some underlying "true" model weight, in the same way that Bayesian posterior probabilities are not typically regarded as imperfect estimates of "true" probabilities.

E Robustness checks

The aim of this section is to do various robustness checks. First, in Section 3.1, we argued that the KL-divergence is a well-motivated penalty term, but one might naturally wonder what were to happen if we used some other penalty. For example, what if we instead decided to use the quadratic (Brier) divergence (Brier, 1950), $\sum_k (w_k^p - w_k)^2$? Second, we argued that $c = 1$ is a natural default option, but one might be interested in knowing what happens if c deviates from 1. Third, we argued for the optimism-penalizing prior (8), but one might wonder how a default option of using a flat prior would fare.

In Figure 5, we explore what happens in the linear regression experiment from Section 4 when we implement these variations. The top figure shows what happens when c takes a value different from 1. As one might expect, $c = 2$ has a better accuracy for smaller sample sizes, whereas $c = 0.5$ has a slightly better accuracy for larger sample sizes. In the bottom left figure, we see what happens when we use the Brier divergence rather than KL divergence as the penalty term. The KL divergence performs substantially better. The bottom right figure compares the flat prior with the optimism-penalizing prior. The flat prior results in much worse predictions.

Another robustness check is to consider what happens when we use a data-generating distribution with a heavy-tailed error. Figure 6 shows the results after implementing the same linear regression experiment with an error distribution that follows a Student's t-distribution with 3 degrees of freedom.

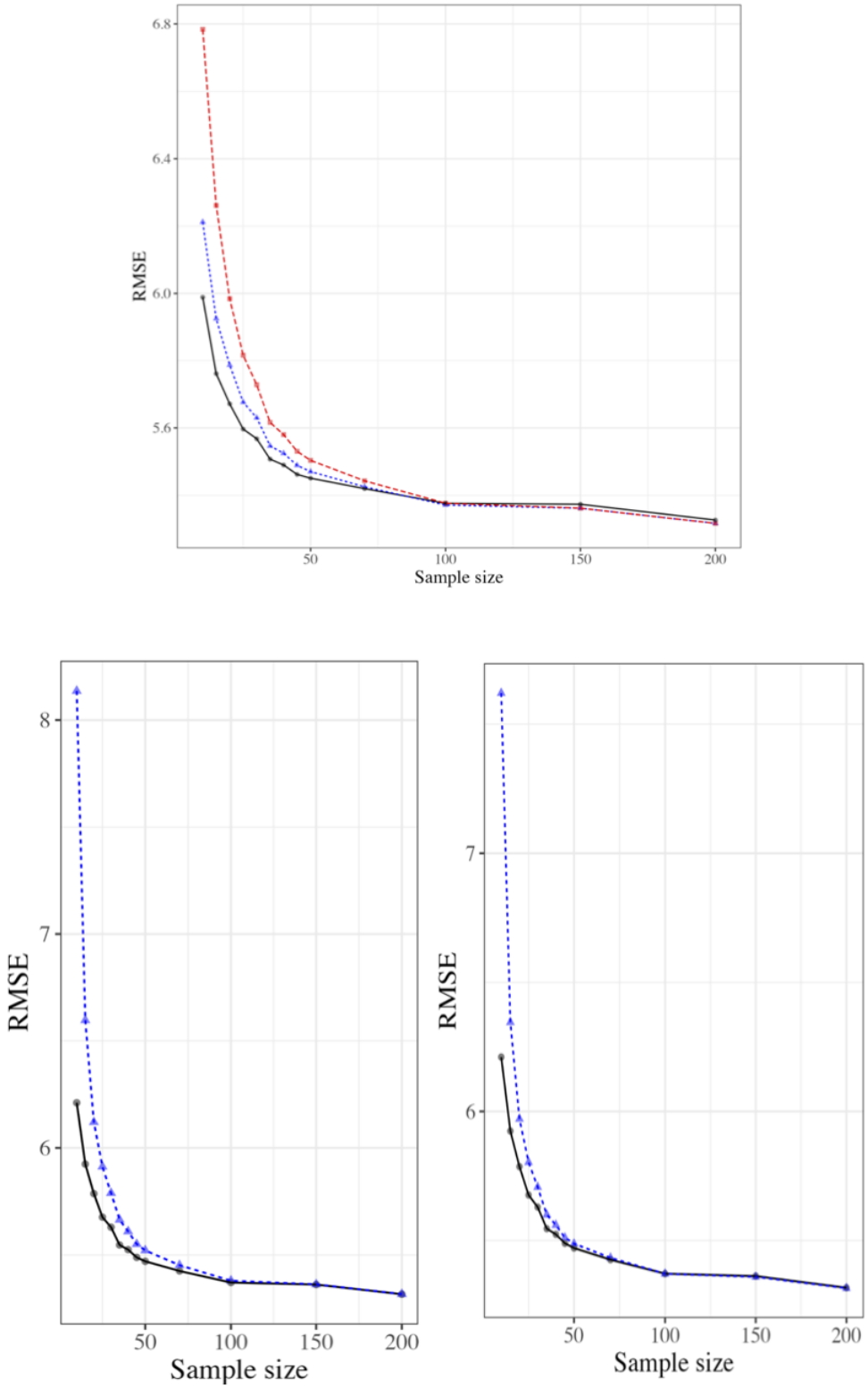

Figure 5: Top: c = 2 (black circles); c = 1 (blue triangles); c = 0.5 (red squares) Bottom left: Brier divergence (blue triangles); KL divergence (black circles) Bottom right: Uniform prior (blue triangles); optimism-penalizing prior (black circles)

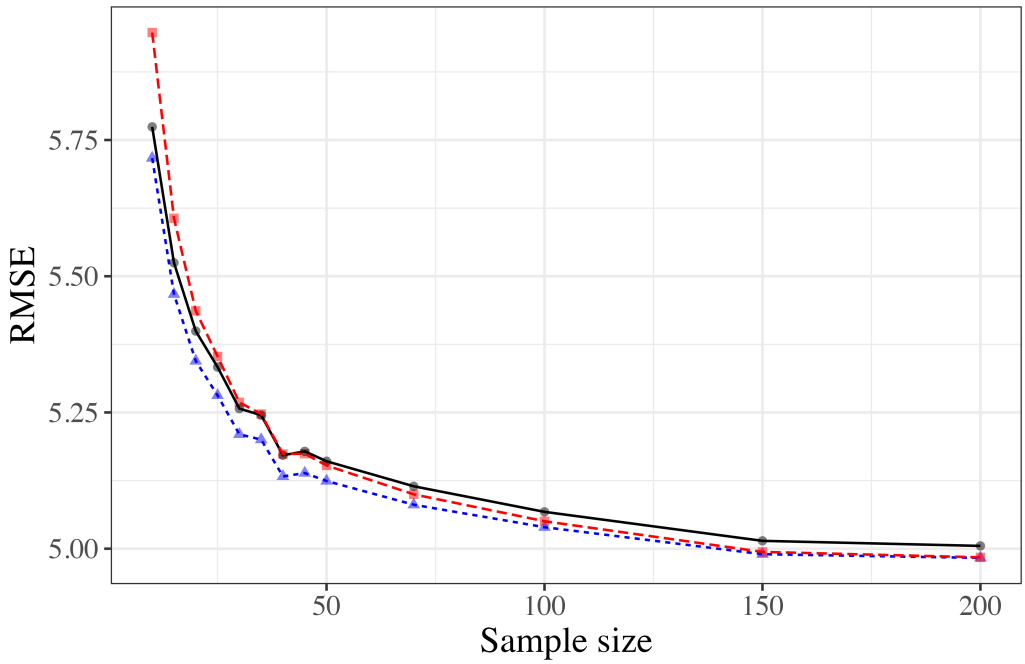

Figure 6: Blue dotted line: Divergence-based model weighting Red dotted line: Stacking with the log score Black solid line: Negative exponentiated model weighting