Draftable Comparison Export

This document is an exported comparison with limited functionality, generated by Draftable Desktop. To access full functionality, use Draftable's powerful comparison viewer in any of our products.

Left document:	2202.11678v1.pdf
Right document:	2202.11678v2.pdf

What is this document?

This is a comparison of two documents. The two documents are interleaved such that the left document is displayed on even pages and the right document is displayed on odd pages.

Is there a specific way I should view this file?

This document is intended to be viewed in Two Page Continuous mode (or sometimes called 'Two Page Scrolling'). It should open in this mode by default when using Adobe Acrobat and most popular PDF readers.

If the document opens in a different view, you can often change this in the settings. In Adobe Acrobat, go to **View > Page Display > Two Page Scrolling**.

Why are there blank pages?

Blank pages are inserted to keep both documents as aligned as much as possible.

How do I read the changes?

Text deleted from the left document and, hence, not in right document is highlighted red. Text added to the right document and, hence, not in left document is highlighted green.

Tip for printing

When printing this document, we recommend printing double-sided and include this first page. This will result in the matching text being displayed on different pages and easily readable, much like a book.

For more information

Draftable offers powerful document comparison solutions for all use-cases. To view our products, please visit our website: draftable.com.

Bayesian Model Selection, the Marginal Likelihood, and Generalization

Sanae Lotfi **Pavel Izmailov** **Gregory Benton** **Micah Goldblum** **Andrew Gordon Wilson**
New York University

Abstract

How do we compare between hypotheses that are entirely consistent with observations? The marginal likelihood (aka Bayesian evidence), which represents the probability of generating our observations from a prior, provides a distinctive approach to this foundational question, automatically encoding Occam's razor. Although it has been observed that the marginal likelihood can overfit and is sensitive to prior assumptions, its limitations for hyperparameter learning and discrete model comparison have not been thoroughly investigated. We first revisit the appealing properties of the marginal likelihood for learning constraints and hypothesis testing. We then highlight the conceptual and practical issues in using the marginal likelihood as a proxy for generalization. Namely, we show how marginal likelihood can be negatively correlated with generalization, with implications for neural architecture search, and can lead to both underfitting and overfitting in hyperparameter learning. We provide a partial remedy through a conditional marginal likelihood, which we show is more aligned with generalization, and practically valuable for large-scale hyperparameter learning, such as in deep kernel learning.

1. Introduction

The search for scientific truth is elusive. No matter how consistent a theory may be with all available data, it is always possible to propose an alternative theory that is equally consistent. Moreover, no theory is entirely correct: there will always be missed nuances, or phenomena we have not or cannot measure. To decide between different possible explanations, we heavily rely on a notion of *Occam's razor* — that the "simplest" explanation of data consistent with our observations is most likely to be true. For example, there are alternative theories of gravity to general relativity that are similarly consistent with observations, but general relativity is preferred because of its simplicity and intuitive appeal.

Jeffreys (1939), and many follow up works, showed that Occam's razor is not merely an ad-hoc rule of thumb, but a rigorous quantifiable consequence of probability theory. MacKay (2003, Chapter 28) arguably makes this point most clearly. Suppose we observe what appears to be a block behind a tree. If we had x-ray vision, perhaps we would see that there are in fact two blocks of equal height standing next to each other. The two block model can generate many more observations, but as a consequence, has to assign these observations lower probabilities. For what we do observe, the one block hypothesis is significantly more likely (see Figure 1(c)), even if we believe each hypothesis is equally likely before we observe the data. This probability of generating a dataset from a prior model is called the *marginal likelihood*, or *Bayesian evidence*. The marginal likelihood is widely applied to hypothesis testing, and *model selection*, where we wish to know which trained model is most likely to provide the best generalization. Marginal likelihood optimization has also been applied with great success for hyperparameter learning, where it is known as *empirical Bayes*, often outperforming cross-validation.

There is a strong polarization in the way marginal likelihood is treated. Advocates make compelling arguments about its philosophical benefits for hypothesis testing, its ability to learn constraints, and its practical successes, especially in Gaussian process kernel learning — often embracing the marginal likelihood as a nearly all-encompassing solution to model selection (e.g., MacKay, 1992c; Minka, 2001; Rasmussen and Williams, 2006; Wilson et al., 2016a). Critics tend to focus narrowly on its sensitivity to prior assumptions, without appreciating its many strengths (e.g., Domingos, 1999; Gelman, 2011; Gelman et al., 2013). There is a great need for a more comprehensive exposition, clearly demonstrating the limits of the marginal likelihood, while acknowledging its unique strengths, especially given the rise of the marginal likelihood in deep learning.

Rather than focus on a specific feature of the marginal likelihood, such as its sensitivity to the prior in isolation, in this paper we aim to fundamentally re-evaluate whether the marginal likelihood is the right metric for predicting the generalization of trained models, and learning hyperparameters. We argue that it does a good job of prior hypothesis testing, which is exactly aligned with the question it is designed to answer. However, we show that the marginal likelihood is only peripherally related to the question of which model we expect to generalize best after training, with significant implications for its use in model selection and hyperparameter learning.

We first highlight the strengths of the marginal likelihood, and its practical successes, in Section 3. We then de-

Bayesian Model Selection, the Marginal Likelihood, and Generalization

Sanae Lotfi **Pavel Izmailov** **Gregory Benton** **Micah Goldblum** **Andrew Gordon Wilson**
New York University

Abstract

How do we compare between hypotheses that are entirely consistent with observations? The marginal likelihood (aka Bayesian evidence), which represents the probability of generating our observations from a prior, provides a distinctive approach to this foundational question, automatically encoding Occam's razor. Although it has been observed that the marginal likelihood can overfit and is sensitive to prior assumptions, its limitations for hyperparameter learning and discrete model comparison have not been thoroughly investigated. We first revisit the appealing properties of the marginal likelihood for learning constraints and hypothesis testing. We then highlight the conceptual and practical issues in using the marginal likelihood as a proxy for generalization. Namely, we show how marginal likelihood can be negatively correlated with generalization, with implications for neural architecture search, and can lead to both underfitting and overfitting in hyperparameter learning. We provide a partial remedy through a conditional marginal likelihood, which we show is more aligned with generalization, and practically valuable for large-scale hyperparameter learning, such as in deep kernel learning.

1. Introduction

The search for scientific truth is elusive. No matter how consistent a theory may be with all available data, it is always possible to propose an alternative theory that is equally consistent. Moreover, no theory is entirely correct: there will always be missed nuances, or phenomena we have not or cannot measure. To decide between different possible explanations, we heavily rely on a notion of *Occam's razor* — that the "simplest" explanation of data consistent with our observations is most likely to be true. For example, there are alternative theories of gravity to general relativity that are similarly consistent with observations, but general relativity is preferred because of its simplicity and intuitive appeal.

Jeffreys (1939), and many follow up works, showed that Occam's razor is not merely an ad-hoc rule of thumb, but a rigorous quantifiable consequence of probability theory. MacKay (2003, Chapter 28) arguably makes this point most clearly. Suppose we observe what appears to be a block behind a tree. If we had x-ray vision, perhaps we would see that there are in fact two blocks of equal height standing next to each other. The two block model can generate many more observations, but as a consequence, has to assign these observations lower probabilities. For what we do observe, the one block hypothesis is significantly more likely (see Figure 1(c)), even if we believe each hypothesis is equally likely before we observe the data. This probability of generating a dataset from a prior model is called the *marginal likelihood*, or *Bayesian evidence*. The marginal likelihood is widely applied to hypothesis testing, and *model selection*, where we wish to know which trained model is most likely to provide the best generalization. Marginal likelihood optimization has also been applied with great success for hyperparameter learning, where it is known as *empirical Bayes*, often outperforming cross-validation.

There is a strong polarization in the way marginal likelihood is treated. Advocates make compelling arguments about its philosophical benefits for hypothesis testing, its ability to learn constraints, and its practical successes, especially in Gaussian process kernel learning — often embracing the marginal likelihood as a nearly all-encompassing solution to model selection (e.g., MacKay, 1992c; Minka, 2001; Rasmussen and Williams, 2006; Wilson et al., 2016a). Critics tend to focus narrowly on its sensitivity to prior assumptions, without appreciating its many strengths (e.g., Domingos, 1999; Gelman, 2011; Gelman et al., 2013). There is a great need for a more comprehensive exposition, clearly demonstrating the limits of the marginal likelihood, while acknowledging its unique strengths, especially given the rise of the marginal likelihood in deep learning.

Rather than focus on a specific feature of the marginal likelihood, such as its sensitivity to the prior in isolation, in this paper we aim to fundamentally re-evaluate whether the marginal likelihood is the right metric for predicting the generalization of trained models, and learning hyperparameters. We argue that it does a good job of prior hypothesis testing, which is exactly aligned with the question it is designed to answer. However, we show that the marginal likelihood is only peripherally related to the question of which model we

scribe several practical and philosophical issues in using the marginal likelihood for selecting between trained models in Section 4, and present a conditional marginal likelihood as a partial remedy for these issues. We exemplify these abstract considerations throughout the remainder of the paper, with several significant findings. We show that the marginal likelihood can lead to both underfitting and overfitting in data space, explaining the fundamental mechanisms behind each. In Section 5, we re-examine the relationship between the marginal likelihood and training efficiency, where we show that a conditional marginal likelihood, unlike the marginal likelihood, is correlated with generalization for both small and large datasizes. In Section 6, we demonstrate that the marginal likelihood can be negatively correlated with the generalization of trained neural network architectures. Finally, in Section 7 we show that the conditional marginal likelihood provides particularly promising performance for deep kernel hyperparameter learning. We make our code available here.

2. Related Work

As as early as Jeffreys (1939), it has been known that the log marginal likelihood (LML) encodes a notion of Occam's razor arising from the principles of probability, providing a foundational approach to hypothesis testing (Good, 1968; 1977; Jaynes, 1979; Gull, 1988; Smith and Spiegelhalter, 1980; Loredo, 1990; Berger and Jefferys, 1991; Jefferys and Berger, 1991; Kass and Raftery, 1995). In machine learning, Bayesian model selection was developed and popularized by the pioneering works of David MacKay (MacKay, 1992c;b;d;a). These works develop early Bayesian neural networks, and use a Laplace approximation of the LML for neural architecture design, and learning hyperparameters such as weight-decay (MacKay, 1992b; 1995).

In addition to the compelling philosophical arguments, the practical success of the marginal likelihood is reason alone to study it closely. For example, LML optimization is now the de facto procedure for kernel learning with Gaussian processes, working much better than other approaches such as standard cross-validation and covariogram fitting, and can be applied in many cases where these standard alternatives are simply intractable (e.g., Rasmussen and Williams, 2006; Wilson, 2014; Lloyd et al., 2014; Wilson et al., 2016a).

Moreover, in variational inference, the evidence lower bound (ELBO) to the LML is often used for automatically setting hyperparameters (Hoffman et al., 2013; Kingma and Welling, 2013; Kingma et al., 2015; Alemi et al., 2018). Notably, in variational auto-encoders (VAE), the whole decoder network (often, with millions of parameters) is treated as a model hyperparameter and is trained by maximizing the ELBO (Kingma and Welling, 2014).

Recently, the Laplace approximation (LA) and its use in marginal likelihood model selection has quickly regained popularity in Bayesian deep learning (Kirkpatrick et al., 2017; Ritter et al., 2018; Daxberger et al., 2021; Immer et al., 2021). Notably, Immer et al. (2021) use a scalable Laplace approximation of the marginal likelihood to predict which architectures will generalize best, and for automatically setting hyperparameters in deep learning, in the vein of MacKay (1992c), but with much larger networks.

MacKay (2003) uses the Laplace approximation to make connections between the marginal likelihood and minimum description length framework. MacKay (1995) also notes that structural risk minimization (Guyon et al., 1992) has the same scaling behaviour as the marginal likelihood. In recent years, PAC-Bayes (e.g., Alquier, 2021) has provided a popular framework for generalization bounds on stochastic networks, although it is distinct from the LML. In particular, PAC-Bayes bounds the expected generalization of a single posterior sample, whereas the LML measures the probability of generating the training data using the prior model average.

Critiques of the marginal likelihood often note its inability to manage improper priors for hypothesis testing, sensitivity to prior assumptions, lack of uncertainty representation over hyperparameters, and its potential misuse in advocating for models with fewer parameters (e.g., Domingos, 1999; Gelman et al., 2013; Gelman, 2011; Ober et al., 2021). To address such issues, Berger and Pericchi (1996) propose the *intrinsic Bayes factor* to enable Bayesian hypothesis testing with improper priors. Decomposing the LML into a sum over the data, Fong and Holmes (2020) use a similar measure to help reduce sensitivity to prior assumptions when comparing trained models. Lyle et al. (2020) also use this decomposition to suggest that LML is connected to training speed. Rasmussen and Ghahramani (2001) additionally note that the LML operates in function space, and can favour models with many parameters, as long as they do not induce a distribution over functions unlikely to generate the data.

Our work complements the current understanding of the LML, and has many features that distinguish it from prior work: (1) We provide a comprehensive treatment of the strengths and weaknesses of the LML across hypothesis testing, model selection, architecture search, and hyperparameter optimization; (2) While it has been noted that LML model selection can be sensitive to prior specification, we argue that the LML is answering an entirely different question than "will my trained model provide good generalization?", even if we have a reasonable prior; (3) We differentiate between LML hypothesis testing of fixed priors, and predicting which trained model will generalize best; (4) We also show that LML optimization can lead to *underfitting* or *overfitting* in function space; (5) We show the recent characterization in Lyle et al. (2020) that "models which

expect to generalize best after training, with significant implications for its use in model selection and hyperparameter learning.

We first highlight the strengths of the marginal likelihood, and its practical successes, in Section 3. We then describe several practical and philosophical issues in using the marginal likelihood for selecting between trained models in Section 4, and present a conditional marginal likelihood as a partial remedy for these issues. We exemplify these abstract considerations throughout the remainder of the paper, with several significant findings. We show that the marginal likelihood can lead to both underfitting and overfitting in data space, explaining the fundamental mechanisms behind each. In Section 5, we re-examine the relationship between the marginal likelihood and training efficiency, where we show that a conditional marginal likelihood, unlike the marginal likelihood, is correlated with generalization for a range of datasizes. In Section 6, we demonstrate that the marginal likelihood can be negatively correlated with the generalization of trained neural network architectures. Finally, in Section 7 we show that the conditional marginal likelihood provides particularly promising performance for deep kernel hyperparameter learning. We make our code available here.

2. Related Work

As as early as Jeffreys (1939), it has been known that the log marginal likelihood (LML) encodes a notion of Occam's razor arising from the principles of probability, providing a foundational approach to hypothesis testing (Good, 1968; 1977; Jaynes, 1979; Gull, 1988; Smith and Spiegelhalter, 1980; Loredo, 1990; Berger and Jeffreys, 1991; Jefferys and Berger, 1991; Kass and Raftery, 1995). In machine learning, Bayesian model selection was developed and popularized by the pioneering works of David MacKay (MacKay, 1992c;b;d;a). These works develop early Bayesian neural networks, and use a Laplace approximation of the LML for neural architecture design, and learning hyperparameters such as weight-decay (MacKay, 1992b; 1995).

In addition to the compelling philosophical arguments, the practical success of the marginal likelihood is reason alone to study it closely. For example, LML optimization is now the de facto procedure for kernel learning with Gaussian processes, working much better than other approaches such as standard cross-validation and covariogram fitting, and can be applied in many cases where these standard alternatives are simply intractable (e.g., Rasmussen and Williams, 2006; Wilson, 2014; Lloyd et al., 2014; Wilson et al., 2016a).

Moreover, in variational inference, the evidence lower bound (ELBO) to the LML is often used for automatically setting hyperparameters (Hoffman et al., 2013; Kingma and Welling, 2013; Kingma et al., 2015; Alemi et al., 2018). Notably, in variational auto-encoders (VAE), the whole decoder network (often, with millions of parameters) is treated as a model hyperparameter and is trained by maximizing the ELBO (Kingma and Welling, 2014).

Recently, the Laplace approximation (LA) and its use in marginal likelihood model selection has quickly regained popularity in Bayesian deep learning (Kirkpatrick et al., 2017; Ritter et al., 2018; Daxberger et al., 2021; Immer et al., 2021). Notably, Immer et al. (2021) use a scalable Laplace approximation of the marginal likelihood to predict which architectures will generalize best, and for automatically setting hyperparameters in deep learning, in the vein of MacKay (1992c), but with much larger networks.

MacKay (2003) uses the Laplace approximation to make connections between the marginal likelihood and minimum description length framework. MacKay (1995) also notes that structural risk minimization (Guyon et al., 1992) has the same scaling behaviour as the marginal likelihood. In recent years, PAC-Bayes (e.g., Alquier, 2021) has provided a popular framework for generalization bounds on stochastic networks, although it is distinct from the LML. In particular, PAC-Bayes bounds the expected generalization of a single posterior sample, whereas the LML measures the probability of generating the training data using the prior model average.

Critiques of the marginal likelihood often note its inability to manage improper priors for hypothesis testing, sensitivity to prior assumptions, lack of uncertainty representation over hyperparameters, and its potential misuse in advocating for models with fewer parameters (e.g., Domingos, 1999; Gelman et al., 2013; Gelman, 2011; Ober et al., 2021). To address such issues, Berger and Pericchi (1996) propose the *intrinsic Bayes factor* to enable Bayesian hypothesis testing with improper priors. Decomposing the LML into a sum over the data, Fong and Holmes (2020) use a similar measure to help reduce sensitivity to prior assumptions when comparing trained models. Lyle et al. (2020) also use this decomposition to suggest that LML is connected to training speed. Rasmussen and Ghahramani (2001) additionally note that the LML operates in function space, and can favour models with many parameters, as long as they do not induce a distribution over functions unlikely to generate the data.

Our work complements the current understanding of the LML, and has many features that distinguish it from prior work: (1) We provide a comprehensive treatment of the strengths and weaknesses of the LML across hypothesis testing, model selection, architecture search, and hyperparameter optimization; (2) While it has been noted that LML model selection can be sensitive to prior specification, we argue that the LML is answering an entirely different question than "will my trained model provide good generalization?", even if we have a reasonable prior; (3) We differentiate between LML hypothesis testing of fixed priors, and predicting

train faster will obtain a higher LML" is not generally true, and revisit the connection between LML and training efficiency; (6) We show that in modern deep learning, the Laplace LML is not well-suited for architecture search and hyperparameter learning despite its recent use; (7) We study a conditional LML (CLML), related to the metrics in Berger and Pericchi (1996) and Fong and Holmes (2020), but with a different rationale and application, where we show it provides compelling results for neural architecture comparison, deep kernel hyperparameter learning, and transfer learning.

3. The Case for the Marginal Likelihood

While we are primarily focused on exploring the limitations of the marginal likelihood, we emphasize that the marginal likelihood distinctively addresses foundational questions in hypothesis testing and constraint learning. By encoding a notion of Occam's razor, the marginal likelihood can outperform cross-validation, without intervention and using training data alone. Since we can directly take gradients of the marginal likelihood with respect to hyperparameters on the training data, it can also be applied where standard cross-validation cannot, for computational reasons.

Definition. The *marginal likelihood* is the probability that we would generate a dataset $\mathcal{D}$ with a model $\mathcal{M}$ if we randomly sample from a prior over its parameters $p(w)$: $p(\mathcal{D}|\mathcal{M}) = \int p(\mathcal{D}|\mathcal{M}, w)p(w|\mathcal{M})dw$. It is named the *marginal* likelihood, because it is a likelihood formed from marginalizing parameters w. It is also known as the *Bayesian evidence*. Maximizing the marginal likelihood is sometimes referred to as *empirical Bayes*, *type-II maximum likelihood estimation*, or *maximizing the evidence*. We can also decompose the marginal likelihood as $p(\mathcal{D}|\mathcal{M}) = \prod_i^n p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M})$, where it can equivalently be understood as how good the model is at predicting each data point in sequence given every data point before it.

Occam factors. In the definition of the marginal likelihood, the argument of the integral is $p(w|\mathcal{D}, \mathcal{M})$ up to a constant of proportionality. If we assume the posterior is relatively concentrated around $\hat{w} = \text{argmax}_w p(w|\mathcal{D}, \mathcal{M})$, then we can perform a rectangular approximation of the integral, as the height of the posterior times its width, $\sigma_{w|\mathcal{D}}$, to find $p(\mathcal{D}|\mathcal{M}) \approx p(\mathcal{D}|\hat{w}, \mathcal{M}) \cdot \frac{\sigma_{w|\mathcal{D}}}{\sigma_w}$, where $p(\mathcal{D}|\hat{w}, \mathcal{M})$ is the data fit and $\frac{\sigma_{w|\mathcal{D}}}{\sigma_w}$ is the *Occam factor* — the width of the posterior over the width of the prior. If the posterior contracts significantly from the prior, there will be a large Occam penalty, leading to a low LML.

Occam's Razor. The marginal likelihood automatically encapsulates a notion of Occam's razor, as in Figure 1(c). If a model can only generate a small number of datasets, it will generate those datasets with high probability, since the marginal likelihood is a normalized probability density. By the same reasoning, a model which can generate many datasets cannot assign significant probability density to all of them. For a given dataset, the marginal likelihood will automatically favour the most constrained model that is consistent with the data. For example, suppose we have $f_1(x, w) = w_1 x$, and $f_2(x, w) = \sum_{i=1}^{100} w_i x^i$, with $p(w) = \mathcal{N}(0, I)$ in both cases, and data given by a straight line with a particular slope. Both models have parameters consistent with the data, yet the first model is significantly more likely to generate this dataset from its prior over functions.

Hypothesis Testing. The marginal likelihood provides an elegant mechanism to select between fixed hypotheses, even if each hypothesis is entirely consistent with our observations, and the prior odds of these hypotheses are equal. For example, in the early twentieth century, it was believed that the correct explanation for the irregularities in Mercury's orbit was either an undiscovered planet, orbital debris, or a modification to Newtonian gravity, but not general relativity. Since the predictions of general relativity are unable to explain other possible orbital trajectories, and thus easy to falsify, but consistent with Mercury's orbit, Jefferys and Berger (1991) show it has a significantly higher marginal likelihood than the alternatives. We emphasize here we are comparing fixed *prior* hypotheses. We are not interested in how parameters of general relativity update based on orbital data, and then deciding whether the updated general relativity is the correct description of orbital trajectories.

Hyperparameter Learning. In practice, the LML is often used to learn hyperparameters of the prior to find $\text{argmax}_\theta p(\mathcal{D}|\theta)$ where $p(\mathcal{D}|\theta) = \int p(D|w)p(w|\theta)$. Gaussian processes (GPs) provide a particularly compelling demonstration of LML hyperparameter learning. The LML does not prefer a small RBF length-scale that would optimize the data fit. Instead, as we show empirically in Figure 21 (Appendix), the LML chooses a value that would make the distribution over functions likely to generate the training data. We note that the LML can be used to learn many such kernel parameters (Rasmussen and Williams, 2006; Wilson and Adams, 2013; Wilson et al., 2016a). Since we can take gradients of the LML with respect to these hypers using only training data, the LML can also be used where cross-validation would suffer from a curse of dimensionality.

Constraint Learning. Typical learning objectives like maximum likelihood are never incentivized to select for constraints, because a constrained model will be a special case of a more flexible model that is more free to increase likelihood. The LML, on the other hand, can provide a consistent estimator for such constraints, automatically selecting the most constrained solution that fits the data, and collapsing to the true value of the constraint in the limit of infinite observations, from training data alone. Bayesian PCA is a clear example of LML constraint learning (Minka,

which trained model will generalize best; (4) We also show that LML optimization can lead to *underfitting* or *overfitting* in function space; (5) We show the recent characterization in Lyle et al. (2020) that "models which train faster will obtain a higher LML" is not generally true, and revisit the connection between LML and training efficiency; (6) We show that in modern deep learning, the Laplace LML is not well-suited for architecture search and hyperparameter learning despite its recent use; (7) We study a conditional LML (CLML), related to the metrics in Berger and Pericchi (1996) and Fong and Holmes (2020), but with a different rationale and application. We are the first to consider the CLML for hyperparameter learning, model selection for neural networks, approximate inference, and classification. We also do not consider prior sensitivity a drawback of the LML, but argue instead that the LML is answering a fundamentally different question than whether a trained model provides good generalization, and contrast this setting with hypothesis testing. Compared to cross-validation, the CLML can more scalable and can be conveniently used to learn thousands of hyperparameters.

3. The Case for the Marginal Likelihood

While we are primarily focused on exploring the limitations of the marginal likelihood, we emphasize that the marginal likelihood distinctively addresses foundational questions in hypothesis testing and constraint learning. By encoding a notion of Occam's razor, the marginal likelihood can outperform cross-validation, without intervention and using training data alone. Since we can directly take gradients of the marginal likelihood with respect to hyperparameters on the training data, it can also be applied where standard cross-validation cannot, for computational reasons.

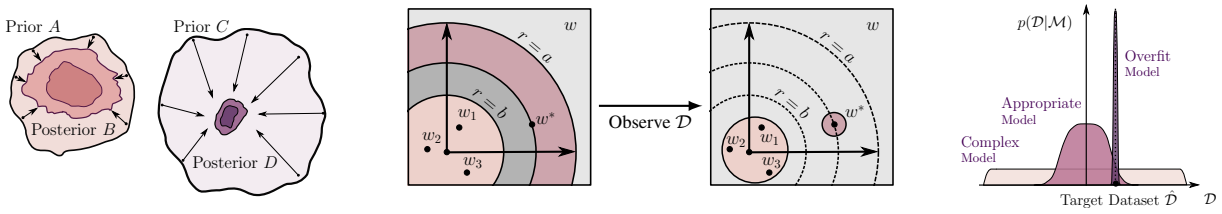
Definition. The *marginal likelihood* is the probability that we would generate a dataset $\mathcal{D}$ with a model $\mathcal{M}$ if we randomly sample from a prior over its parameters $p(w)$: $p(\mathcal{D}|\mathcal{M}) = \int p(\mathcal{D}|\mathcal{M}, w)p(w|\mathcal{M})dw$. It is named the *marginal* likelihood, because it is a likelihood formed from marginalizing parameters w. It is also known as the *Bayesian evidence*. Maximizing the marginal likelihood is sometimes referred to as *empirical Bayes*, *type-II maximum likelihood estimation*, or *maximizing the evidence*. We can also decompose the marginal likelihood as $p(\mathcal{D}|\mathcal{M}) = \prod_i^n p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M})$, where it can equivalently be understood as how good the model is at predicting each data point in sequence given every data point before it.

Occam factors. In the definition of the marginal likelihood, the argument of the integral is $p(w|\mathcal{D}, \mathcal{M})$ up to a constant of proportionality. If we assume the posterior is relatively concentrated around $\hat{w} = \text{argmax}_w p(w|\mathcal{D}, \mathcal{M})$, then we can perform a rectangular approximation of the integral, as the height of the posterior times its width, $\sigma_{w|\mathcal{D}}$, to find $p(\mathcal{D}|\mathcal{M}) \approx p(\mathcal{D}|\hat{w}, \mathcal{M}) \cdot \frac{\sigma_{w|\mathcal{D}}}{\sigma_w}$, where $p(\mathcal{D}|\hat{w}, \mathcal{M})$ is the data fit and $\frac{\sigma_{w|\mathcal{D}}}{\sigma_w}$ is the *Occam factor* — the width of the posterior over the width of the prior. If the posterior contracts significantly from the prior, there will be a large Occam penalty, leading to a low LML.

Occam's Razor. The marginal likelihood automatically encapsulates a notion of Occam's razor, as in Figure 1(c). If a model can only generate a small number of datasets, it will generate those datasets with high probability, since the marginal likelihood is a normalized probability density. By the same reasoning, a model which can generate many datasets cannot assign significant probability density to all of them. For a given dataset, the marginal likelihood will automatically favour the most constrained model that is consistent with the data. For example, suppose we have $f_1(x, w) = w_1 x$, and $f_2(x, w) = \sum_{i=1}^{100} w_i x^i$, with $p(w) = \mathcal{N}(0, I)$ in both cases, and data given by a straight line with a particular slope. Both models have parameters consistent with the data, yet the first model is significantly more likely to generate this dataset from its prior over functions.

Hypothesis Testing. The marginal likelihood provides an elegant mechanism to select between fixed hypotheses, even if each hypothesis is entirely consistent with our observations, and the prior odds of these hypotheses are equal. For example, in the early twentieth century, it was believed that the correct explanation for the irregularities in Mercury's orbit was either an undiscovered planet, orbital debris, or a modification to Newtonian gravity, but not general relativity. Since the predictions of general relativity are unable to explain other possible orbital trajectories, and thus easy to falsify, but consistent with Mercury's orbit, Jefferys and Berger (1991) show it has a significantly higher marginal likelihood than the alternatives. We emphasize here we are comparing fixed *prior* hypotheses. We are not interested in how parameters of general relativity update based on orbital data, and then deciding whether the updated general relativity is the correct description of orbital trajectories.

Hyperparameter Learning. In practice, the LML is often used to learn hyperparameters of the prior to find $\text{argmax}_\theta p(\mathcal{D}|\theta)$ where $p(\mathcal{D}|\theta) = \int p(D|w)p(w|\theta)dw$. Gaussian processes (GPs) provide a particularly compelling demonstration of LML hyperparameter learning. The LML does not prefer a small RBF length-scale that would optimize the data fit. Instead, as we show empirically in Figure 22 (Appendix), the LML chooses a value that would make the distribution over functions likely to generate the training data. We note that the LML can be used to learn many such kernel parameters (Rasmussen and Williams, 2006; Wilson and Adams, 2013; Wilson et al., 2016a). Since we can take gradients of the LML with respect to these hypers using only training data, the LML can also be used where cross-validation would suffer from a curse of dimensionality.



(a) Posterior contraction (b) Marginal likelihood underfitting (c) Marginal likelihood overfitting

Figure 1. **Pitfalls of marginal likelihood. (a)**: Prior B is vague, but contains easily identifiable solutions and quickly collapses to posterior D after observing a small number of datapoints. Prior A describes the data better than prior B, but posterior D describes the data better than posterior B. The marginal likelihood will prefer model A, but model C generalizes better. **(b)**: Example of misalignment between marginal likelihood and generalization. The marginal likelihood will pick prior scale b, and not include the best solution w^*, leading to suboptimal generalization performance. **(c)**: The complex model spreads its mass thinly on a broad support, while the appropriate model concentrates its mass on a particular class of problems. The overfit model is a δ-distribution on the target dataset $\hat{\mathcal{D}}$.

2001). Suppose the data are generated from a linear subspace, plus noise. While maximum likelihood always selects for the largest possible subspace dimensionality, and cross-validation tends to be cumbersome and inaccurate, the LML provides a consistent and practically effective estimator for the true dimensionality.

4. Pitfalls of the Marginal Likelihood

We now discuss general conceptual challenges in working with the marginal likelihood, and present the conditional marginal likelihood as a partial remedy. The remainder of this paper concretely exemplifies each of these challenges.

4.1. Marginal Likelihood is not Generalization

The marginal likelihood answers the question *"what is the probability that a prior model generated the training data?"*. This question is subtly different from asking *"how likely is the posterior, conditioned on the training data, to have generated withheld points drawn from the same distribution?"*. Although the marginal likelihood is often used as a proxy for generalization (e.g. MacKay, 1992c; Immer et al., 2021; Daxberger et al., 2021), it is the latter question we wish to answer in deciding whether a model will provide good generalization performance.

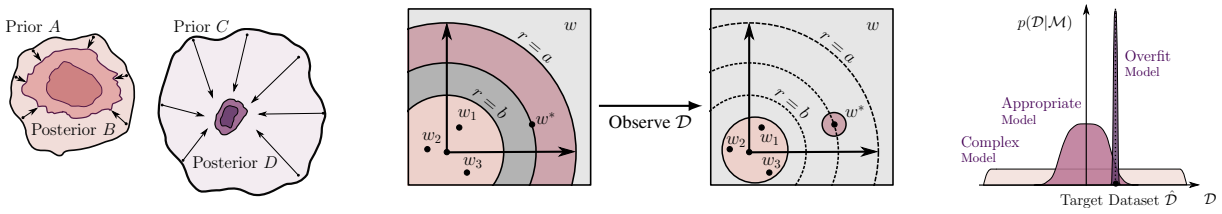
Indeed, if after observing data, prior A leads to posterior B, and prior C leads to posterior D, it can be the case that the same data are *less* probable under B than D, and also that D provides better generalization on fresh points from the same distribution, *even if the prior A explains the data better than C*. Consider, for example, the situation where we have a prior over a diffuse set of solutions which are easily identifiable from the data. We will then observe significant posterior contraction, as many of these solutions provide poor likelihood. While the marginal likelihood will be poor, the posterior could be perfectly reasonable for making predictions: in the product decomposition of the marginal likelihood in Section 3, the first terms will have low probability density, even if the posterior updates quickly to become a good description of the data. A different prior, which allocates significant mass to moderately consistent solutions, could then give rise to a much higher marginal likelihood, but a posterior which provides poorer generalization. We illustrate this effect in Figure 1(a) and provide concrete examples in Section 5.

There are several ways of understanding why the marginal likelihood will be poor in this instance: (1) the diffuse prior is unlikely to generate the data we observe, since it allocates significant mass to generating other datasets; (2) we pay a significant Occam factor penalty, which is the width of the posterior over the width of the prior, in the posterior contraction; (3) in the product decomposition of the marginal likelihood in Section 3, the first terms will have low probability density, even if the posterior updates quickly to become a good description of the data.

Model Selection. In hypothesis testing, our interest is in evaluating priors, whereas in *model selection* we wish to evaluate posteriors. In other words, in model selection we are not interested in a fixed hypothesis class A corresponding to a prior (such as the theory of general relativity in the example of Section 3), but instead the posterior B that arises when A is combined with data. Marginal likelihood is answering the question most pertinent to hypothesis testing, but is not generally well-aligned with model selection. We provide several examples in Sections 5, 6.

4.2. Marginal Likelihood Optimization and Overfitting

Marginal likelihood optimization for hyperparameter learning, also known as *type-II maximum likelihood* or *empirical Bayes*, is a special case of model selection. In this setting, we are typically comparing between many models — often a continuous spectrum of models — corresponding to different hyperparameter settings. In practice the marginal



(a) Posterior contraction (b) Marginal likelihood underfitting (c) Marginal likelihood overfitting

Figure 1. **Pitfalls of marginal likelihood. (a)**: Prior B is vague, but contains easily identifiable solutions and quickly collapses to posterior D after observing a small number of datapoints. Prior A describes the data better than prior B, but posterior D describes the data better than posterior B. The marginal likelihood will prefer model A, but model C generalizes better. **(b)**: Example of misalignment between marginal likelihood and generalization. The marginal likelihood will pick prior scale b, and not include the best solution w^*, leading to suboptimal generalization performance. **(c)**: The complex model spreads its mass thinly on a broad support, while the appropriate model concentrates its mass on a particular class of problems. The overfit model is a δ-distribution on the target dataset $\hat{\mathcal{D}}$.

Constraint Learning. Typical learning objectives like maximum likelihood are never incentivized to select for constraints, because a constrained model will be a special case of a more flexible model that is more free to increase likelihood. The LML, on the other hand, can provide a consistent estimator for such constraints, automatically selecting the most constrained solution that fits the data, and collapsing to the true value of the constraint in the limit of infinite observations, from training data alone. Bayesian PCA is a clear example of LML constraint learning (Minka, 2001). Suppose the data are generated from a linear subspace, plus noise. While maximum likelihood always selects for the largest possible subspace dimensionality, and cross-validation tends to be cumbersome and inaccurate, the LML provides a consistent and practically effective estimator for the true dimensionality. Another clear example is in automatically learning symmetries, such as rotation invariance (van der Wilk et al., 2018).

4. Pitfalls of the Marginal Likelihood

We now discuss general conceptual challenges in working with the marginal likelihood, and present the conditional marginal likelihood as a partial remedy. The remainder of this paper concretely exemplifies each of these challenges.

4.1. Marginal Likelihood is not Generalization

The marginal likelihood answers the question *"what is the probability that a prior model generated the training data?"*. This question is subtly different from asking *"how likely is the posterior, conditioned on the training data, to have generated withheld points drawn from the same distribution?"*. Although the marginal likelihood is often used as a proxy for generalization (e.g. MacKay, 1992c; Immer et al., 2021; Daxberger et al., 2021), it is the latter question we wish to answer in deciding whether a model will provide good generalization performance.

Indeed, if after observing data, prior A leads to posterior B, and prior C leads to posterior D, it can be the case that the same data are *less* probable under B than D, and also that D provides better generalization on fresh points from the same distribution, *even if the prior A explains the data better than C*. Consider, for example, the situation where we have a prior over a diffuse set of solutions which are easily identifiable from the data. We will then observe significant posterior contraction, as many of these solutions provide poor likelihood. While the marginal likelihood will be poor, the posterior could be perfectly reasonable for making predictions: in the product decomposition of the marginal likelihood in Section 3, the first terms will have low probability density, even if the posterior updates quickly to become a good description of the data. A different prior, which allocates significant mass to moderately consistent solutions, could then give rise to a much higher marginal likelihood, but a posterior which provides poorer generalization. We illustrate this effect in Figure 1(a) and provide concrete examples in Section 5.

There are several ways of understanding why the marginal likelihood will be poor in this instance: (1) the diffuse prior is unlikely to generate the data we observe, since it allocates significant mass to generating other datasets; (2) we pay a significant Occam factor penalty, which is the width of the posterior over the width of the prior, in the posterior contraction; (3) in the product decomposition of the marginal likelihood in Section 3, the first terms will have low probability density, even if the posterior updates quickly to become a good description of the data.

Model Selection. In hypothesis testing, our interest is in evaluating priors, whereas in *model selection* we wish to evaluate posteriors. In other words, in model selection we are not interested in a fixed hypothesis class A corresponding to a prior (such as the theory of general relativity in the example of Section 3), but instead the posterior B that arises when A is combined with data. Marginal likelihood is

likelihood can be effective for tuning hyperparameters, as discussed in Section 3. However, marginal likelihood optimization can be prone to both underfitting and overfitting.

Overfitting by ignoring uncertainty. We can overfit the marginal likelihood, as we can overfit the likelihood. Indeed, a likelihood for one model can always be seen as a marginal likelihood for another model. For example, suppose we include in our search space a prior model concentrated around a severely overfit maximum likelihood solution. Such a model would be "simple" in that it is extremely constrained — it can essentially only generate the dataset under consideration — and would thus achieve high marginal likelihood, but would provide poor generalization (Figure 1(c)).

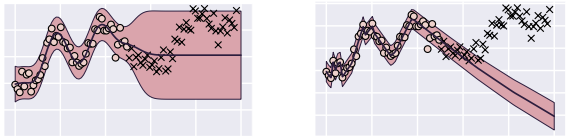


Figure 2. **LML overfitting in Gaussian Processes. Left:** A fit of a GP regression model with a constant prior mean. **Right:** a prior mean parameterized with an MLP and trained via marginal likelihood optimization. Train data is shown with circles, test data is shown with crosses and the shaded region visualizes the 2σ-predictive region of the GP. Given enough flexibility with the prior mean, the marginal likelihood overfits the data, providing poor overconfident predictions outside of the train region.

As an example, we parameterize the mean function in an RBF GP with a small multi-layer perceptron (MLP) and learn the parameters of the MLP by optimizing the LML. We show the results in Figure 2, where the learned mean function overfits the train data, leading to poor and overconfident predictions outside of the train region. We note that the mean of a GP does not appear in the Occam factor of the marginal likelihood. Therefore the Occam factor does not directly influence neural network hyperparameter learning in this instance, which is different from deep kernel learning (Wilson et al., 2016b). We provide additional details in Appendix C.

While it may not appear surprising that we can overfit the marginal likelihood, the narratives relating the marginal likelihood to Occam's razor often give the impression that it is safe to expand our model search, and that by favouring a "constrained model", we are protected from conventional overfitting. For example, Iwata and Ghahramani (2017) proposed a model analogous to the example in Figure 2. where a neural network serves as a mean function of a GP and argued that "since the proposed method is based on Bayesian inference, it can help alleviate overfitting". Furthermore, MacKay (1992c, Chapter 3.4) argues that if the correlation between the marginal likelihood and generalization is poor for a set of models under consideration, then we should expand our search to find new model structures that can achieve better marginal likelihood. While a mismatch between generalization and marginal likelihood can indicate that we should revisit our modelling assumptions, this advice could easily lead to overfitting.

Underfitting in hyperparameter selection. The above example involves overfitting that arises by ignoring uncertainty. The marginal likelihood also has a bias towards underfitting. This bias arises because supporting a good solution could involve also supporting many solutions that are unlikely to generate the training data from the prior. As an example, consider a zero-centred Gaussian prior on a set of parameters, $p(w) = \mathcal{N}(0, \sigma^2 I)$. Now suppose the parameters $w^\star$ that provide the best generalization have large norm, $\|w^\star\|$, but there are several settings of the parameters that provide moderate fits to the data with smaller norms $b \ll \|w^\star\|$. Further suppose that parameters with norms $b < \|w\| < \|w^\star\|$ provide very poor fits to the data. The marginal likelihood will not favour a large value of σ that makes $w^\star$ likely under the prior — even though such a value could lead to a posterior with much better generalization, as in Figure 1(b). With more data, the likelihood signal for $w^\star$ will dominate, and the underfitting bias disappears.

4.3. Laplace Approximation in Deep Learning

Outside of a few special cases, such as Gaussian process regression, the marginal likelihood is intractable. Because the marginal likelihood is integrating with the respect to the prior, and we thus cannot effectively perform simple Monte Carlo, it is also much harder to approximate than the posterior predictive distribution. Moreover, modern neural networks contain millions of parameters, leaving few practical options. The *Laplace approximation* (LA) for model selection in Bayesian neural networks was originally proposed by MacKay (1992c), and has recently seen a resurgence of popularity (e.g. Immer et al., 2021). Moreover, several generalization metrics, such as the Aikake Information Criterion (AIC) and Bayesian Information Criterion (BIC), can be derived by further approximating the Laplace approximate marginal likelihood (Bishop, 2006a).

The Laplace approximation represents the parameter posterior with a Gaussian distribution centred at a local optima of the posterior, w_{MAP}, with covariance matrix given by the inverse Hessian at w_{MAP}.

Drawbacks of the Laplace approximation. The actual posterior distribution for a modern neural network is highly multimodal. By representing only a single mode, the Laplace approximation provides a poor representation of the true Occam factor, which is the posterior volume divided by the prior volume. As a consequence, the Laplace marginal likelihood will overly penalize diffuse priors that capture multiple reasonable parameter settings across differ-

answering the question most pertinent to hypothesis testing, but is not generally well-aligned with model selection. We provide several examples in Sections 5, 6.

4.2. Marginal Likelihood Optimization and Overfitting

Marginal likelihood optimization for hyperparameter learning, also known as *type-II maximum likelihood* or *empirical Bayes*, is a special case of model selection. In this setting, we are typically comparing between many models — often a continuous spectrum of models — corresponding to different hyperparameter settings. In practice the marginal likelihood can be effective for tuning hyperparameters, as discussed in Section 3. However, marginal likelihood optimization can be prone to both underfitting and overfitting.

Overfitting by ignoring uncertainty. We can overfit the marginal likelihood, as we can overfit the likelihood. Indeed, a likelihood for one model can always be seen as a marginal likelihood for another model. For example, suppose we include in our search space a prior model concentrated around a severely overfit maximum likelihood solution. Such a model would be "simple" in that it is extremely constrained — it can essentially only generate the dataset under consideration — and would thus achieve high marginal likelihood, but would provide poor generalization (Figure 1(c)).

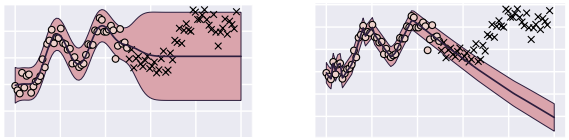


Figure 2. **LML overfitting in Gaussian Processes. Left:** A fit of a GP regression model with a constant prior mean. **Right:** a prior mean parameterized with an MLP and trained via marginal likelihood optimization. Train data is shown with circles, test data is shown with crosses and the shaded region visualizes the 2σ-predictive region of the GP. Given enough flexibility with the prior mean, the marginal likelihood overfits the data, providing poor overconfident predictions outside of the train region.

As an example, we parameterize the mean function in an RBF GP with a small multi-layer perceptron (MLP) and learn the parameters of the MLP by optimizing the LML. We show the results in Figure 2, where the learned mean function overfits the train data, leading to poor and overconfident predictions outside of the train region. We note that the mean of a GP does not appear in the Occam factor of the marginal likelihood. Therefore the Occam factor does not directly influence neural network hyperparameter learning in this instance, which is different from deep kernel learning (Wilson et al., 2016b). We provide additional details in Appendix C.

While it may not appear surprising that we can overfit the marginal likelihood, the narratives relating the marginal likelihood to Occam's razor often give the impression that it is safe to expand our model search, and that by favouring a "constrained model", we are protected from conventional overfitting. For example, Iwata and Ghahramani (2017) proposed a model analogous to the example in Figure 2. where a neural network serves as a mean function of a GP and argued that "since the proposed method is based on Bayesian inference, it can help alleviate overfitting". Furthermore, MacKay (1992c, Chapter 3.4) argues that if the correlation between the marginal likelihood and generalization is poor for a set of models under consideration, then we should expand our search to find new model structures that can achieve better marginal likelihood. While a mismatch between generalization and marginal likelihood can indicate that we should revisit our modelling assumptions, this advice could easily lead to overfitting.

Underfitting in hyperparameter selection. The above example involves overfitting that arises by ignoring uncertainty. The marginal likelihood also has a bias towards underfitting. This bias arises because supporting a good solution could involve also supporting many solutions that are unlikely to generate the training data from the prior. As an example, consider a zero-centred Gaussian prior on a set of parameters, $p(w) = \mathcal{N}(0, \sigma^2 I)$. Now suppose the parameters $w^\star$ that provide the best generalization have large norm, $\|w^\star\|$, but there are several settings of the parameters that provide moderate fits to the data with smaller norms $b \ll \|w^\star\|$. Further suppose that parameters with norms $b < \|w\| < \|w^\star\|$ provide very poor fits to the data. The marginal likelihood will not favour a large value of σ that makes $w^\star$ likely under the prior — even though such a value could lead to a posterior with much better generalization, as in Figure 1(b). With more data, the likelihood signal for $w^\star$ will dominate, and the underfitting bias disappears.

4.3. Laplace Approximation in Deep Learning

Outside of a few special cases, such as Gaussian process regression, the marginal likelihood is intractable. Because the marginal likelihood is integrating with the respect to the prior, and we thus cannot effectively perform simple Monte Carlo, it is also much harder to approximate than the posterior predictive distribution. Moreover, modern neural networks contain millions of parameters, leaving few practical options. The *Laplace approximation* (LA) for model selection in Bayesian neural networks was originally proposed by MacKay (1992c), and has recently seen a resurgence of popularity (e.g. Immer et al., 2021). Moreover, several generalization metrics, such as the Aikake Information Criterion (AIC) and Bayesian Information Criterion (BIC), can be derived by further approximating the Laplace approximate marginal likelihood (Bishop, 2006a).

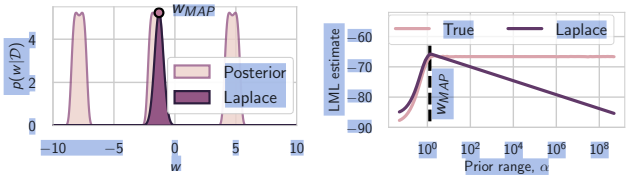


Figure 3. **Laplace approximation.** Model $x \sim \mathcal{N}(\sin(w), 1)$ with a uniform prior $w \sim U[-\alpha, \alpha]$. **(Left):** Posterior density and a Laplace approximation to the posterior (scaled for visualization); **(Right):** True marginal likelihood and the Laplace estimate as a function of α. As Laplace only captures a single mode, the Laplace estimate of marginal likelihood decreases linearly with α while the true marginal likelihood is roughly constant.

ent modes. We provide an example in Figure 3, with further details in Appendix A.

Laplace is also highly sensitive to the number of parameters in the model, as we show in Section 6. *We note the ELBO used in variational inference suffers from the same drawbacks*. We provide further details in Appendices A, B.

4.4. The Conditional Marginal Likelihood

Using the product decomposition of the marginal likelihood (see Section 3), we can write the LML as $\log p(\mathcal{D}|\mathcal{M}) = \sum_{i=1}^{n} \log p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M})$. Each term $\log p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M})$ is the predictive log-likelihood of the data point $\mathcal{D}_i$ under the Bayesian model average after observing the data $\mathcal{D}_{<i}$. The terms for i close to n are clearly indicative of generalization of the model to new test data: we train on the available data, and test on the remaining, unseen data. On the other hand, the terms corresponding to small i have an equally large effect on the marginal likelihood, but may have little to do with generalization.

Inspired by the reasoning above, we consider the *conditional log marginal likelihood (CLML)*: $\sum_{i=m}^{n} \log p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M}) = \log p(\mathcal{D}_{\geq m}|\mathcal{D}_{<m})$, where $m \in \{1, \ldots, n\}$ is the cut-off number, and $\mathcal{D}_{\geq m}$ is the set of datapoints $\mathcal{D}_m, \ldots, \mathcal{D}_n$. In CLML, we simply drop the first $m-1$ terms of the LML decomposition, to obtain a metric that is more aligned with generalization. In Appendix D, we provide further details on the CLML, including a permutation-invariant version, and study how performance varies with the choice of m in Figure 23 (Appendix K). We note the CLML can be written as $\int p(\mathcal{D}_{m:n}|w)p(w|\mathcal{D}_{1:m-1})dw$, and thus can be more easily estimated by Monte Carlo sampling than the LML, since samples from the posterior over $m-1$ points will typically have much greater likelihood than samples from the prior.

Variants of the CLML were considered in Berger and Pericchi (1996) as an *intrinsic Bayes factor* for handling improper uniform priors in hypothesis testing, and Fong and Holmes (2020) to show a connection with cross-validation and reduce the sensitivity to the prior. Our rationale and applications are different, motivated by understanding how the marginal likelihood can be fundamentally misaligned with generalization, with applications in neural architecture comparison, hyperparameter learning, and transfer learning. We expect the CLML to address the issues we have presented in this section, with the exception of overfitting, since CLML optimization is still fitting to withheld points. For hyperparameter optimization, we expect the CLML to be at its best relative to the LML for small datasets. As in Figure 1(b), the LML suffers because it has to assign mass to parameters that are unlikely to generate the data in order to reach parameters that are likely to generate the data. But as we get more data, the likelihood signal for the good parameters becomes overwhelming, and the marginal likelihood selects a reasonable value. Even for small datasets, the CLML is more free to select parameters that provide good generalization, since it is based on the posterior $p(w|\mathcal{D}_{<m})$ that is re-centred from the prior, as shown in Figure 1(b).

5. Training Speed and Learning Curves

How a model updates based on new information is a crucial factor determining its generalization properties. We will explore this behaviour with *learning curves* — graphs showing how $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$ changes as a function of n. The LML can be thought of as the area under the learning curve (Lyle et al., 2020). We will see that the first few terms in the learning curve corresponding to small n often decide which model is preferred by the LML. These terms are typically maximized by small, inflexible models, biasing the LML towards underfitting. We illustrate this behaviour in Figure 1(a): the marginal likelihood penalizes models with vague priors, even if after observing a few datapoints the posterior collapses, and generalizes well to the remaining datapoints.

Density Estimation. Consider the process where x is generated from a Gaussian distribution $\mathcal{N}(u, 1)$ and the mean parameter is in turn generated from a Gaussian distribution $u \sim \mathcal{N}(\mu, \sigma^2)$. Figure 4(a) shows the LML and the test predictive log likelihood as a function of the prior variance σ^2. The posterior over u and the predictive distribution are stable above a threshold of the prior variance σ^2, as the likelihood of the training data constrains the model and outweighs the increasingly weak prior. However, as we increase σ^2, the training data becomes increasingly unlikely according to the *prior*, so the marginal likelihood sharply decreases with σ^2. We provide analytical results in Appendix G.

A direct consequence of this behaviour is that two models may have the same generalization performance but very different values of the marginal likelihood; or worse, the marginal likelihood might favor a model with a poor generalization performance. We can see this effect in Figure 4(b),

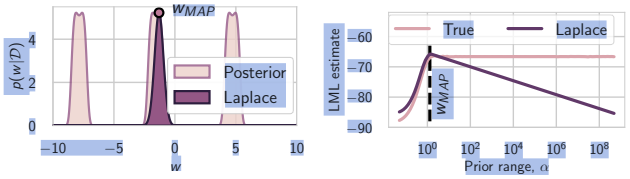


Figure 3. **Laplace approximation.** Model $x \sim \mathcal{N}(\sin(w), 1)$ with a uniform prior $w \sim U[-\alpha, \alpha]$. **(Left):** Posterior density and a Laplace approximation to the posterior (scaled for visualization); **(Right):** True marginal likelihood and the Laplace estimate as a function of α. As Laplace only captures a single mode, the Laplace estimate of marginal likelihood decreases linearly with α while the true marginal likelihood is roughly constant.

The Laplace approximation represents the parameter posterior with a Gaussian distribution centred at a local optima of the posterior, w_{MAP}, with covariance matrix given by the inverse Hessian at w_{MAP}.

Drawbacks of the Laplace approximation. The actual posterior distribution for a modern neural network is highly multimodal. By representing only a single mode, the Laplace approximation provides a poor representation of the true Occam factor, which is the posterior volume divided by the prior volume. As a consequence, the Laplace marginal likelihood will overly penalize diffuse priors that capture multiple reasonable parameter settings across different modes. We provide an example in Figure 3, with further details in Appendix A. Laplace is also highly sensitive to the number of parameters in the model, as we show in Section 6.

We note the ELBO used in variational inference suffers from the same drawbacks. We provide further details in Appendices A, B.

4.4. The Conditional Marginal Likelihood

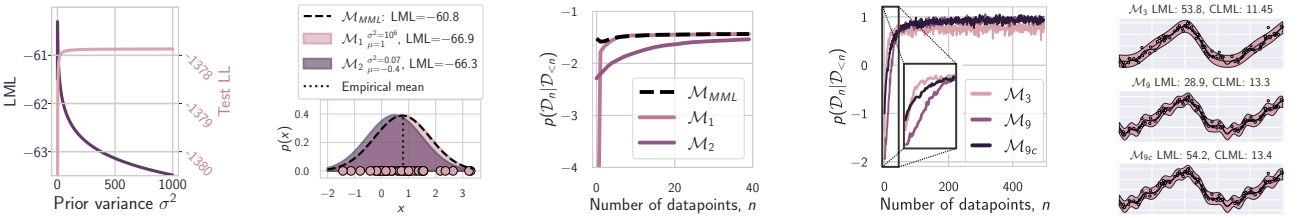
Using the product decomposition of the marginal likelihood (see Section 3), we can write the LML as $\log p(\mathcal{D}|\mathcal{M}) = \sum_{i=1}^{n} \log p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M})$. Each term $\log p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M})$ is the predictive log-likelihood of the data point $\mathcal{D}_i$ under the Bayesian model average after observing the data $\mathcal{D}_{<i}$. The terms for i close to n are clearly indicative of generalization of the model to new test data: we train on the available data, and test on the remaining, unseen data. On the other hand, the terms corresponding to small i have an equally large effect on the marginal likelihood, but may have little to do with generalization.

Inspired by the reasoning above, we consider the *conditional log marginal likelihood (CLML)*: $\sum_{i=m}^{n} \log p(\mathcal{D}_i|\mathcal{D}_{<i}, \mathcal{M}) = \log p(\mathcal{D}_{\geq m}|\mathcal{D}_{<m})$, where $m \in \{1, \ldots, n\}$ is the cut-off number, and $\mathcal{D}_{\geq m}$ is the set of datapoints $\mathcal{D}_m, \ldots, \mathcal{D}_n$. In CLML, we simply drop the first $m-1$ terms of the LML decomposition, to obtain a metric that is more aligned with generalization. In Appendix D, we provide further details on the CLML, including a permutation-invariant version, and study how performance varies with the choice of m in Figure 24 (Appendix K). We note the CML can be written as $\int p(\mathcal{D}_{m:n}|w)p(w|\mathcal{D}_{1:m-1})dw$, and thus can be more easily estimated by Monte Carlo sampling than the LML, since samples from the posterior over $m-1$ points will typically have much greater likelihood than samples from the prior.

Variants of the CLML were considered in Berger and Pericchi (1996) as an *intrinsic Bayes factor* for handling improper uniform priors in hypothesis testing, and Fong and Holmes (2020) to show a connection with cross-validation and reduce the sensitivity to the prior. Our rationale and applications are different, motivated by understanding how the marginal likelihood can be fundamentally misaligned with generalization. We do not consider prior sensitivity a deficiency of the marginal likelihood, since the marginal likelihood is evaluating the probability the data were generated from the prior. We also are the first to consider the CLML for neural architecture comparison, hyperparameter learning, approximate inference, and transfer learning. We expect the CLML to address the issues we have presented in this section, with the exception of overfitting, since CLML optimization is still fitting to withheld points. For hyperparameter optimization, we expect the CLML to be at its best relative to the LML for small datasets. As in Figure 1(b), the LML suffers because it has to assign mass to parameters that are unlikely to generate the data in order to reach parameters that are likely to generate the data. But as we get more data, the likelihood signal for the good parameters becomes overwhelming, and the marginal likelihood selects a reasonable value. Even for small datasets, the CLML is more free to select parameters that provide good generalization, since it is based on the posterior $p(w|\mathcal{D}_{<m})$ that is re-centred from the prior, as shown in Figure 1(b).

5. Training Speed and Learning Curves

How a model updates based on new information is a crucial factor determining its generalization properties. We will explore this behaviour with *learning curves* — graphs showing how $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$ changes as a function of n. The LML can be thought of as the area under the learning curve (Lyle et al., 2020). We will see that the first few terms in the learning curve corresponding to small n often decide which model is preferred by the LML. These terms are typically maximized by small, inflexible models, biasing the LML towards underfitting. We illustrate this behaviour in Figure 1(a): the marginal likelihood penalizes models with vague priors, even if after observing a few datapoints the posterior collapses, and generalizes well to the remaining datapoints.



(a) Fixed $\mu = 0$ (b) LML prefers $\mathcal{M}_1$ (c) Density learning curves (d) Fourier learning curves (e) Fourier model fits

Figure 4. **Training speed and learning curves. (a):** The prior variance continues to affect the marginal likelihood as its value increases whereas the test predictive distribution becomes insensitive to its value starting at a certain threshold. **(b):** While $\mathcal{M}_1$ and $\mathcal{M}_{MML}$ provide identical fits of the data, LML favors $\mathcal{M}_{MML}$. Moreover, LML prefers the model $\mathcal{M}_2$ with a poor data fit to $\mathcal{M}_1$. **(c)**: $\mathcal{M}_1$ trains faster than $\mathcal{M}_2$, but has a worse LML than $\mathcal{M}_2$. **(d)**: $\mathcal{M}_9$ provides a better fit after observing 60 datapoints, but the LML prefers $\mathcal{M}_3$ until $n = 297$. The model $\mathcal{M}_{9c}$ provides a near-identical fit to $\mathcal{M}_9$ after observing 50 datapoints, but is preferred by the LML. **(e)**: Data fit for $\mathcal{M}_3, \mathcal{M}_9$ and $\mathcal{M}_{9c}$. $\mathcal{M}_3$ undefits, while the other two models get identical fits.

where the predictive distributions of $\mathcal{M}_1$ and the maximum marginal likelihood (MML) model $\mathcal{M}_{MML}$ almost coincide, but the LML values are very different. Moreover, we can design a third model, $\mathcal{M}_2$, with a prior variance 0.07 and prior mean 2 which leads to a poor fit of the data but achieves higher marginal likelihood than $\mathcal{M}_1$. This simple example illustrates the general point presented in Section 4.1: LML measures the likelihood of the data according to the prior, which can be very different from the generalization performance of the corresponding posterior.

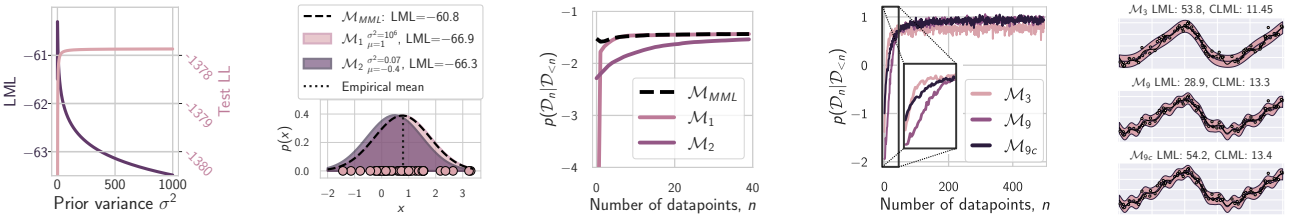
In Figure 4(c) we show $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$ as a function of n, averaged over 100 orderings of the data. We see that $\mathcal{M}_1$ trains faster than $\mathcal{M}_2$ — where the *training speed* is defined by Lyle et al. (2020) as "the number of data points required by a model to form an accurate posterior" — but achieves a lower LML, contradicting recent claims that "models which train faster will obtain a higher LML" (Lyle et al., 2020). These claims seem to implicitly rely on the assumption that all models start from the same $\log p(\mathcal{D}_1|\mathcal{M})$, which is not true in general as we demonstrate in Figure 4(c).

Fourier Model. Consider the Fourier model $f(x, a, b) = \sum_{d=1}^{D} a_d \sin(d \cdot x) + b_d \cos(d \cdot x)$, where $\{a_d, b_d\}_{d=1}^{D}$ are the parameters of the model, and D is the order of the model. To generate the data, we use a model of order $D = 9$. We sample the model parameters $\hat{a}_d, \hat{b}_d \sim \mathcal{N}(0, (1/d^2)^2)$. We sample 100 data points $x \sim \text{Uniform}[0, 1]$, and compute the corresponding $y = f(x, \hat{a}, \hat{b}) + \epsilon$, with noise $\epsilon \sim \mathcal{N}(0, 0.1^2)$. We then compare an order-9 model $\mathcal{M}_9$ and an order-3 $\mathcal{M}_3$ model on this dataset using LML and CLML. For both models, we use the prior $p(a_d) = p(b_d) = \mathcal{N}(0, 1)$. Note that the $\mathcal{M}_9$ model includes ground truth, while the $\mathcal{M}_3$ model does not. We show the fit for both models in Figure 4(e) (top and middle). $\mathcal{M}_9$ provides a much better fit of the true function, while $\mathcal{M}_3$ finds an overly simple solution. However, the LML strongly prefers the simpler $\mathcal{M}_3$ model, which achieves a value of 53.8 compared to 28.9 for the model $\mathcal{M}_9$. We additionally evaluate the CLML using 200 random orders and conditioning on $m = 85$ datapoints. CLML strongly prefers the flexible $\mathcal{M}_9$ model with a value of 28.9 compared to 11.45 for $\mathcal{M}_3$.

We can understand the behaviour of LML and CLML by examining the decomposition of LML into a sum over data and Figure 1(a). In Figure 4(d) we plot the terms $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$ of the decomposition as a function of n, averaged over 200 orderings of the data. For $n > 50$ observed datapoints, the more flexible model $\mathcal{M}_9$ achieves a better generalization log-likelihood $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$. However, for small n the simpler $\mathcal{M}_3$ model achieves better generalization, where the difference between $\mathcal{M}_3$ and $\mathcal{M}_9$ is more pronounced. As a result, LML prefers the simpler $\mathcal{M}_3$ for up to $n = 297$ datapoints! For $n \in [50, 296]$ the LML picks the model with suboptimal generalization performance. We can achieve the best of both worlds with the corrected model $\mathcal{M}_{9c}$ with the parameter prior $a_d, b_d \sim \mathcal{N}(0, (1/d^2)^2)$: strong generalization performance both for small and large training dataset sizes n. For further details, please see Appendix E.

Neural Networks. We show the rank of 6 different neural network architectures on their BMA test accuracy on CIFAR-10 for different dataset sizes in Figure 10(b) (Appendix). We see that DenseNet121 and GoogLeNet train faster than ResNet-18 and VGG19, but rank worse with more data. In Figure 10(a) (Appendix), we show the correlation of the BMA test log-likelihood with the LML is positive for small datasets and negative for larger datasets, whereas the correlation with the CLML is consistently positive. Finally, Figure 10(a) (Appendix) shows that the Laplace LML heavily penalizes the number of parameters, as in Section 4.3. We provide additional details in Appendix F.

Summary. In contrast with Lyle et al. (2020), we find that models that train faster do not necessarily have higher marginal likelihood, or better generalization. Indeed, the opposite can be true: fast training is associated with rapid



(a) Fixed $\mu = 0$ (b) LML prefers $\mathcal{M}_2$ (c) Density learning curves (d) Fourier learning curves (e) Fourier model fits

Figure 4. **Training speed and learning curves. (a):** The prior variance continues to affect the marginal likelihood as its value increases whereas the test predictive distribution becomes insensitive to its value starting at a certain threshold. **(b):** While $\mathcal{M}_1$ and $\mathcal{M}_{MML}$ provide identical fits of the data, LML favors $\mathcal{M}_{MML}$. Moreover, LML prefers the model $\mathcal{M}_2$ with a poor data fit to $\mathcal{M}_1$. **(c)**: $\mathcal{M}_1$ trains faster than $\mathcal{M}_2$, but has a worse LML than $\mathcal{M}_2$. **(d)**: $\mathcal{M}_9$ provides a better fit after observing 60 datapoints, but the LML prefers $\mathcal{M}_3$ until $n = 297$. The model $\mathcal{M}_{9c}$ provides a near-identical fit to $\mathcal{M}_9$ after observing 50 datapoints, but is preferred by the LML. **(e)**: Data fit for $\mathcal{M}_3$, $\mathcal{M}_9$ and $\mathcal{M}_{9c}$. $\mathcal{M}_3$ undefits, while the other two models get identical fits.

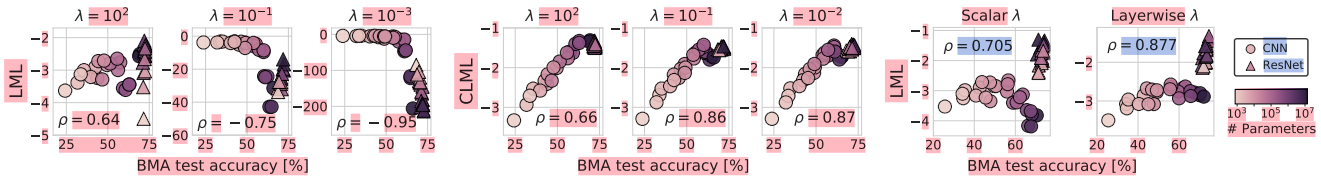
Density Estimation. Consider the process where x is generated from a Gaussian distribution $\mathcal{N}(u, 1)$ and the mean parameter is in turn generated from a Gaussian distribution $u \sim \mathcal{N}(\mu, \sigma^2)$. Figure 4(a) shows the LML and the test predictive log likelihood as a function of the prior variance σ^2. The posterior over u and the predictive distribution are stable above a threshold of the prior variance σ^2, as the likelihood of the training data constrains the model and outweighs the increasingly weak prior. However, as we increase σ^2, the training data becomes increasingly unlikely according to the *prior*, so the marginal likelihood sharply decreases with σ^2. We provide analytical results in Appendix G.

A direct consequence of this behaviour is that two models may have the same generalization performance but very different values of the marginal likelihood; or worse, the marginal likelihood might favor a model with a poor generalization performance. We can see this effect in Figure 4(b), where the predictive distributions of $\mathcal{M}_1$ and the maximum marginal likelihood (MML) model $\mathcal{M}_{MML}$ almost coincide, but the LML values are very different. Moreover, we can design a third model, $\mathcal{M}_2$, with a prior variance 0.07 and prior mean 2 which leads to a poor fit of the data but achieves higher marginal likelihood than $\mathcal{M}_1$. This simple example illustrates the general point presented in Section 4.1: LML measures the likelihood of the data according to the prior, which can be very different from the generalization performance of the corresponding posterior.

In Figure 4(c) we show $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$ as a function of n, averaged over 100 orderings of the data. We see that $\mathcal{M}_1$ trains faster than $\mathcal{M}_2$ — where the *training speed* is defined by Lyle et al. (2020) as "the number of data points required by a model to form an accurate posterior" — but achieves a lower LML, contradicting recent claims that "models which train faster will obtain a higher LML" (Lyle et al., 2020). These claims seem to implicitly rely on the assumption that all models start from the same $\log p(\mathcal{D}_1|\mathcal{M})$, which is not true in general as we demonstrate in Figure 4(c).

Fourier Model. Consider the Fourier model $f(x, a, b) = \sum_{d=1}^{D} a_d \sin(d \cdot x) + b_d \cos(d \cdot x)$, where $\{a_d, b_d\}_{d=1}^{D}$ are the parameters of the model, and D is the order of the model. To generate the data, we use a model of order $D = 9$. We sample the model parameters $\hat{a}_d, \hat{b}_d \sim \mathcal{N}(0, (1/d^2)^2)$. We sample 100 data points $x \sim \text{Uniform}[0, 1]$, and compute the corresponding $y = f(x, \hat{a}, \hat{b}) + \epsilon$, with noise $\epsilon \sim \mathcal{N}(0, 0.1^2)$. We then compare an order-9 model $\mathcal{M}_9$ and an order-3 $\mathcal{M}_3$ model on this dataset using LML and CLML. For both models, we use the prior $p(a_d) = p(b_d) = \mathcal{N}(0, 1)$. Note that the $\mathcal{M}_9$ model includes ground truth, while the $\mathcal{M}_3$ model does not. We show the fit for both models in Figure 4(e) (top and middle). $\mathcal{M}_9$ provides a much better fit of the true function, while $\mathcal{M}_3$ finds an overly simple solution. However, the LML strongly prefers the simpler $\mathcal{M}_3$ model, which achieves a value of 53.8 compared to 28.9 for the model $\mathcal{M}_9$. We additionally evaluate the CLML using 200 random orders and conditioning on $m = 85$ datapoints. CLML strongly prefers the flexible $\mathcal{M}_9$ model with a value of 28.9 compared to 11.45 for $\mathcal{M}_3$.

We can understand the behaviour of LML and CLML by examining the decomposition of LML into a sum over data and Figure 1(a). In Figure 4(d) we plot the terms $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$ of the decomposition as a function of n, averaged over 200 orderings of the data. For $n > 50$ observed datapoints, the more flexible model $\mathcal{M}_9$ achieves a better generalization log-likelihood $\log p(\mathcal{D}_n|\mathcal{D}_{<n})$. However, for small n the simpler $\mathcal{M}_3$ model achieves better generalization, where the difference between $\mathcal{M}_3$ and $\mathcal{M}_9$ is more pronounced. As a result, LML prefers the simpler $\mathcal{M}_3$ for up to $n = 297$ datapoints! For $n \in [50, 296]$ the LML picks the model with suboptimal generalization performance. We can achieve the best of both worlds with the corrected model $\mathcal{M}_{9c}$ with the parameter prior $a_d, b_d \sim \mathcal{N}(0, (1/d^2)^2)$: strong generalization performance both for small and large training dataset



(a) LML vs BMA accuracy (b) CLML vs BMA accuracy (c) Optimized λ

Figure 5. **Neural hyperparameter optimization for CIFAR-100.** The correlation (Spearman's ρ) between the model rankings and generalization. For panels **(a), (b)**, we consider a fixed prior precision $\lambda = 10^2$, 10^{-1}, and 10^{-3}. **(a)** Correlation between the Laplace BMA test accuracy and the LML. **(b)**: Correlation between the BMA test accuracy and the CLML. **(c)**: Correlation between the BMA test accuracy and the LML for optimized global (left) and layer-wise (right) prior precision. The correlation between the LML and test accuracy highly depends on the value of the prior precision. The CLML does not suffer from this sensitivity to the prior precision.

posterior contraction, which can incur a significant Occam factor penalty (Section 3), because the first few terms in the LML expansion are very negative. We also show that, unlike the LML, the CLML is positively correlated with generalization in both small and large n regimes, and that it is possible for a single model to do well in both regimes.

6. Model Selection and Architecture Search

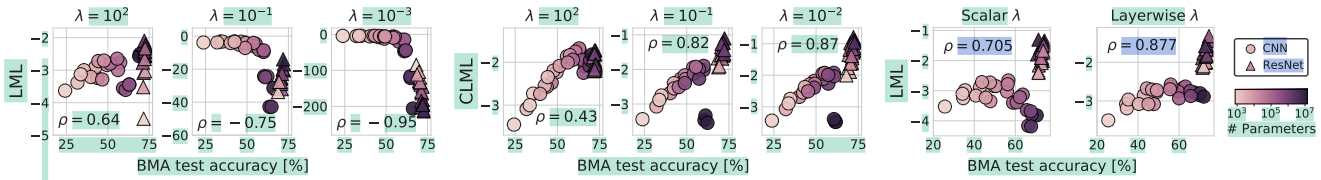
In Section 4.1, we discussed how the marginal likelihood can be misaligned with generalization because the two notions answer fundamentally different questions. In the model selection and architecture search, we aim to find the model with the best predictive distribution, not the prior most likely to generate the training data. Here, we consider neural architecture selection, following recent works on using the Laplace marginal likelihood for ranking neural architectures (Immer et al., 2021; Daxberger et al., 2021). We investigate the correlation between LML and generalization performance across 25 convolutional (CNN) and residual (ResNet) architectures of varying depth and width on CIFAR-10 and CIFAR-100, following the setup of Immer et al. (2021). See Appendix H for more details.

First, we investigate the correlation between the Laplace marginal likelihood and BMA test accuracy, when the prior precision (aka weight decay) λ is fixed. Figure 5(a) shows the results for fixed prior precision $\lambda = 10^2$, 10^{-1}, and 10^{-3}. In each panel, we additionally report the Spearman's correlation coefficient ρ (Kendall, 1948) between the model rankings according to the BMA test accuracy and the LML. LML is positively correlated with the BMA test accuracy when the prior precision is high, $\lambda = 10^2$, but the correlation becomes increasingly negative as λ decreases. While the prior precision has little effect on the BMA test accuracy, it has a significant effect on the approximation of the LML values and model ranking! As discussed in Section 4.1, the marginal likelihood heavily penalizes vague priors, especially in large, flexible models. Moreover, as discussed in Section 4.3, Laplace is especially sensitive to the prior variance, and the number of parameters in the model.

By the same rationale, we expect the conditional marginal likelihood to help alleviate this problem, since it evaluates the likelihood of the data under the posterior, rather than the prior. Moreover, CLML is evaluated in *function space* rather than in parameter space (see Appendix D for details), and consequently is not sensitive to the number of parameters in the model. Indeed, in Figure 5(b) the CLML exhibits a positive correlation with the generalization performance for both large and small values of the prior precision. In Appendix H we show that unlike the LML, the CLML is positively correlated with BMA accuracy, BMA log-likelihood, MAP accuracy and MAP log-likelihood across a wide range of prior precision values both on CIFAR-10 and CIFAR-100.

Prior precision optimization. In Figure 5(c), we show that optimizing the global or layer-wise prior precision leads to a positive correlation between the LML and the BMA test accuracy, following the online procedure in Immer et al. (2021). This optimization selects high-precision priors, leading to a positive correlation between the LML estimate and the test performance. Notably, optimizing a separate prior scale for each layer leads to higher correlation, an observation that was also made in MacKay (1992c).

Summary. Claims that "the marginal likelihood can be used to choose between two discrete model alternatives after training" and that "we only need to choose the model with a higher LML value" (Immer et al., 2021) do not hold universally: we see in Figure 5(a) that the marginal likelihood can be negatively correlated with generalization in practice! In Figure 5(c) we have seen that this correlation can be fixed by optimizing the prior precision, but in general there is no recipe for how many prior hyperparameters we should be optimizing to ensure a positive correlation. For example, in Figure 5(c) optimizing the global prior precision leads to a positive correlation for ResNet models but not for CNNs. The CLML on the other hand consistently provides a positive correlation with the generalization performance.



(a) LML vs BMA accuracy (b) CLML vs BMA accuracy (c) Optimized λ

Figure 5. **Neural hyperparameter optimization for CIFAR-100.** The correlation (Spearman's ρ) between the model rankings and generalization. For panels **(a), (b)**, we consider a fixed prior precision $\lambda = 10^2$, 10^{-1}, and 10^{-3}. **(a)** Correlation between the Laplace BMA test accuracy and the LML. **(b)**: Correlation between the BMA test accuracy and the CLML. **(c)**: Correlation between the BMA test accuracy and the LML for optimized global (left) and layer-wise (right) prior precision. The correlation between the LML and test accuracy highly depends on the value of the prior precision. The CLML does not suffer from this sensitivity to the prior precision.

sizes n. These results are qualitatively what we expect: for small datasizes, the prior, and thus the LML, are relatively predictive of generalization. For intermediate size data, the first terms in the LML decomposition have a negative effect on how well LML predicts generalization. For asymptotically large data sizes, the first terms have a diminishing effect, and the LML becomes a consistent estimator for the true model if it is contained within its support. For further details, please see Appendix E.

Neural Networks. We show the rank of 6 different neural network architectures on their BMA test accuracy on CIFAR-10 for different dataset sizes in Figure 10(b) (Appendix). We see that DenseNet121 and GoogLeNet train faster than ResNet-18 and VGG19, but rank worse with more data. In Figure 10(a) (Appendix), we show the correlation of the BMA test log-likelihood with the LML is positive for small datasets and negative for larger datasets, whereas the correlation with the CLML is consistently positive. Finally, Figure 10(a) (Appendix) shows that the Laplace LML heavily penalizes the number of parameters, as in Section 4.3. We provide additional details in Appendix F.

Summary. In contrast with Lyle et al. (2020), we find that models that train faster do not necessarily have higher marginal likelihood, or better generalization. Indeed, the opposite can be true: fast training is associated with rapid posterior contraction, which can incur a significant Occam factor penalty (Section 3), because the first few terms in the LML expansion are very negative. We also show that, unlike the LML, the CLML is positively correlated with generalization in both small and large n regimes, and that it is possible for a single model to do well in both regimes.

6. Model Selection and Architecture Search

In Section 4.1, we discussed how the marginal likelihood can be misaligned with generalization because the two notions answer fundamentally different questions. In the model selection and architecture search, we aim to find the model with the best predictive distribution, not the prior most likely to generate the training data. Here, we consider neural architecture selection, following recent works on using the Laplace marginal likelihood for ranking neural architectures (Immer et al., 2021; Daxberger et al., 2021). We investigate the correlation between LML and generalization performance across 25 convolutional (CNN) and residual (ResNet) architectures of varying depth and width on CIFAR-10 and CIFAR-100, following the setup of Immer et al. (2021). See Appendix H for more details.

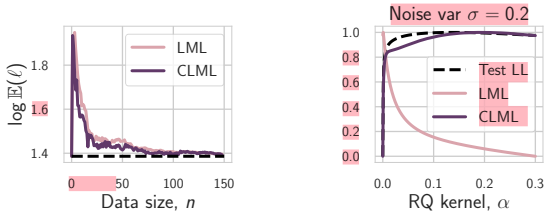
First, we investigate the correlation between the Laplace marginal likelihood and BMA test accuracy, when the prior precision (aka weight decay) λ is fixed. Figure 5(a) shows the results for fixed prior precision $\lambda = 10^2$, 10^{-1}, and 10^{-3}. In each panel, we additionally report the Spearman's correlation coefficient ρ (Kendall, 1948) between the model rankings according to the BMA test accuracy and the LML. LML is positively correlated with the BMA test accuracy when the prior precision is high, $\lambda = 10^2$, but the correlation becomes increasingly negative as λ decreases. While the prior precision has little effect on the BMA test accuracy, it has a significant effect on the approximation of the LML values and model ranking! As discussed in Section 4.1, the marginal likelihood heavily penalizes vague priors, especially in large, flexible models. Moreover, as discussed in Section 4.3, Laplace is especially sensitive to the prior variance, and the number of parameters in the model.

By the same rationale, we expect the conditional marginal likelihood to help alleviate this problem, since it evaluates the likelihood of the data under the posterior, rather than the prior. Moreover, CLML is evaluated in *function space* rather than in parameter space (see Appendix D for details), and consequently is not sensitive to the number of parameters in the model. Indeed, in Figure 5(b) the CLML exhibits a positive correlation with the generalization performance for both large and small values of the prior precision. In Appendix H we show that unlike the LML and the negative validation loss, the CLML is positively correlated with BMA accuracy, BMA log-likelihood, MAP accuracy and MAP log-likelihood across a wide range of prior precision values

7. Hyperparameter Learning

We want to select hyperparameters that provide the best possible generalization. We have argued that LML optimization is not always aligned with generalization. As in Section 4.2, there are two ways LML optimization can go awry. The first is associated with overfitting through ignoring uncertainty. The second is associated with underfitting as a consequence of needing to support many unreasonable functions. CLML optimization can help address this second issue, but not the first, since it still ignores uncertainty in the hyperparameters.

We provide examples of both issues in GP kernel hyperparameter learning. Curiously, overfitting the marginal likelihood through ignoring uncertainty can lead to *underfitting* in function space, which is not a feature of standard maximum likelihood overfitting. We then demonstrate that the CLML provides a highly practical mechanism for deep kernel hyperparameter learning, significantly improving performance over LML optimization. The performance gains can be explained as a consequence of the second issue, where we accordingly see the biggest performance gains on smaller datasets, as we predict in the discussion in Section 4.2.



(a) Underfitting bias (b) Underfitting with the RQ kernel

Figure 6. **LML for hyperparameter learning in Gaussian processes. (a):** The log-lengthscale learned by LML and CLML in a GP regression model averaged over 100 datasets generated from a GP model with a lengthscale of 4. Unlike the train likelihood, LML has a bias towards underfitting, consistently overestimating the lengthscale, particularly for small $n < 20$. **(b):** Test log-likelihood, LML and CLML as a function of the α hyper-parameter in the rational quadratic kernel with noise variance $\sigma^2 = 0.2$. The CLML is closely aligned with test log likelihood, unlike the LML.

7.1. Two issues with LML Optimization

Using Gaussian process (GP) kernel learning, we provide illustrative examples of two conceptually different ways LML optimization can select hyperparameters that provide poor generalization, discussed in Section 4.2.

If we severely overfit the GP LML by optimizing with respect to the covariance matrix itself, subject to no constraints, the solution is the empirical covariance of the data, which is degenerate and biased. Figure 6(a) shows RBF kernel learning inherits this bias by over-estimating the length-scale parameter, which pushes the eigenvalues of the covariance matrix closer to the degenerate unconstrained solution. As we observe more data, the RBF kernel becomes increasingly constrained, and the bias disappears (Wilson et al., 2015). This finding is curious in that it shows how ignoring uncertainty in LML can lead to *underfitting* in data space, since a larger length-scale will lead to a worse fit of the data. This behaviour is not a feature of standard maximum likelihood overfitting, and is also not a property of the LML overfitting in the example of Figure 9 (Appendix). But since it is overfitting arising from a lack of uncertainty representation, the CLML suffers from the same issue.

In our next experiment, we generate data from a GP with a rational quadratic (RQ) kernel. Figure 6(b) shows that if we overestimate the observation noise, then the LML is completely misaligned with the shape of the test log-likelihood as a function of the α hyper-parameter of the RQ kernel, whereas the CLML is still strongly correlated with the test likelihood. We see here the underfitting bias of Figure 1(b), where supporting an α of any reasonable size leads to a prior over functions unlikely to generate the training data. In Appendix I, we show that under the ground truth observation noise both LML and CLML provide adequate representations of the test log-likelihood in this instance. Indeed, the CLML is additionally *more robust to misspecification* than the LML.

We provide further details in Appendix I.

7.2. Deep Kernel Learning

Deep kernel learning (DKL) presents a scenario in which a large number of hyperparameters are tuned through marginal likelihood optimization. While it has been noted that DKL can overfit through ignoring hyperparameter uncertainty (Ober et al., 2021), in this section we are primarily concerned with the underfitting described in Section 4.2, where the CLML will lead to improvements.

Here we showcase CLML optimization as a practical tool in both UCI regression tasks and transfer learning tasks from Patacchiola et al. (2020). In UCI regression tasks, we examine the performance of LML vs CLML in terms of test performance when training with limited amounts of training data. In Figure 7 we see a common trend: when we are restricted to a small number of training examples, LML optimization is outperformed by CLML optimization. As the number of training examples increases, the gap between LML and CLML optimized models closes. We provide further details, with complete results including a comparison of negative log-likelihoods in Appendix J.

In transfer learning tasks we are typically concerned with how well our method performs on unseen data, which may be from a different distribution than the training data, rather than how well aligned our prior is to the training data. In Fig-

both on CIFAR-10 and CIFAR-100.

Prior precision optimization. In Figure 5(c), we show that optimizing the global or layer-wise prior precision leads to a positive correlation between the LML and the BMA test accuracy, following the online procedure in Immer et al. (2021). This optimization selects high-precision priors, leading to a positive correlation between the LML estimate and the test performance. Notably, optimizing a separate prior scale for each layer leads to higher correlation, an observation that was also made in MacKay (1992c).

Summary. Claims that "the marginal likelihood can be used to choose between two discrete model alternatives after training" and that "we only need to choose the model with a higher LML value" (Immer et al., 2021) do not hold universally: we see in Figure 5(a) that the marginal likelihood can be negatively correlated with generalization in practice! In Figure 5(c) we have seen that this correlation can be fixed by optimizing the prior precision, but in general there is no recipe for how many prior hyperparameters we should be optimizing to ensure a positive correlation. For example, in Figure 5(c) optimizing the global prior precision leads to a positive correlation for ResNet models but not for CNNs. The CLML on the other hand consistently provides a positive correlation with the generalization performance.

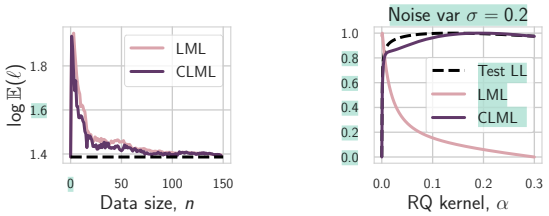
7. Hyperparameter Learning

We want to select hyperparameters that provide the best possible generalization. We have argued that LML optimization is not always aligned with generalization. As in Section 4.2, there are two ways LML optimization can go awry. The first is associated with overfitting through ignoring uncertainty. The second is associated with underfitting as a consequence of needing to support many unreasonable functions. CLML optimization can help address this second issue, but not the first, since it still ignores uncertainty in the hyperparameters.

We provide examples of both issues in GP kernel hyperparameter learning. Curiously, overfitting the marginal likelihood through ignoring uncertainty can lead to *underfitting* in function space, which is not a feature of standard maximum likelihood overfitting. We then demonstrate that the CLML provides a highly practical mechanism for deep kernel hyperparameter learning, significantly improving performance over LML optimization. The performance gains can be explained as a consequence of the second issue, where we accordingly see the biggest performance gains on smaller datasets, as we predict in the discussion in Section 4.2.

7.1. Two issues with LML Optimization

Using Gaussian process (GP) kernel learning, we provide illustrative examples of two conceptually different ways LML optimization can select hyperparameters that provide poor generalization, discussed in Section 4.2.



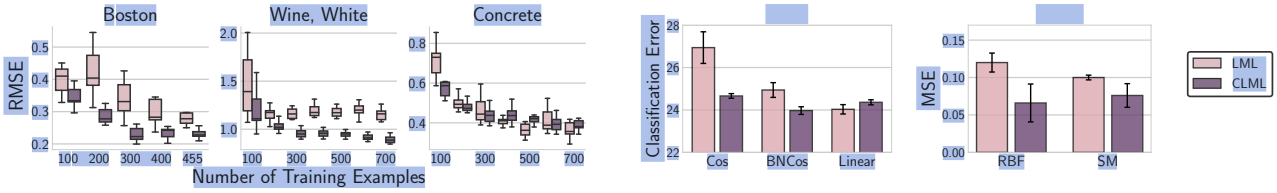
(a) Underfitting bias (b) Underfitting with the RQ kernel

Figure 6. **LML for hyperparameter learning in Gaussian processes. (a):** The log-lengthscale learned by LML and CLML in a GP regression model averaged over 100 datasets generated from a GP model with a lengthscale of 4. Unlike the train likelihood, LML has a bias towards underfitting, consistently overestimating the lengthscale, particularly for small $n < 20$. **(b):** Test log-likelihood, LML and CLML as a function of the α hyper-parameter in the rational quadratic kernel with noise variance $\sigma^2 = 0.2$. The CLML is closely aligned with test log likelihood, unlike the LML.

If we severely overfit the GP LML by optimizing with respect to the covariance matrix itself, subject to no constraints, the solution is the empirical covariance of the data, which is degenerate and biased. Figure 6(a) shows RBF kernel learning inherits this bias by over-estimating the length-scale parameter, which pushes the eigenvalues of the covariance matrix closer to the degenerate unconstrained solution. As we observe more data, the RBF kernel becomes increasingly constrained, and the bias disappears (Wilson et al., 2015). This finding is curious in that it shows how ignoring uncertainty in LML can lead to *underfitting* in data space, since a larger length-scale will lead to a worse fit of the data. This behaviour is not a feature of standard maximum likelihood overfitting, and is also not a property of the LML overfitting in the example of Figure 9 (Appendix). But since it is overfitting arising from a lack of uncertainty representation, the CLML suffers from the same issue.

In our next experiment, we generate data from a GP with a rational quadratic (RQ) kernel. Figure 6(b) shows that if we overestimate the observation noise, then the LML is completely misaligned with the shape of the test log-likelihood as a function of the α hyper-parameter of the RQ kernel, whereas the CLML is still strongly correlated with the test likelihood. We see here the underfitting bias of Figure 1(b), where supporting an α of any reasonable size leads to a prior over functions unlikely to generate the training data. In Appendix I, we show that under the ground truth observation noise both LML and CLML provide adequate representations of the test log-likelihood in this instance. Indeed, the CLML is additionally *more robust to misspecification* than the LML.

We provide further details in Appendix I.



(a) Deep Kernel Learning Regression (b) Transfer to Omniglot (c) Transfer to QMUL

Figure 7. **Deep kernel learning (DKL). (a):** DKL trained with LML and CLML optimization with the cutoff m at 90% of the training data. CLML optimization outperforms LML optimization in low data regimes. **(b):** Classification error ($\pm$ stderr) for Deep Kernel Transfer (DKT) models trained on the Omniglot dataset and evaluated on EMNIST. For both Cosine similarity and BatchNorm Cosine similarity kernels, CLML optimization outperforms LML optimization. **(c):** MSE for DKT ($\pm$ stderr) applied to the QMUL dataset shift problem of recognizing head poses. For both RBF and SM kernels, CLML optimization outperforms LML optimization. In all panels results are collected over 10 random initializations. Test likelihood results (Appendix) are qualitatively similar.

7.2. Deep Kernel Learning

Deep kernel learning (DKL) (Wilson et al., 2016b) presents a scenario in which a large number of hyperparameters are tuned through marginal likelihood optimization. While it has been noted that DKL can overfit through ignoring hyperparameter uncertainty (Ober et al., 2021), in this section we are primarily concerned with the underfitting described in Section 4.2, where the CLML will lead to improvements.

Here we showcase CLML optimization as a practical tool in both UCI regression tasks and transfer learning tasks from Patacchiola et al. (2020). In UCI regression tasks, we examine the performance of LML vs CLML in terms of test performance when training with limited amounts of training data. In Figure 7 we see a common trend: when we are restricted to a small number of training examples, LML optimization is outperformed by CLML optimization. As the number of training examples increases, the gap between LML and CLML optimized models closes. We provide further details, with complete results including a comparison of negative log-likelihoods in Appendix J.

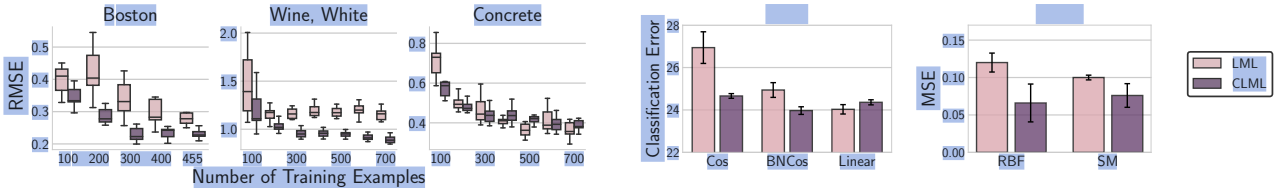
In transfer learning tasks we are typically concerned with how well our method performs on unseen data, which may be from a different distribution than the training data, rather than how well aligned our prior is to the training data. In Figure 7 we reproduce the Deep Kernel Transfer (DKT) transfer learning experiments from Patacchiola et al. (2020), replacing LML optimization with CLML optimization. In these experiments DKL models are trained on one task with either LML or CLML optimization, and then evaluated on a separate but related task. Figure 4(a), and Table 4 (Appendix), shows a comparison of methods on a transfer learning task in which we train on the Omniglot dataset and test on the EMNIST dataset. In both experiments CLML optimization provides a clear improvement over LML optimization. Figure 7(b), and Table 3 (Appendix), shows a comparison of methods on the QMUL head pose regression problem for estimating the angular pose of gray-scale images of faces, where the individuals in the test set are distinct from those in the training set leading to dataset shift. For experimental details see Patacchiola et al. (2020).

8. Discussion

While the marginal likelihood provides a powerful mechanism for hypothesis testing, and can be practical for hyperparameter tuning, we show that it is in many ways misaligned with generalization. These results are particularly timely in light of recent work proposing the marginal likelihood for model selection and hyperparameter tuning in deep learning. We show that a conditional marginal likelihood retains the convenient properties of the marginal likelihood, but helps resolve the misalignment between the marginal likelihood and generalization. We find that the conditional marginal likelihood provides particularly compelling performance in learning deep kernel hyperparameters, especially on smaller datasets, and transfer learning problems.

To what extent does approximate inference affect our results? Almost all of our experiments use the exact LML and CLML: the density model, Fourier features, Gaussian processes, and deep kernel learning. While the neural architecture in Section 6 necessarily use approximate inference, the results are qualitatively similar to the exact experiments, and these results are also what we expect from Section 4. A key advantage of working with the CLML is that it can be effectively approximated by sampling. However, what we observe about the LML behaviour stands on its own, independently of the CLML.

How does the correlation between LML and generalization vary with dataset size n? The relationship between the LML and generalization is non-monotonic with dataset size. For very small datasets, the first (and only) terms in the LML decomposition are typically predictive of generalization. For intermediate datasets, these terms have a negative



(a) Deep Kernel Learning Regression (b) Transfer to Omniglot (c) Transfer to QMUL

Figure 7. **Deep kernel learning (DKL). (a):** DKL trained with LML and CLML optimization with the cutoff m at 90% of the training data. CLML optimization outperforms LML optimization in low data regimes. **(b):** Classification error ($\pm$ stderr) for Deep Kernel Transfer (DKT) models trained on the Omniglot dataset and evaluated on EMNIST. For both Cosine similarity and BatchNorm Cosine similarity kernels, CLML optimization outperforms LML optimization. **(c):** MSE for DKT ($\pm$ stderr) applied to the QMUL dataset shift problem of recognizing head poses. For both RBF and SM kernels, CLML optimization outperforms LML optimization. In all panels results are collected over 10 random initializations. Test likelihood results (Appendix) are qualitatively similar.

ure 7 we reproduce the Deep Kernel Transfer (DKT) transfer learning experiments from Patacchiola et al. (2020), replacing LML optimization with CLML optimization. In these experiments DKL models are trained on one task with either LML or CLML optimization, and then evaluated on a separate but related task. Figure 4(a), and Table 4 (Appendix), shows a comparison of methods on a transfer learning task in which we train on the Omniglot dataset and test on the EMNIST dataset. In both experiments CLML optimization provides a clear improvement over LML optimization. Figure 7(b), and Table 3 (Appendix), shows a comparison of methods on the QMUL head pose regression problem for estimating the angular pose of gray-scale images of faces, where the individuals in the test set are distinct from those in the training set leading to dataset shift. For experimental details see Patacchiola et al. (2020).

8. Conclusion

While the marginal likelihood provides a powerful mechanism for hypothesis testing, and can be practical for hyperparameter tuning, we show that it is in many ways misaligned with generalization. These results are particularly timely in light of recent work proposing the marginal likelihood for model selection and hyperparameter tuning in deep learning. We show that a conditional marginal likelihood retains the convenient properties of the marginal likelihood, but helps resolve the misalignment between the marginal likelihood and generalization. We find that the conditional marginal likelihood provides particularly compelling performance in learning deep kernel hyperparameters, especially on smaller datasets, and transfer learning problems.

Acknowledgements. We thank Sam Stanton, Marc Finzi, and Tom Minka for helpful discussions. This research is supported by an Amazon Research Award, Facebook Research, Google Research, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, and NSF 1922658 NRT-HDR.

References

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. (2018). Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168. PMLR.

Alquier, P. (2021). User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216*.

Berger, J. and Jeffreys, W. (1991). Minimal bayesian testing of precise hypotheses, model selection, and ockham's razor. Technical report, Technical Report, Purdue University.

Berger, J. O. and Pericchi, L. R. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122.

Bishop, C. M. (2006a). *Pattern Recognition and Machine Learning*. Springer.

Bishop, C. M. (2006b). *Pattern recognition and machine learning*. springer.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34.

Domingos, P. (1999). The role of occam's razor in knowledge discovery. *Data mining and knowledge discovery*, 3(4):409–425.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fong, E. and Holmes, C. C. (2020). On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496.

effect on the correlation with generalization, as the posterior can differ significantly from the prior. Finally, for asymptotically large datasets, the first terms have a diminishing effect, and the LML becomes a consistent estimator for the correct model, assuming it is within the set of considered models. We observe these results empirically in Figure 4(d), where LML picks the better generalizing model for $n < 50$ and $n > 296$. For n in $[50, 296]$ it picks the wrong model.

Can we construct a model which performs well for both small and large n? While we are primarily concerned with model *selection*, model *construction* is intimately related. There is a conventional wisdom that one should use small models for small datasets, and large models for large datasets. We show in Figure 4(e) that this prescription is incorrect: we can achieve the best of both worlds, a model which is good in both small and large n regimes, by combining flexibility with reasonable inductive biases, aligned with the discussion in Wilson and Izmailov (2020).

Is the CLML "just" cross-validation? The LML itself is arguably a form of cross-validation, although it is not standard cross-validation (Fong and Holmes, 2020). The CLML can be significantly more efficient and practical than standard cross-validation for gradient-based learning of many hyperparameters. However, our goal with the CLML was not to explore a measure that is starkly different from cross-validation, nor do we consider any arguable similarity with cross-validation a deficiency. Instead, we show how a minor modification to the LML can improve alignment with generalization, and be practical for hyperparameter learning. We also show in Appendix H that the CLML correlates better than negative of the validation with the BMA test accuracy.

What are the most important practical take-aways? The DKL hyperparameter learning with the CLML is of particular practical significance. These experiments, in Section 7.2, show that the CLML can be a promising drop-in replacement for the LML for learning many hyperparameters, especially for transfer learning and small datasets, and are our most substantial experiments involving DNNs. Future work in this area could have a substantial impact on the way we estimate hyperparameters in probabilistic models. Another practical take-away is that in general the standard LML cannot reliably select models with strong generalization.

Acknowledgements. We thank Sam Stanton, Marc Finzi, Rich Turner, and Tom Minka for helpful discussions. We also thank Andreas Kirsch for thoughtful questions and catching a bug in Figure 5b, which has been corrected. This research is supported by an Amazon Research Award, Facebook Research, Google Research, NSF I-DISRE 193471, NIH R01DA048764-01A1, NSF IIS-1910266, and NSF 1922658 NRT-HDR.

References

Alemi, A., Poole, B., Fischer, I., Dillon, J., Saurous, R. A., and Murphy, K. (2018). Fixing a broken elbo. In *International Conference on Machine Learning*, pages 159–168. PMLR.

Alquier, P. (2021). User-friendly introduction to PAC-Bayes bounds. *arXiv preprint arXiv:2110.11216*.

Berger, J. and Jeffreys, W. (1991). Minimal bayesian testing of precise hypotheses, model selection, and ockham's razor. Technical report, Technical Report, Purdue University.

Berger, J. O. and Pericchi, L. R. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122.

Bishop, C. M. (2006a). *Pattern Recognition and Machine Learning*. Springer.

Bishop, C. M. (2006b). *Pattern recognition and machine learning*. springer.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34.

Domingos, P. (1999). The role of occam's razor in knowledge discovery. *Data mining and knowledge discovery*, 3(4):409–425.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fong, E. and Holmes, C. C. (2020). On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496.

Gelman, A. (2011). David MacKay and Occam's Razor. https://statmodeling.stat.columbia.edu/2011/12/04/david-mackay-and-occams-razor/.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Good, I. (1977). Explicativity: a mathematical theory of explanation with statistical applications. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 354(1678):303–330.

Gelman, A. (2011). David MacKay and Occam's Razor. https://statmodeling.stat.columbia.edu/2011/12/04/david-mackay-and-occams-razor/.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Good, I. (1977). Explicativity: a mathematical theory of explanation with statistical applications. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 354(1678):303–330.

Good, I. J. (1968). Corroboration, explanation, evolving probability, simplicity and a sharpened razor. *The British Journal for the Philosophy of Science*, 19(2):123–143.

Gull, S. F. (1988). Bayesian inductive inference and maximum entropy. In *Maximum-entropy and Bayesian methods in science and engineering*, pages 53–74. Springer.

Guyon, I., Vapnik, V., Boser, B., Bottou, L., and Solla, S. A. (1992). Structural risk minimization for character recognition. In *Advances in neural information processing systems*, pages 471–479.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR*.

Immer, A., Bauer, M., Fortuin, V., Rätsch, G., and Khan, M. E. (2021). Scalable marginal likelihood estimation for model selection in deep learning. *arXiv preprint arXiv:2104.04975*.

Iwata, T. and Ghahramani, Z. (2017). Improving output uncertainty estimation and generalization in deep learning via neural network gaussian processes. *arXiv preprint arXiv:1707.05922*.

Jaynes, E. T. (1979). Inference, method, and decision: Towards a bayesian philosophy of science.

Jefferys, W. H. and Berger, J. O. (1991). Sharpening ockham's razor on a bayesian strop. *Tech nical Report*, page 157.

Jeffreys, H. (1939). *The theory of probability*. The Clarendon Press, Oxford.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kendall, M. G. (1948). Rank correlation methods.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. *arXiv preprint arXiv:1506.02557*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lloyd, J., Duvenaud, D., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2014). Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Loredo, T. J. (1990). From laplace to supernova sn 1987a: Bayesian inference in astrophysics. In *Maximum entropy and Bayesian methods*, pages 81–142. Springer.

Lyle, C., Schut, L., Ru, B., Gal, Y., and van der Wilk, M. (2020). A bayesian perspective on training speed and model selection. In *Advances in Neural Information Processing Systems*.

MacKay, D. J. (1992a). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. J. (1992b). A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

MacKay, D. J. (1995). Probable networks and plausible predictions?a review of practical Bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

MacKay, D. J. C. (1992c). *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology.

Good, I. J. (1968). Corroboration, explanation, evolving probability, simplicity and a sharpened razor. *The British Journal for the Philosophy of Science*, 19(2):123–143.

Gull, S. F. (1988). Bayesian inductive inference and maximum entropy. In *Maximum-entropy and Bayesian methods in science and engineering*, pages 53–74. Springer.

Guyon, I., Vapnik, V., Boser, B., Bottou, L., and Solla, S. A. (1992). Structural risk minimization for character recognition. In *Advances in neural information processing systems*, pages 471–479.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR*.

Immer, A., Bauer, M., Fortuin, V., Rätsch, G., and Khan, M. E. (2021). Scalable marginal likelihood estimation for model selection in deep learning. *arXiv preprint arXiv:2104.04975*.

Iwata, T. and Ghahramani, Z. (2017). Improving output uncertainty estimation and generalization in deep learning via neural network gaussian processes. *arXiv preprint arXiv:1707.05922*.

Jaynes, E. T. (1979). Inference, method, and decision: Towards a bayesian philosophy of science.

Jefferys, W. H. and Berger, J. O. (1991). Sharpening ockham's razor on a bayesian strop. *Tech nical Report*, page 157.

Jeffreys, H. (1939). *The theory of probability*. The Clarendon Press, Oxford.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kendall, M. G. (1948). Rank correlation methods.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. *arXiv preprint arXiv:1506.02557*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *International Conference on Learning Representations*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lloyd, J., Duvenaud, D., Grosse, R., Tenenbaum, J., and Ghahramani, Z. (2014). Automatic construction and natural-language description of nonparametric regression models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Loredo, T. J. (1990). From laplace to supernova sn 1987a: Bayesian inference in astrophysics. In *Maximum entropy and Bayesian methods*, pages 81–142. Springer.

Lyle, C., Schut, L., Ru, B., Gal, Y., and van der Wilk, M. (2020). A bayesian perspective on training speed and model selection. In *Advances in Neural Information Processing Systems*.

MacKay, D. J. (1992a). Bayesian interpolation. *Neural Computation*, 4(3):415–447.

MacKay, D. J. (1992b). A practical Bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

MacKay, D. J. (1995). Probable networks and plausible predictions?a review of practical Bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469–505.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

MacKay, D. J. C. (1992c). *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology.

MacKay, D. J. C. (1992d). The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736.

Minka, T. P. (2001). Automatic choice of dimensionality for pca. In *Advances in neural information processing systems*, pages 598–604.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. (2021). The promises and pitfalls of deep kernel learning. *arXiv preprint arXiv:2102.12108*.

MacKay, D. J. C. (1992d). The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736.

Minka, T. P. (2001). Automatic choice of dimensionality for pca. In *Advances in neural information processing systems*, pages 598–604.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. (2021). The promises and pitfalls of deep kernel learning. *arXiv preprint arXiv:2102.12108*.

Patacchiola, M., Turner, J., Crowley, E. J., O'Boyle, M., and Storkey, A. (2020). Bayesian meta-learning for the few-shot setting via deep kernels.

Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. In *Neural Information Processing Systems (NIPS)*.

Rasmussen, C. E. and Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research (JMLR)*, 11:3011–3015.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for Machine Learning*. The MIT Press.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, A. F. and Spiegelhalter, D. J. (1980). Bayes factors and choice criteria for linear models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):213–220.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Wilson, A. G. (2014). *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge.

Wilson, A. G. and Adams, R. P. (2013). Gaussian process kernels for pattern discovery and extrapolation. *International Conference on Machine Learning (ICML)*.

Wilson, A. G., Dann, C., Lucas, C., and Xing, E. P. (2015). The human kernel. In *Advances in Neural Information Processing Systems*, pages 2854–2862.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016a). Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016b). Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378.

Zhang, H., Dauphin, Y. N., and Ma, T. (2019). Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*.

Patacchiola, M., Turner, J., Crowley, E. J., O'Boyle, M., and Storkey, A. (2020). Bayesian meta-learning for the few-shot setting via deep kernels.

Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. In *Neural Information Processing Systems (NIPS)*.

Rasmussen, C. E. and Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research (JMLR)*, 11:3011–3015.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for Machine Learning*. The MIT Press.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable Laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, A. F. and Spiegelhalter, D. J. (1980). Bayes factors and choice criteria for linear models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):213–220.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

van der Wilk, M., Bauer, M., John, S., and Hensman, J. (2018). Learning invariances using the marginal likelihood. In *Advances in Neural Information Processing Systems*, pages 9938–9948.

Wilson, A. G. (2014). *Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes*. PhD thesis, University of Cambridge.

Wilson, A. G. and Adams, R. P. (2013). Gaussian process kernels for pattern discovery and extrapolation. *International Conference on Machine Learning (ICML)*.

Wilson, A. G., Dann, C., Lucas, C., and Xing, E. P. (2015). The human kernel. In *Advances in Neural Information Processing Systems*, pages 2854–2862.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016a). Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2016b). Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. In *Advances in Neural Information Processing Systems*.

Zhang, H., Dauphin, Y. N., and Ma, T. (2019). Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*.

A. The Drawbacks of the Laplace Approximation: Example

We consider the following example to further demonstrate the drawbacks of the Laplace approximation discussed in Section 4.3: we generate data from $x \sim \mathcal{N}(\sin(w), 1)$ with uniform prior $w \sim U[-\alpha, \alpha]$, then estimate the posterior on w and evaluate the marginal likelihood to estimate the parameter α. The posterior is periodic with a period of 2π. Consequently, as we increase α, the marginal likelihood will be roughly constant for $\alpha > w_{MAP}$, where w_{MAP} is the lowest norm maximum a posteriori (MAP) solution, as the ratio of the posterior volume to the prior volume (Occam factor) is roughly constant in this regime. We visualize the posterior and the true LML in Figure 8. However, the Laplace approximation only captures a single mode of the posterior, and thus greatly underestimates the posterior volume. As a result, the Laplace marginal likelihood estimate decreases linearly with α. This toy example shows that Laplace can be problematic for tuning the prior scale in Bayesian neural networks, where covering multiple diverse modes is beneficial for generalization.

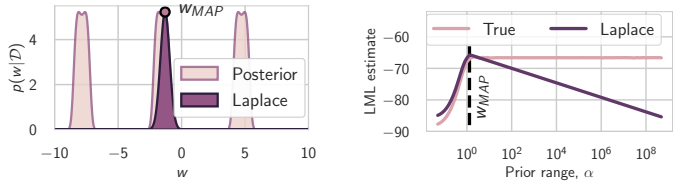


Figure 8. **Laplace approximation.** Model $x \sim \mathcal{N}(\sin(w), 1)$ with a uniform prior $w \sim U[-\alpha, \alpha]$. **(Left):** Posterior density and a Laplace approximation to the posterior (scaled for visualization); **(Right):** True marginal likelihood and the Laplace estimate as a function of α. As Laplace only captures a single mode, the Laplace estimate of marginal likelihood decreases linearly with α while the true marginal likelihood is roughly constant.

B. Variational Inference and ELBO

In variational inference, the evidence lower bound (ELBO), a lower bound on log-marginal likelihood is often used for automatically setting hyperparameters (Hoffman et al., 2013; Kingma and Welling, 2013; Kingma et al., 2015; Alemi et al., 2018). In variational auto-encoders (VAE), the whole decoder network (often, with millions of parameters) is treated as a model hyper-parameter and is trained by maximizing the ELBO (Kingma and Welling, 2014).

The ELBO is given by

$$\log p(\mathcal{D}|\mathcal{M}) \geq \underbrace{\mathbb{E}_{q(w)} \log p(\mathcal{D}|w)}_{\text{Data fit}} - \underbrace{\mathrm{KL}(q(w)||p(w))}_{\text{Complexity penalty}}. \tag{1}$$

In VI for Bayesian neural networks, the posterior is often approximated with a unimodal Gaussian distribution. For a complex model, the ELBO will suffer from the same drawbacks described in Section 4.3 and Appendix A.

C. Overfitting in Gaussian Processes

In Figure 9 we provide a simple example in which LML optimization leads to severe overfitting. We generate a set of 100 evenly spaced points from a GP prior with an RBF kernel with a lengthscale of 0.75 and observation noise of 0.02.

We then use LML optimization to train two GP models on the first 50 data points: the first model is a standard GP with constant mean and an RBF kernel, and the second is a GP with an RBF kernel, but where we have replaced the mean function with a small neural network. The mean function of the second model is a feed forward ReLU network with two hidden layers each with 50 units.

In Figure 9 (left) we see that by fitting the training data with a GP with constant mean and an RBF kernel we do not necessarily extrapolate well, but our prediction is reasonable and our uncertainty appears to be well-calibrated. In Figure 9 (right) we see that in training a GP with an overly flexible mean function we are able to overfit to the training data, and produce extrapolation predictions that are both incorrect, and very confident. By building a model with an incredibly flexible prior, we are able to optimize out LML to concentrate heavily around a single solution. This model has a high likelihood of generating the data, but does not extrapolate well, similar to the effect presented in Figure 1(c).

A. The Drawbacks of the Laplace Approximation: Example

We consider the following example to further demonstrate the drawbacks of the Laplace approximation discussed in Section 4.3: we generate data from $x \sim \mathcal{N}(\sin(w), 1)$ with uniform prior $w \sim U[-\alpha, \alpha]$, then estimate the posterior on w and evaluate the marginal likelihood to estimate the parameter α. The posterior is periodic with a period of 2π. Consequently, as we increase α, the marginal likelihood will be roughly constant for $\alpha > w_{MAP}$, where w_{MAP} is the lowest norm maximum a posteriori (MAP) solution, as the ratio of the posterior volume to the prior volume (Occam factor) is roughly constant in this regime. We visualize the posterior and the true LML in Figure 8. However, the Laplace approximation only captures a single mode of the posterior, and thus greatly underestimates the posterior volume. As a result, the Laplace marginal likelihood estimate decreases linearly with α. This toy example shows that Laplace can be problematic for tuning the prior scale in Bayesian neural networks, where covering multiple diverse modes is beneficial for generalization.

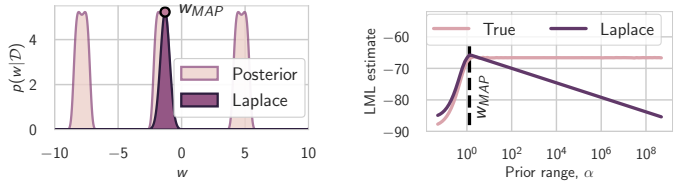


Figure 8. **Laplace approximation.** Model $x \sim \mathcal{N}(\sin(w), 1)$ with a uniform prior $w \sim U[-\alpha, \alpha]$. **(Left):** Posterior density and a Laplace approximation to the posterior (scaled for visualization); **(Right):** True marginal likelihood and the Laplace estimate as a function of α. As Laplace only captures a single mode, the Laplace estimate of marginal likelihood decreases linearly with α while the true marginal likelihood is roughly constant.

B. Variational Inference and ELBO

In variational inference, the evidence lower bound (ELBO), a lower bound on log-marginal likelihood is often used for automatically setting hyperparameters (Hoffman et al., 2013; Kingma and Welling, 2013; Kingma et al., 2015; Alemi et al., 2018). In variational auto-encoders (VAE), the whole decoder network (often, with millions of parameters) is treated as a model hyper-parameter and is trained by maximizing the ELBO (Kingma and Welling, 2014).

The ELBO is given by

$$\log p(\mathcal{D}|\mathcal{M}) \geq \underbrace{\mathbb{E}_{q(w)} \log p(\mathcal{D}|w)}_{\text{Data fit}} - \underbrace{\mathrm{KL}(q(w)||p(w))}_{\text{Complexity penalty}}. \tag{1}$$

In VI for Bayesian neural networks, the posterior is often approximated with a unimodal Gaussian distribution. For a complex model, the ELBO will suffer from the same drawbacks described in Section 4.3 and Appendix A.

C. Overfitting in Gaussian Processes

In Figure 9 we provide a simple example in which LML optimization leads to severe overfitting. We generate a set of 100 evenly spaced points from a GP prior with an RBF kernel with a lengthscale of 0.75 and observation noise of 0.02.

We then use LML optimization to train two GP models on the first 50 data points: the first model is a standard GP with constant mean and an RBF kernel, and the second is a GP with an RBF kernel, but where we have replaced the mean function with a small neural network. The mean function of the second model is a feed forward ReLU network with two hidden layers each with 50 units.

In Figure 9 (left) we see that by fitting the training data with a GP with constant mean and an RBF kernel we do not necessarily extrapolate well, but our prediction is reasonable and our uncertainty appears to be well-calibrated. In Figure 9 (right) we see that in training a GP with an overly flexible mean function we are able to overfit to the training data, and produce extrapolation predictions that are both incorrect, and very confident. By building a model with an incredibly flexible prior, we are able to optimize out LML to concentrate heavily around a single solution. This model has a high likelihood of generating the data, but does not extrapolate well, similar to the effect presented in Figure 1(c).

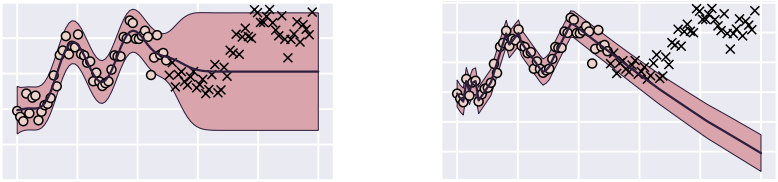


Figure 9. **LML overfitting in Gaussian Processes. Left:** A fit of a GP regression model with a constant prior mean and **Right:** a prior mean parameterized with an MLP and trained via marginal likelihood optimization. Train data is shown with circles, test data is shown with crosses and the shaded region visualizes the 2σ-predictive region of the GP. Given enough flexibility with the prior mean, marginal likelihood overfits the data, providing poor overconfident predictions outside of the train region.

D. Details on the Conditional Marginal Likelihood

Note that unlike LML, CLML depends on the ordering of the datapoints. To remove this undesirable dependence, we can average the CLML over all possible orderings of the data:

$$\frac{1}{n!}\sum_{\sigma\in S_n}\sum_{i=m}^{n}\log p(\mathcal{D}_{\sigma(i)}|\mathcal{D}_{\sigma(1)},\dots,\mathcal{D}_{\sigma(i)},\mathcal{M}), \tag{2}$$

where S_n is the set of all the possible permutations of n elements. Using all the $n!$ permutations is prohibitively expensive in most practical scenarios, so we approximate Eq. (2) as $\frac{1}{|\hat{S}|}\sum_{\sigma\in\hat{S}}\sum_{i=m}^{n}\log p(\mathcal{D}_{\sigma(i)}|\mathcal{D}_{\sigma(1)},\dots,\mathcal{D}_{\sigma(i)},\mathcal{M})$, where $\hat{S}\subset S_n$ is a set containing several random permutations of the dataset. When $\mathcal{D}$ is a large dataset such that $\mathcal{D}_{<m}$ and $\mathcal{D}_{\geq m}$ are both sufficiently large, a single permutation may suffice.

Implementation. For all experiments involving the Laplace approximation, we compute the conditional marginal likelihood as follows:

1. We train a model on 80% of the training data, and fit the LA approximation on the same subset of the data.
2. We tune a hyperparameter T that we use to re-scale the Laplace posterior covariance matrix to ensure that it does not lead to very low BMA accuracies. We choose the value of T that achieves the highest BMA accuracy (average over 20 samples) on 5% of the training data. Our experimental results show that the optimal values of T generally ranges between 0.1 and 0.001, so the LA posterior does not collapse on the MAP solution even as we use this re-scaling parameter.
3. Finally, we directly compute the CLML $p(\mathcal{D}_{\geq m}|\mathcal{D}_{<m})$ using the remaining 15% of the training data. This quantity corresponds simply to the log predictive likelihood of the 15% of the data approximated using a Bayesian model average of LA over 20 samples.

It is important to note that in all our plots, we show the BMA test accuracy and log-likelihood of the model trained on the **entire** training data and for the Laplace approximation fit on the entire training data as well, and not just the 80% subset that we condition the CLML on.

Function space. Note that in the procedure described above, we approximate CLML purely in function space: the estimate only depends on the predictions made by the Bayesian model average and not the values of individual parameters. The standard Laplace approximation of the LML is on the other hand quite sensitive to the number of parameters in the model. Approximating the LML directly in function space is hard, because it would require approximating the integral over the prior with simple Monte Carlo, but sampling from the prior over the weights of a neural network we will never randomly sample the parameters that are likely to generate the data.

E. Details on the Fourier model

We can construct a model that achieves the best of both worlds: strong generalization performance both for small and large training dataset sizes n. To do so, we consider the corrected model $\mathcal{M}_{9c}$, an order-9 model with a prior $a_d, b_d \sim$

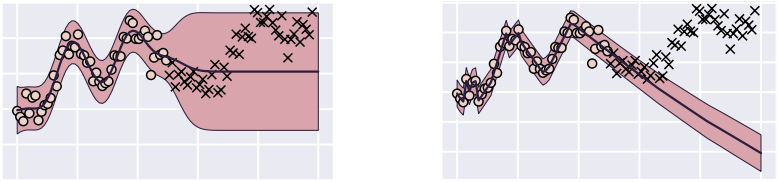


Figure 9. **LML overfitting in Gaussian Processes. Left:** A fit of a GP regression model with a constant prior mean and **Right:** a prior mean parameterized with an MLP and trained via marginal likelihood optimization. Train data is shown with circles, test data is shown with crosses and the shaded region visualizes the 2σ-predictive region of the GP. Given enough flexibility with the prior mean, marginal likelihood overfits the data, providing poor overconfident predictions outside of the train region.

D. Details on the Conditional Marginal Likelihood

Note that unlike LML, CLML depends on the ordering of the datapoints. To remove this undesirable dependence, we can average the CLML over all possible orderings of the data:

$$\frac{1}{n!}\sum_{\sigma\in S_n}\sum_{i=m}^{n}\log p(\mathcal{D}_{\sigma(i)}|\mathcal{D}_{\sigma(1)},\dots,\mathcal{D}_{\sigma(i)},\mathcal{M}), \tag{2}$$

where S_n is the set of all the possible permutations of n elements. Using all the $n!$ permutations is prohibitively expensive in most practical scenarios, so we approximate Eq. (2) as $\frac{1}{|\hat{S}|}\sum_{\sigma\in\hat{S}}\sum_{i=m}^{n}\log p(\mathcal{D}_{\sigma(i)}|\mathcal{D}_{\sigma(1)},\dots,\mathcal{D}_{\sigma(i)},\mathcal{M})$, where $\hat{S}\subset S_n$ is a set containing several random permutations of the dataset. When $\mathcal{D}$ is a large dataset such that $\mathcal{D}_{<m}$ and $\mathcal{D}_{\geq m}$ are both sufficiently large, a single permutation may suffice.

Implementation. For all experiments involving the Laplace approximation, we compute the conditional marginal likelihood as follows:

1. We train a model on 80% of the training data, and fit the LA approximation on the same subset of the data.
2. We tune a hyperparameter T that we use to re-scale the Laplace posterior covariance matrix to ensure that it does not lead to very low BMA accuracies. We choose the value of T that achieves the highest BMA accuracy (average over 20 samples) on 5% of the training data. Our experimental results show that the optimal values of T generally ranges between 0.1 and 0.001, so the LA posterior does not collapse on the MAP solution even as we use this re-scaling parameter.
3. Finally, we directly compute the CLML $p(\mathcal{D}_{\geq m}|\mathcal{D}_{<m})$ using the remaining 15% of the training data. This quantity corresponds simply to the log predictive likelihood of the 15% of the data approximated using a Bayesian model average of LA over 20 samples.

It is important to note that in all our plots, we show the BMA test accuracy and log-likelihood of the model trained on the **entire** training data and for the Laplace approximation fit on the entire training data as well, and not just the 80% subset that we condition the CLML on.

Function space. Note that in the procedure described above, we approximate CLML purely in function space: the estimate only depends on the predictions made by the Bayesian model average and not the values of individual parameters. The standard Laplace approximation of the LML is on the other hand quite sensitive to the number of parameters in the model. Approximating the LML directly in function space is hard, because it would require approximating the integral over the prior with simple Monte Carlo, but sampling from the prior over the weights of a neural network we will never randomly sample the parameters that are likely to generate the data.

E. Details on the Fourier model

We can construct a model that achieves the best of both worlds: strong generalization performance both for small and large training dataset sizes n. To do so, we consider the corrected model $\mathcal{M}_{9c}$, an order-9 model with a prior $a_d, b_d \sim$

$\mathcal{N}(0, (1/d^2)^2)$, following the data generating process. We show the data fit and the learning curve for $\mathcal{M}_{9c}$ in Figure 4(d) and (e) (bottom). While the predictive distribution of $\mathcal{M}_{9c}$ is almost identical to that of $\mathcal{M}_9$ (see Figure 4(e) (middle)), $\mathcal{M}_{9c}$ achieves the best marginal likelihood (comparable to $\mathcal{M}_3$) and the best CLML (comparable to $\mathcal{M}_9$) when evaluated on 100 datapoints. In the learning curve, $\mathcal{M}_{9c}$ provides comparable performance to $\mathcal{M}_3$ for small n, and comparable performance to $\mathcal{M}_9$ for large n.

F. Training speed of Deep Neural Networks

Experimental details We consider 6 deep neural networks with architectures shown in Table F (LeCun et al., 1998; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Huang et al., 2017). We also consider 8 different sizes of training datasets, $\{250, 500, 1000, 2000, 5000, 10,000, 20,000, 45,000\}$, each constructed by randomly sampling a subset of CIFAR-10. To produce a MAP estimate, we train a neural network using hyperparameters found in Table 2. All models are trained using SGD with weight decay coefficient 0.0005, momentum coefficient of 0.9, initial learning rate 0.1, and learning rate drops by a factor of 10 after $\frac{1}{2}$ and $\frac{3}{4}$ of the epochs. Data augmentations include random horizontal flips and crops. We use the diagonal Laplace approximation to approximate the marginal likelihood and perform a Bayesian model average over 20 samples to obtain the BMA test accuracy and log-likelihood. The CLML is computed using a $80\% - 20\%$ split of the training data as described in detail in Section D. We note that the BMA test accuracy and log-likelihood that we show in Figure 10 are computed with respect to all available training data and not just 80% of it.

Discussion Figure 10 (b) shows the ranking of the models according to their generalization performance, where a lower ranking indicates a higher value of the BMA test accuracy. In particular, we see that `VGG19` and `GoogLeNet` train faster than `ResNet-18` and `DenseNet121` but generalize worse for bigger sizes of the CIFAR-10 dataset. This observation extends to many neural architectures that can perform better or worse depending on the size of the dataset (Dosovitskiy et al., 2020). This proves that a faster training speed does not necessarily imply a better generalization performance. Results in Figure 10 (a) (left) are coherent with our conclusions from the Fourier example: the correlation of the BMA test log-likelihood with the LML is positive for small sizes of the training data and negative for higher sizes, whereas the correlation with CLML is consistently positive. Finally, Figure 10 (a) (right) shows that the LML approximated with LA heavily penalizes the number of parameters, which reflects the sensitivity of LA to the number of parameters as discussed in Sections 4.3.

Architecture	Number of parameters
`LeNet`	$62,006$
`ResNet-6`	$609,354$
`GoogLeNet`	$6,166,250$
`DenseNet121`	$6,956,298$
`VGG19`	$20,040,522$
`ResNet-18`	$11,173,962$

Table 1. Neural architectures that we consider in Section 5.

# Samples	Epochs	Batch Size
250	900	32
500	900	32
1000	900	32
2000	600	64
5000	600	64
10000	300	128
20000	300	128
45000	300	128

Table 2. Training hyperparameters for experiments that we consider for training neural networks in Section 5.

G. The Density Estimation Example

Consider the generative process where x is generated using a Gaussian distribution $\mathcal{N}(u, 1)$ and the mean parameter is in turn generated using a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Given dataset $D = \{x_i\}_1^N$, we can show that the marginal likelihood is equal to,

$$p(D|\sigma, \mu) = \mathcal{N}\left(\mu_N, I + \sigma^2 1_{N,N}\right),$$

where $\mu_N = \mu \times 1_N = \underbrace{\begin{bmatrix}\mu & \dots & \mu\end{bmatrix}^T}_{N}$, and $1_N \in \mathbb{R}^N$ is a column vector with all N elements equal to 1.

Proof. Indeed, we have $x_i = u + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 1)$, and $u \sim \mathcal{N}(\mu, \sigma^2)$. Thus, the observations x_i are jointly Gaussian

$\mathcal{N}(0, (1/d^2)^2)$, following the data generating process. We show the data fit and the learning curve for $\mathcal{M}_{9c}$ in Figure 4(d) and (e) (bottom). While the predictive distribution of $\mathcal{M}_{9c}$ is almost identical to that of $\mathcal{M}_9$ (see Figure 4(e) (middle)), $\mathcal{M}_{9c}$ achieves the best marginal likelihood (comparable to $\mathcal{M}_3$) and the best CLML (comparable to $\mathcal{M}_9$) when evaluated on 100 datapoints. In the learning curve, $\mathcal{M}_{9c}$ provides comparable performance to $\mathcal{M}_3$ for small n, and comparable performance to $\mathcal{M}_9$ for large n.

F. Training speed of Deep Neural Networks

Experimental details We consider 6 deep neural networks with architectures shown in Table F (LeCun et al., 1998; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Huang et al., 2017). We also consider 8 different sizes of training datasets, $\{250, 500, 1000, 2000, 5000, 10,000, 20,000, 45,000\}$, each constructed by randomly sampling a subset of CIFAR-10. To produce a MAP estimate, we train a neural network using hyperparameters found in Table 2. All models are trained using SGD with weight decay coefficient 0.0005, momentum coefficient of 0.9, initial learning rate 0.1, and learning rate drops by a factor of 10 after $\frac{1}{2}$ and $\frac{3}{4}$ of the epochs. Data augmentations include random horizontal flips and crops. We use the diagonal Laplace approximation to approximate the marginal likelihood and perform a Bayesian model average over 20 samples to obtain the BMA test accuracy and log-likelihood. The CLML is computed using a $80\% - 20\%$ split of the training data as described in detail in Section D. We note that the BMA test accuracy and log-likelihood that we show in Figure 10 are computed with respect to all available training data and not just 80% of it.

Discussion Figure 10 (b) shows the ranking of the models according to their generalization performance, where a lower ranking indicates a higher value of the BMA test accuracy. In particular, we see that `VGG19` and `GoogLeNet` train faster than `ResNet-18` and `DenseNet121` but generalize worse for bigger sizes of the CIFAR-10 dataset. This observation extends to many neural architectures that can perform better or worse depending on the size of the dataset (Dosovitskiy et al., 2020). This proves that a faster training speed does not necessarily imply a better generalization performance. Results in Figure 10 (a) (left) are coherent with our conclusions from the Fourier example: the correlation of the BMA test log-likelihood with the LML is positive for small sizes of the training data and negative for higher sizes, whereas the correlation with CLML is consistently positive. Finally, Figure 10 (a) (right) shows that the LML approximated with LA heavily penalizes the number of parameters, which reflects the sensitivity of LA to the number of parameters as discussed in Sections 4.3.

Architecture	Number of parameters
`LeNet`	$62,006$
`ResNet-6`	$609,354$
`GoogLeNet`	$6,166,250$
`DenseNet121`	$6,956,298$
`VGG19`	$20,040,522$
`ResNet-18`	$11,173,962$

Table 1. Neural architectures that we consider in Section 5.

# Samples	Epochs	Batch Size
250	900	32
500	900	32
1000	900	32
2000	600	64
5000	600	64
10000	300	128
20000	300	128
45000	300	128

Table 2. Training hyperparameters for experiments that we consider for training neural networks in Section 5.

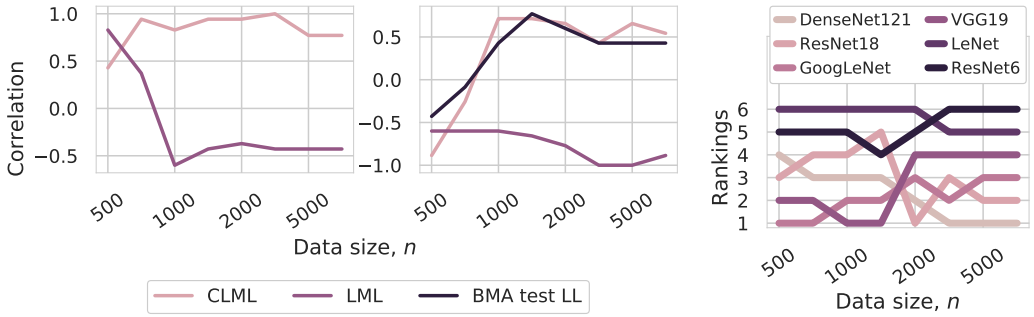
G. The Density Estimation Example

Consider the generative process where x is generated using a Gaussian distribution $\mathcal{N}(u, 1)$ and the mean parameter is in turn generated using a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Given dataset $D = \{x_i\}_1^N$, we can show that the marginal likelihood is equal to,

$$p(D|\sigma, \mu) = \mathcal{N}\left(\mu_N, I + \sigma^2 1_{N,N}\right),$$

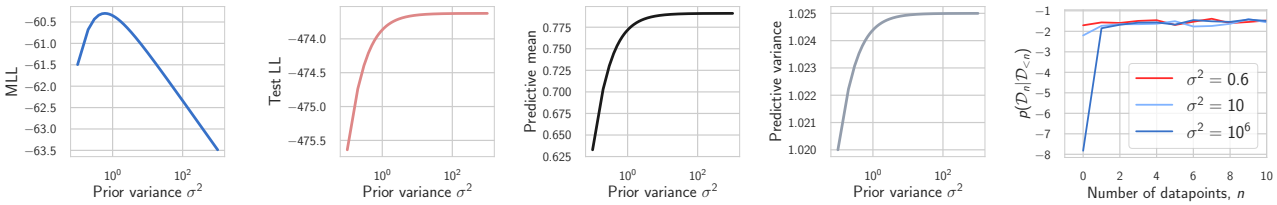
where $\mu_N = \mu \times 1_N = \underbrace{\begin{bmatrix}\mu & \dots & \mu\end{bmatrix}^T}_{N}$, and $1_N \in \mathbb{R}^N$ is a column vector with all N elements equal to 1.

Proof. Indeed, we have $x_i = u + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 1)$, and $u \sim \mathcal{N}(\mu, \sigma^2)$. Thus, the observations x_i are jointly Gaussian



(a) Correlations with the BMA LL and the inverse number of parameters (b) Rankings of DNNs

Figure 10. **Training speed, generalization, and LML for deep neural networks (DNN). (a)**: **(Left)**: The correlation of the CLML and the LML with the BMA test log-likelihood. We show that the LML correlated positively with the BMA test log-likelihood (LL) for small sizes of the training data, but negatively for larger sizes, whereas CLML is correlated consistently positively with the BMA test log-likelihood for all sizes of the data. **(Right)**: The correlation of of the CLML, LML and the BMA test log-likelihood with the inverse number of parameters. The LML approximated with LA correlates negatively with number of parameters and assigns higher values to more constrained models. This negative correlation might reflect one of the sensitivity of the Laplace approximation to the number of parameters as discussed in Section 4.3. **(b)**: Ranking of DNNs according to their Bayesian model average (BMA) test accuracy approximated with LA for different sizes of the CIFAR-10 training data. A lower ranking indicates a higher BMA test accuracy. `VGG19` and `GoogLeNet`, in contrast with `ResNet-18` and `DenseNet121`, train faster but generalize worse for bigger sizes of the CIFAR-10 dataset.



(a) Marginal log-likelihood (b) Test log-likelihood (c) Predictive mean (d) Predictive variance (e) Learning curves

Figure 11. **Density estimation (details). (a)**: Marginal log-likelihood, **(b)**: test log-likelihood and **(c)**: mean and **(d)**: variance of the predictive distribution as a function of the prior variance σ^2. The predictive distribution is virtually constant with respect to prior variance for $\sigma^2 > 10$, while marginal likelihood sharply decreases. **(e)**: Learning curves for three different choices of prior standard deviation σ^2. After observing $n = 5$ datapoints, the predictive distributions are almost indistinguishable between the different values of σ^2, but due to $\sigma^2 = 0.6$ providing the best fit for $n = 1$ datapoint, marginal likelihood strongly prefers this choice.

with a mean $\mathbb{E}x_i = \mathbb{E}(u + \epsilon_i) = \mathbb{E}u = \mu$. The covariance structure is given by

$$\begin{aligned}\operatorname{cov}(x_i, x_j) = \mathbb{E}(x_i - \mu) \cdot (x_j - \mu) = \mathbb{E}(u + \epsilon_i - \mu) \cdot (u + \epsilon_j - \mu) = \mathbb{E}(u - \mu) \cdot (u - \mu) + \mathbb{E}\epsilon_i\epsilon_j \\ = \operatorname{cov}(u, u) + \operatorname{cov}(\epsilon_i, \epsilon_j) = \sigma^2 + \delta_{ij},\end{aligned}$$

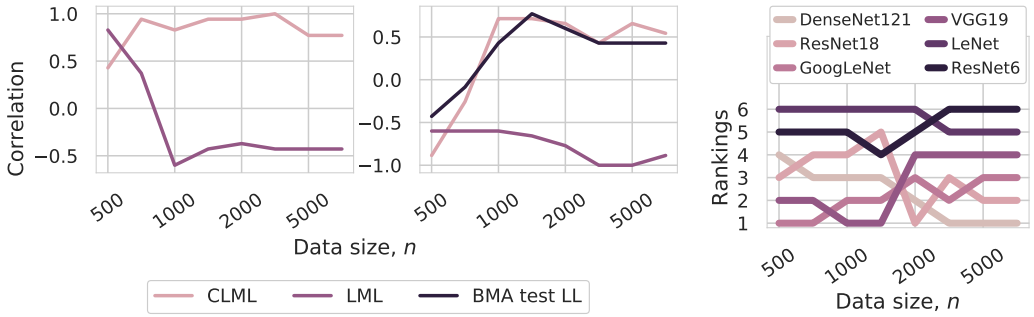
where δ_{ij} is equal to 1 if $i = j$ and 0 otherwise. Thus, we get $\mathcal{D} \sim \mathcal{N}(\mu_N, I + \sigma^2 1_{N,N})$. □

The posterior distribution is equal to,

$$p(u|D, \sigma, \mu) = \mathcal{N}\left(\frac{1}{1/\sigma^2 + N}\left(\sum_{i=1}^{N} x_i + \frac{1}{\sigma^2}\mu\right), \frac{1}{1/\sigma^2 + N}\right). \tag{3}$$

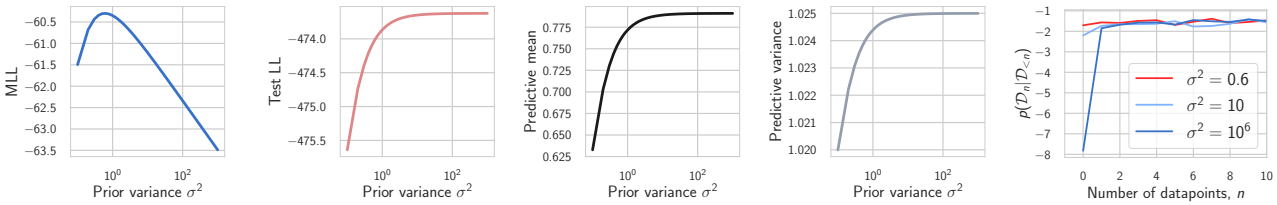
Proof. Let us denote $x = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$. We can write down the joint distribution over x and u, following from the derivation marginal distribution of x above:

$$\begin{bmatrix} x \\ u \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_N \\ \mu \end{bmatrix}, \begin{bmatrix} 1_N 1_N^T \sigma^2 + I & 1_N \cdot \sigma^2 \\ 1_N^T \cdot \sigma^2 & \sigma^2 \end{bmatrix}\right).$$



(a) Correlations with the BMA LL and the inverse number of parameters (b) Rankings of DNNs

Figure 10. **Training speed, generalization, and LML for deep neural networks (DNN). (a)**: **(Left)**: The correlation of the CLML and the LML with the BMA test log-likelihood. We show that the LML correlated positively with the BMA test log-likelihood (LL) for small sizes of the training data, but negatively for larger sizes, whereas CLML is correlated consistently positively with the BMA test log-likelihood for all sizes of the data. **(Right)**: The correlation of of the CLML, LML and the BMA test log-likelihood with the inverse number of parameters. The LML approximated with LA correlates negatively with number of parameters and assigns higher values to more constrained models. This negative correlation might reflect one of the sensitivity of the Laplace approximation to the number of parameters as discussed in Section 4.3. **(b)**: Ranking of DNNs according to their Bayesian model average (BMA) test accuracy approximated with LA for different sizes of the CIFAR-10 training data. A lower ranking indicates a higher BMA test accuracy. `VGG19` and `GoogLeNet`, in contrast with `ResNet-18` and `DenseNet121`, train faster but generalize worse for bigger sizes of the CIFAR-10 dataset.



(a) Marginal log-likelihood (b) Test log-likelihood (c) Predictive mean (d) Predictive variance (e) Learning curves

Figure 11. **Density estimation (details). (a)**: Marginal log-likelihood, **(b)**: test log-likelihood and **(c)**: mean and **(d)**: variance of the predictive distribution as a function of the prior variance σ^2. The predictive distribution is virtually constant with respect to prior variance for $\sigma^2 > 10$, while marginal likelihood sharply decreases. **(e)**: Learning curves for three different choices of prior standard deviation σ^2. After observing $n = 5$ datapoints, the predictive distributions are almost indistinguishable between the different values of σ^2, but due to $\sigma^2 = 0.6$ providing the best fit for $n = 1$ datapoint, marginal likelihood strongly prefers this choice.

with a mean $\mathbb{E}x_i = \mathbb{E}(u + \epsilon_i) = \mathbb{E}u = \mu$. The covariance structure is given by

$$\begin{aligned}\operatorname{cov}(x_i, x_j) = \mathbb{E}(x_i - \mu) \cdot (x_j - \mu) = \mathbb{E}(u + \epsilon_i - \mu) \cdot (u + \epsilon_j - \mu) = \mathbb{E}(u - \mu) \cdot (u - \mu) + \mathbb{E}\epsilon_i\epsilon_j \\ = \operatorname{cov}(u, u) + \operatorname{cov}(\epsilon_i, \epsilon_j) = \sigma^2 + \delta_{ij},\end{aligned}$$

where δ_{ij} is equal to 1 if $i = j$ and 0 otherwise. Thus, we get $\mathcal{D} \sim \mathcal{N}(\mu_N, I + \sigma^2 1_{N,N})$. □

The posterior distribution is equal to,

$$p(u|D, \sigma, \mu) = \mathcal{N}\left(\frac{1}{1/\sigma^2 + N}\left(\sum_{i=1}^{N} x_i + \frac{1}{\sigma^2}\mu\right), \frac{1}{1/\sigma^2 + N}\right). \tag{3}$$

Proof. Let us denote $x = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}^T$. We can write down the joint distribution over x and u, following from the derivation marginal distribution of x above:

$$\begin{bmatrix} x \\ u \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_N \\ \mu \end{bmatrix}, \begin{bmatrix} 1_N 1_N^T \sigma^2 + I & 1_N \cdot \sigma^2 \\ 1_N^T \cdot \sigma^2 & \sigma^2 \end{bmatrix}\right).$$

Using the properties of Gaussian distributions, we can compute the posterior $p(u|x)$ as a Gaussian conditional (see e.g. Ch. 3 of Bishop, 2006b):

$$u \mid x \sim \mathcal{N}\Big(\mu + 1_N^T \cdot \sigma^2 \cdot (1_N 1_N^T \sigma^2 + I)^{-1} \cdot (x - \mu_N), \sigma^2 - 1_N^T \cdot \sigma^2 \cdot (1_N 1_N^T \sigma^2 + I)^{-1} 1_N \cdot \sigma^2\Big). \tag{4}$$

Now, note that $(1_N 1_N^T \sigma^2 + I)^{-1} = I - \frac{\sigma^2}{1+N\sigma^2} 1_N 1_N^T$ which can be verified by direct multiplication. Substituting the expression for the inverse in Eq. (4), we recover Eq. (3).

□

The predictive distribution is equal to,

$$p(x^*|D, \sigma, \mu) = \mathcal{N}\left(\frac{1}{1/\sigma^2 + N}\left(\sum_{i=1}^{N} x_i + \frac{1}{\sigma^2}\mu\right), 1 + \frac{1}{1/\sigma^2 + N}\right). \tag{5}$$

Proof. Conditioned on u, the observations are Gaussian: $p(x^*|u) = \mathcal{N}(u, 1)$. Furthermore, we have shown that the posterior $p(u|D, \sigma, \mu) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ is also Gaussian, with the parameters $\hat{\mu}, \hat{\sigma}^2$ given by Eq. (3). Then, the predictive distribution is simply $p(p(x^*|D, \sigma, \mu) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2 + 1)$, recovering Eq. (5). □

We see that as the variance of the prior mean $\sigma^2 \to +\infty$, both the predictive distribution and the posterior distribution do not depend on this hyperparameter, whereas the marginal likelihood depends on it. This is another example whereas the marginal likelihood is more sensitive to a hyperparameter that has little influence on the quality of future predictions. Hence, the potential mismatch between marginal likelihood and generation.

H. Neural Architecture Search Details

We investigate the correlation between the log marginal likelihood (LML) and generalization in the context of image classification using the CIFAR-10 and CIFAR-100 datasets. In particular, we consider two tasks: (1) model selection with fixed prior precision, and (2) tuning the prior precision then performing a similar model selection task.

Experimental details We reconstruct the neural architecture search experiments with convolutional (CNN) and residual (ResNet) networks for CIFAR-100 and CIFAR-10 from Immer et al. (2021). We use the same architectures:

- The CNNs consist of up to 5 blocks of 33 convolutions, followed by a ReLU activation function, and MaxPooling, except in the first layer. BatchNorm is replaced by the fixup initialization (Zhang et al., 2019) as in Immer et al. (2021). The width (number of channels) of the first channel varies from 2 to 32 for both datasets. The last layer is a fully-connected layer to the class logit.
- ResNets of depths varying from 8 to 32 are used for CIFAR-10 and from 20 to 101 for CIFAR-100. The width varies from 16 to 48 for CIFAR-10 and from 32 to 64 for CIFAR-100.

All models were trained for 250 epochs with an SGD optimizer and an initial learning rate of 0.01. The batch-size was fixed to 128. For experiments where the prior precision was optimized, we used online optimization where the prior precision was updated every 5 epochs for 100 iterations using an Adam optimizer with an initial learning equal to 1.0.

For all experiments in this section, we used the Kronecker Laplace approximation and computed the BMA test accuracy and log-likelihood by averaging over 20 samples. The CLML was computed using a $80\% - 20\%$ split of the training data as described in detail in Section D. We note that the BMA test accuracy and log-likelihood that we show in all figures are computed using all available training data and not just 80% of it that we condition CLML on.

Discussion We visualize the correlation of the LML and the CLML in the top and bottom rows of Figure 12, respectively for CIFAR-100. We also report the Spearman's correlation coefficient ρ (Kendall, 1948), which measures the correlation between the model rankings according to the BMA test accuracy and the LML/CLML. We see that the LML correlates positively with the BMA test accuracy for high values of the prior precision, but negatively for lower values. This can be understood in the light of what we discussed in Section 4.1; the LML penalizes low values of the prior precision because

Using the properties of Gaussian distributions, we can compute the posterior $p(u|x)$ as a Gaussian conditional (see e.g. Ch. 3 of Bishop, 2006b):

$$u \mid x \sim \mathcal{N}\Big(\mu + 1_N^T \cdot \sigma^2 \cdot (1_N 1_N^T \sigma^2 + I)^{-1} \cdot (x - \mu_N), \sigma^2 - 1_N^T \cdot \sigma^2 \cdot (1_N 1_N^T \sigma^2 + I)^{-1} 1_N \cdot \sigma^2\Big). \tag{4}$$

Now, note that $(1_N 1_N^T \sigma^2 + I)^{-1} = I - \frac{\sigma^2}{1+N\sigma^2} 1_N 1_N^T$ which can be verified by direct multiplication. Substituting the expression for the inverse in Eq. (4), we recover Eq. (3).

□

The predictive distribution is equal to,

$$p(x^*|D, \sigma, \mu) = \mathcal{N}\left(\frac{1}{1/\sigma^2 + N}\left(\sum_{i=1}^{N} x_i + \frac{1}{\sigma^2}\mu\right), 1 + \frac{1}{1/\sigma^2 + N}\right). \tag{5}$$

Proof. Conditioned on u, the observations are Gaussian: $p(x^*|u) = \mathcal{N}(u, 1)$. Furthermore, we have shown that the posterior $p(u|D, \sigma, \mu) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ is also Gaussian, with the parameters $\hat{\mu}, \hat{\sigma}^2$ given by Eq. (3). Then, the predictive distribution is simply $p(p(x^*|D, \sigma, \mu) = \mathcal{N}(\hat{\mu}, \hat{\sigma}^2 + 1)$, recovering Eq. (5). □

We see that as the variance of the prior mean $\sigma^2 \to +\infty$, both the predictive distribution and the posterior distribution do not depend on this hyperparameter, whereas the marginal likelihood depends on it. This is another example whereas the marginal likelihood is more sensitive to a hyperparameter that has little influence on the quality of future predictions. Hence, the potential mismatch between marginal likelihood and generation.

H. Neural Architecture Search Details

We investigate the correlation between the log marginal likelihood (LML) and generalization in the context of image classification using the CIFAR-10 and CIFAR-100 datasets. In particular, we consider two tasks: (1) model selection with fixed prior precision, and (2) tuning the prior precision then performing a similar model selection task.

Experimental details We reconstruct the neural architecture search experiments with convolutional (CNN) and residual (ResNet) networks for CIFAR-100 and CIFAR-10 from Immer et al. (2021). We use the same architectures:

- The CNNs consist of up to 5 blocks of 33 convolutions, followed by a ReLU activation function, and MaxPooling, except in the first layer. BatchNorm is replaced by the fixup initialization (Zhang et al., 2019) as in Immer et al. (2021). The width (number of channels) of the first channel varies from 2 to 32 for both datasets. The last layer is a fully-connected layer to the class logit.
- ResNets of depths varying from 8 to 32 are used for CIFAR-10 and from 20 to 101 for CIFAR-100. The width varies from 16 to 48 for CIFAR-10 and from 32 to 64 for CIFAR-100.

All models were trained for 250 epochs with an SGD optimizer and an initial learning rate of 0.01. The batch-size was fixed to 128. For experiments where the prior precision was optimized, we used online optimization where the prior precision was updated every 5 epochs for 100 iterations using an Adam optimizer with an initial learning equal to 1.0.

For all experiments in this section, we used the Kronecker Laplace approximation and computed the BMA test accuracy and log-likelihood by averaging over 20 samples. The CLML was computed using a $80\% - 20\%$ split of the training data as described in detail in Section D. We note that the BMA test accuracy and log-likelihood that we show in all figures are computed using all available training data and not just 80% of it that we condition CLML on.

Discussion We visualize the correlation of the LML and the CLML in the top and bottom rows of Figure 12, respectively for CIFAR-100. We also report the Spearman's correlation coefficient ρ (Kendall, 1948), which measures the correlation between the model rankings according to the BMA test accuracy and the LML/CLML. We see that the LML correlates positively with the BMA test accuracy for high values of the prior precision, but negatively for lower values. This can be understood in the light of what we discussed in Section 4.1; the LML penalizes low values of the prior precision because

they correspond to diffuse priors. A similar trend is observed in Figure 13 for the MAP test accuracy, Figure 14 for the BMA test log-likelihood, and Figure 15 for the MAP test log-likelihood for the CIFAR-100 dataset. Similar results are obtained for CIFAR-10 in Figures 16, 17, 18, 19.

Figure 20 shows that optimizing the global and layerwise prior precision helps improve the correlation between LML and the BMA test accuracy. To understand this effect, consider Figure 4(a): two models with the same test performance can have very different values of the marginal likelihood depending on the prior variance. However, if we update the prior variance such that it maximizes the LML, we can expect that the final prior variance to be low without a major effect on the accuracy, therefore leading to a positive correlation between the LML and the BMA test accuracy.

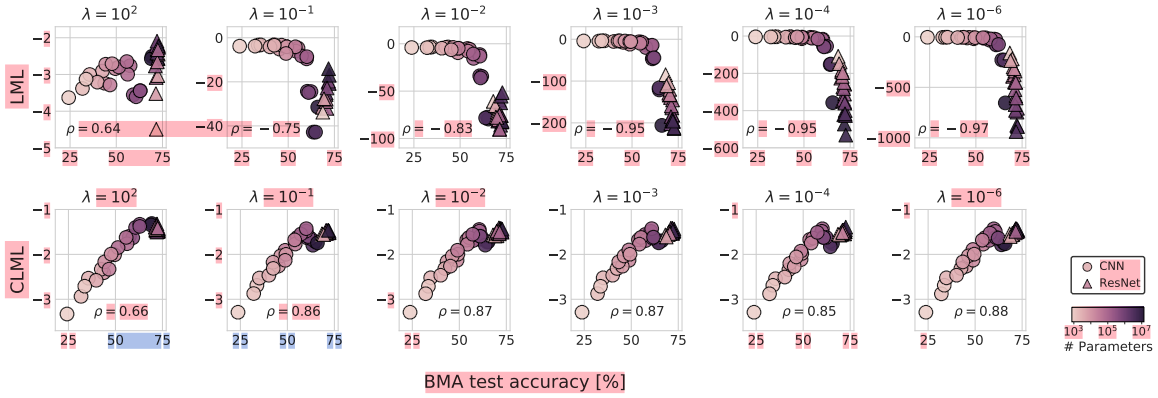


Figure 12. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test accuracy and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test accuracy and the conditional log marginal likelihood (CLML). The LML correlates positively with the BMA test accuracy for high values of the prior precision, and negatively for low values of the prior precision (vague priors). The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test accuracy.

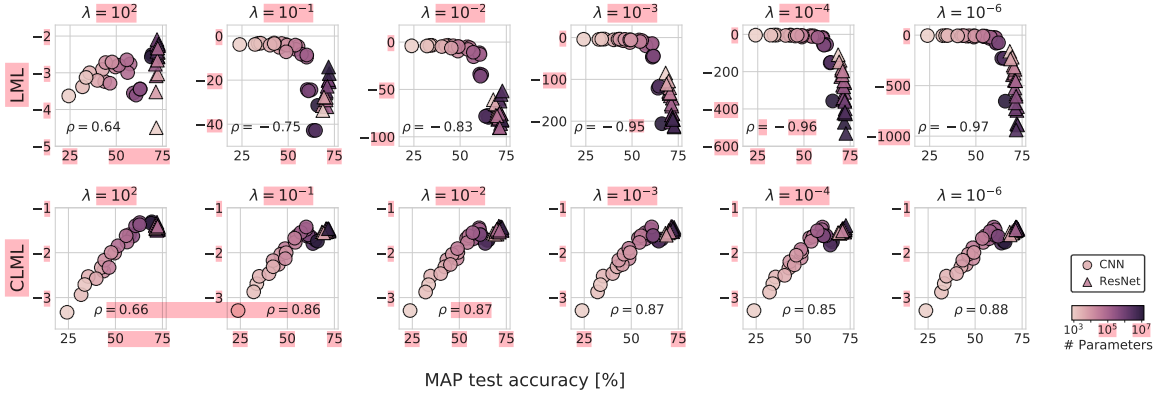


Figure 13. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the maximum-a-posterior (MAP) test accuracy and the LML. **(Top row)**: Correlation between the MAP test accuracy and the CLML. The LML correlates positively with the MAP test accuracy for high values of the prior precision, and negatively for low values of the prior precision, which correspond to vague priors. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test accuracy.

I. Extended Gaussian Process Results

GPs: RBF kernel. In Figure 21(a) we illustrate the bias of LML towards underfitting. We follow the experiment presented in Wilson et al. (2015), and generate 100 datasets from an RBF Gaussian process prior with a lengthscale of $l = 4$. The

they correspond to diffuse priors. A similar trend is observed in Figure 13 for the MAP test accuracy, Figure 14 for the BMA test log-likelihood, and Figure 15 for the MAP test log-likelihood for the CIFAR-100 dataset. Similar results are obtained for CIFAR-10 in Figures 16, 17, 18, 19.

Figure 20 shows that optimizing the global and layerwise prior precision helps improve the correlation between LML and the BMA test accuracy. To understand this effect, consider Figure 4(a): two models with the same test performance can have very different values of the marginal likelihood depending on the prior variance. However, if we update the prior variance such that it maximizes the LML, we can expect that the final prior variance to be low without a major effect on the accuracy, therefore leading to a positive correlation between the LML and the BMA test accuracy.

Comparison to the validation loss Figure 21 shows the correlation between the BMA test accuracy and the CLML (left) as opposed to the negative validation loss (right). The negative validation loss correlates positively overall with the BMA test accuracy for CNNs, but not for ResNets, resulting in a negative total correlation for low prior precision values. In contrast, the CLML exhibits a positive correlation with the BMA test accuracy across different values of the prior precision.

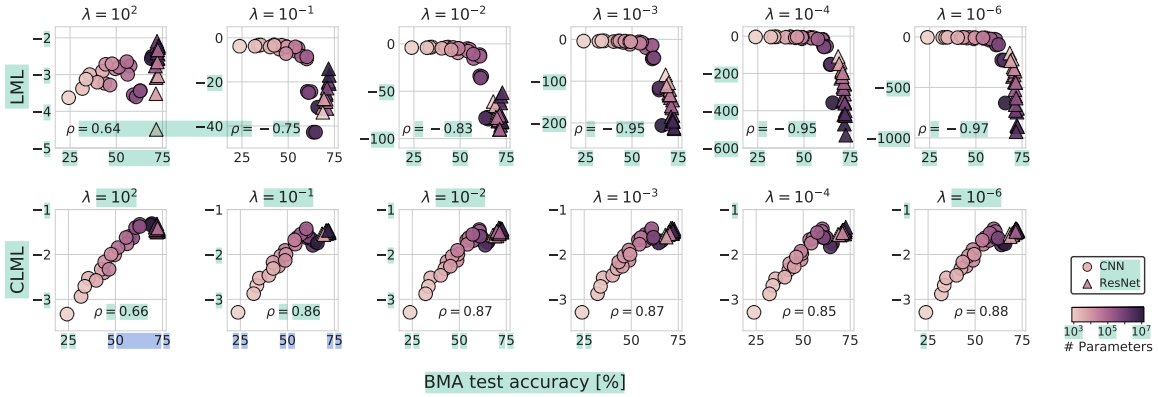


Figure 12. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test accuracy and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test accuracy and the conditional log marginal likelihood (CLML). The LML correlates positively with the BMA test accuracy for high values of the prior precision, and negatively for low values of the prior precision (vague priors). The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test accuracy.

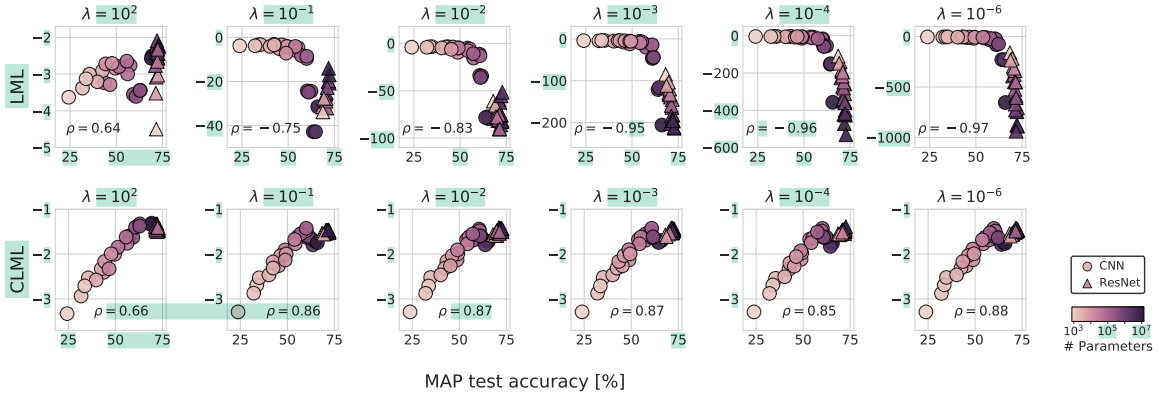


Figure 13. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the maximum-a-posterior (MAP) test accuracy and the LML. **(Top row)**: Correlation between the MAP test accuracy and the CLML. The LML correlates positively with the MAP test accuracy for high values of the prior precision, and negatively for low values of the prior precision, which correspond to vague priors. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test accuracy.

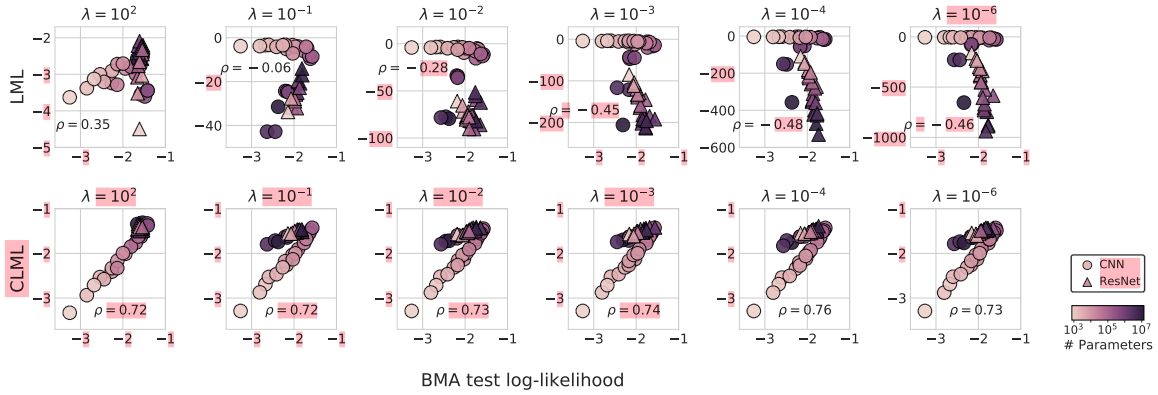


Figure 14. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test log-likelihood and the conditional log marginal likelihood (CLML). The LML correlates positively with the BMA test log-likelihood for high values of the prior precision, and negatively for low values of the prior precision (vague priors). The correlation shifts around $\lambda = 10^{-1}$ as it remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test log-likelihood.

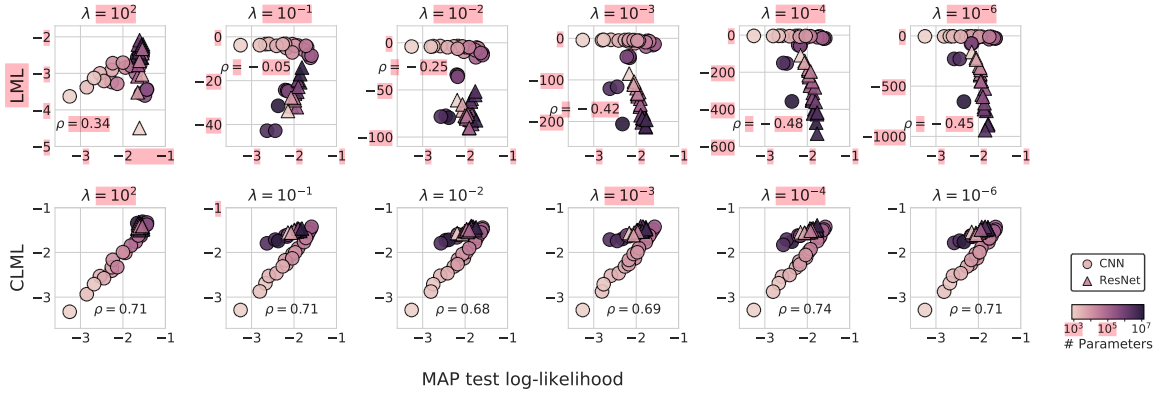


Figure 15. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the MAP test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the MAP test log-likelihood and the conditional log marginal likelihood (CLML). The LML correlates positively with the MAP test log-likelihood for high values of the prior precision, and negatively for low values of the prior precision (vague priors). We can that the correlation shift occurs around $\lambda = 10^{-1}$ as the correlation remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test log-likelihood.

datapoints are located at positions $\{1, \ldots, 150\}$, the output scale is 1, and the observation noise is 0.2. For each dataset and each $n \in \{1, \ldots, 150\}$, we fit a new GP model to the the first n datapoints of the dataset: we maximize the LML or CLML with respect to the lengthscale of the RBF kernel, using the ground truth value $l = 4$ as the initialization. We plot the learned lengthscales averaged over the datasets as a function of n in 21(a). This experiment illustrates a unique quality of marginal likelihood that distinguishes it from conventional maximum likelihood training: while low lengthscales would lead to a better fit of the training data, marginal likelihood has a *bias towards underfitting* in the data space. Indeed, LML consistently selects lengthscales that are larger than the the lengthscale that was used to generate the data, especially for small n. We note that CLML does not remove this bias, and provides a very similar curve.

GPs: RQ kernel. Above, we have seen how marginal likelihood can over-estimate the lengthscale of an RBF kernel leading to underfitting in data space. Here, we construct a more extreme example of this behavior using the rational quadratic

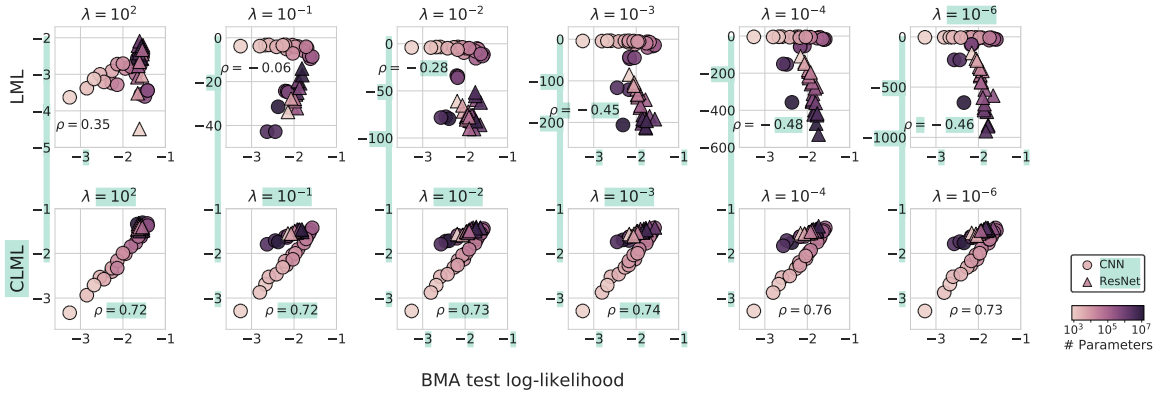


Figure 14. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test log-likelihood and the conditional log marginal likelihood (CLML). The LML correlates positively with the BMA test log-likelihood for high values of the prior precision, and negatively for low values of the prior precision (vague priors). The correlation shifts around $\lambda = 10^{-1}$ as it remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test log-likelihood.

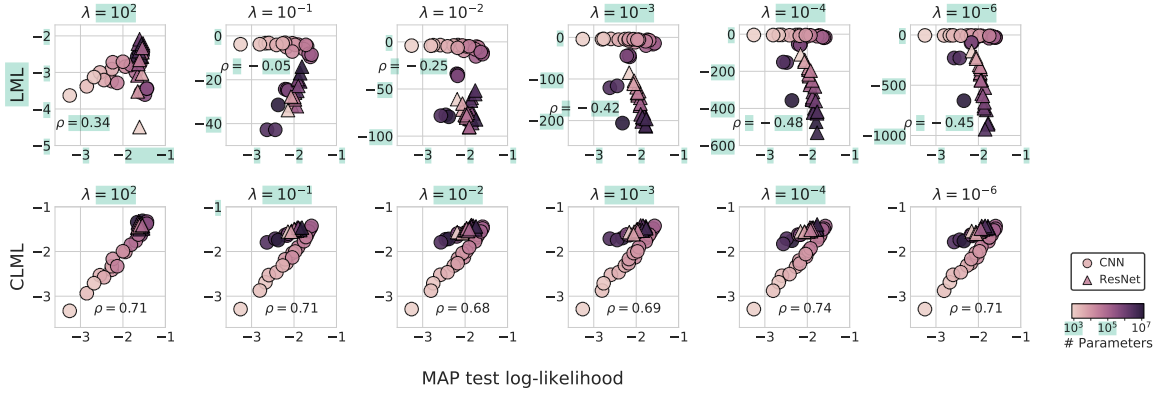


Figure 15. **Neural architecture search for CIFAR-100.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-6}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the MAP test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the MAP test log-likelihood and the conditional log marginal likelihood (CLML). The LML correlates positively with the MAP test log-likelihood for high values of the prior precision, and negatively for low values of the prior precision (vague priors). We can that the correlation shift occurs around $\lambda = 10^{-1}$ as the correlation remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test log-likelihood.

I. Extended Gaussian Process Results

GPs: RBF kernel. In Figure 22(a) we illustrate the bias of LML towards underfitting. We follow the experiment presented in Wilson et al. (2015), and generate 100 datasets from an RBF Gaussian process prior with a lengthscale of $l = 4$. The datapoints are located at positions $\{1, \ldots, 150\}$, the output scale is 1, and the observation noise is 0.2. For each dataset and each $n \in \{1, \ldots, 150\}$, we fit a new GP model to the the first n datapoints of the dataset: we maximize the LML or CLML with respect to the lengthscale of the RBF kernel, using the ground truth value $l = 4$ as the initialization. We plot the learned lengthscales averaged over the datasets as a function of n in 22(a). This experiment illustrates a unique quality of marginal likelihood that distinguishes it from conventional maximum likelihood training: while low lengthscales would lead to a better fit of the training data, marginal likelihood has a *bias towards underfitting* in the data space. Indeed, LML consistently selects lengthscales that are larger than the the lengthscale that was used to generate the data, especially for

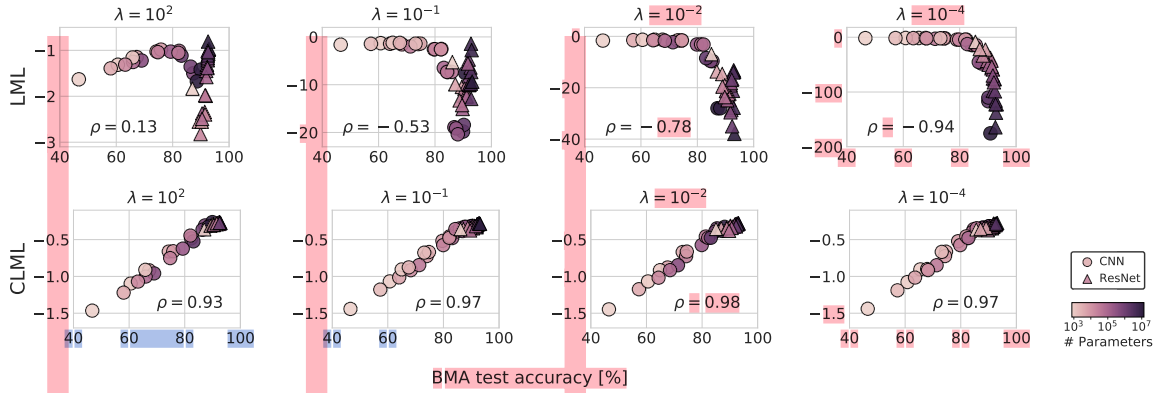


Figure 16. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test accuracy and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test accuracy and the conditional log marginal likelihood (CLML). The LML correlates positively with the BMA test accuracy for high values of the prior precision, and negatively for low values of the prior precision (vague priors). The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test accuracy.

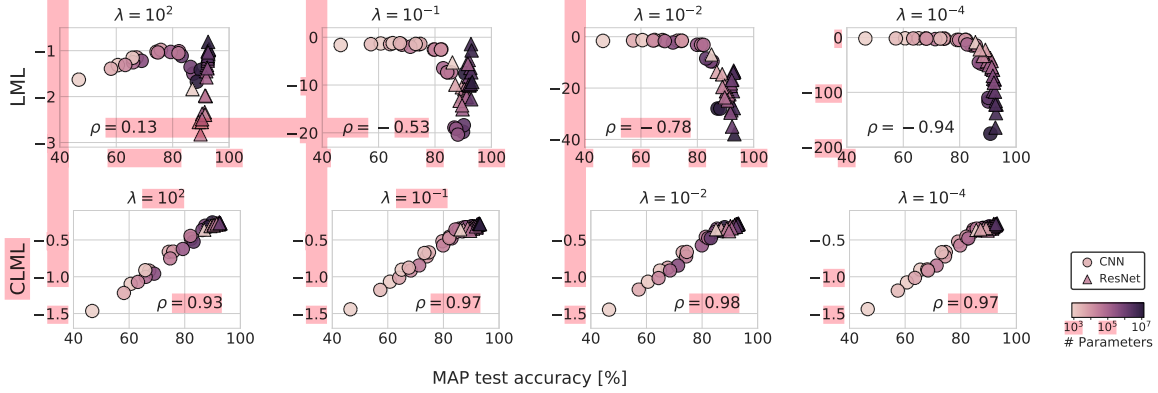


Figure 17. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the maximum-a-posterior (MAP) test accuracy and the LML. **(Top row)**: Correlation between the MAP test accuracy and the CLML. The LML correlates positively with the MAP test accuracy for high values of the prior precision, and negatively for low values of the prior precision, which correspond to vague priors. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test accuracy.

(RQ) kernel (see Rasmussen and Nickisch (2010)): $k_{RQ}(x_1, x_2) = \left(1 + \frac{\|x_1 - x_2\|^2}{2\alpha l^2}\right)^{-\alpha}$. The hyperparameters are the lengthscale l and α; lower values of α correspond to higher prior correlations, while as $\alpha \to \infty$ the kernel approaches the RBF kernel with lengthscale l.

We generate the data from a GP with an RQ kernel with hyperparameters $\hat{\alpha} = 0.05$, $\hat{l} = 0.5$, and observation noise standard deviation $\hat{\sigma} = 0.1$. The dataset is shown in Appendix Figure 22. We then evaluate the LML and CLML as a function of α and compare them to the true BMA predictive likelihood of test data. For this experiment we set the lengthscale $l = \hat{l}$ to its ground truth value, and we consider two values of the observation noise standard deviation: ground truth $\sigma = \hat{\sigma} = 0.1$ and over-estimated noise $\sigma = 2 \cdot \hat{\sigma} = 0.2$. We show the results in Figure 21(b). For the ground-truth noise scale, both LML and CLML provide an adequate representation of the test likelihood, although they both peak at a lower α value than the test likelihood surface. However, for $\sigma = 0.2$ the marginal likelihood is completely misaligned with the test log-likelihood: LML peaks at $\alpha \approx 0$, and then sharply decreases, while test LL is the lowest near $\alpha = 0$, and increases with α. The CLML does a much better job of tracking the test LL curve.

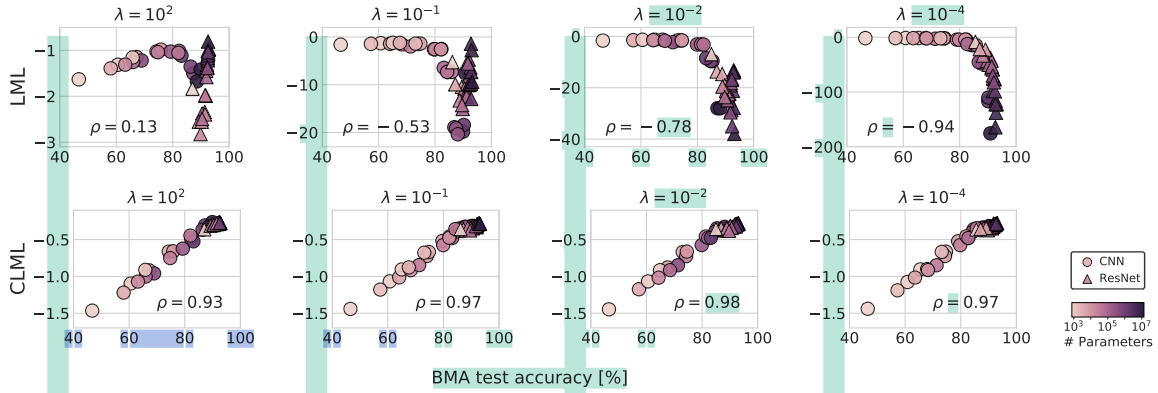


Figure 16. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test accuracy and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test accuracy and the conditional log marginal likelihood (CLML). The LML correlates positively with the BMA test accuracy for high values of the prior precision, and negatively for low values of the prior precision (vague priors). The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test accuracy.

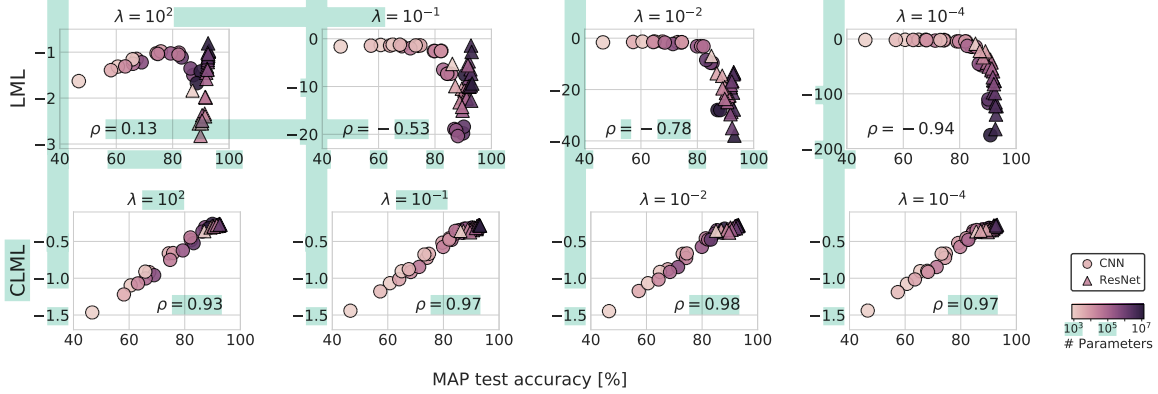


Figure 17. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the maximum-a-posterior (MAP) test accuracy and the LML. **(Top row)**: Correlation between the MAP test accuracy and the CLML. The LML correlates positively with the MAP test accuracy for high values of the prior precision, and negatively for low values of the prior precision, which correspond to vague priors. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test accuracy.

small n. We note that CLML does not remove this bias, and provides a very similar curve.

GPs: RQ kernel. Above, we have seen how marginal likelihood can over-estimate the lengthscale of an RBF kernel leading to underfitting in data space. Here, we construct a more extreme example of this behavior using the rational quadratic (RQ) kernel (see Rasmussen and Nickisch (2010)): $k_{RQ}(x_1, x_2) = \left(1 + \frac{\|x_1 - x_2\|^2}{2\alpha l^2}\right)^{-\alpha}$. The hyperparameters are the lengthscale l and α; lower values of α correspond to higher prior correlations, while as $\alpha \to \infty$ the kernel approaches the RBF kernel with lengthscale l.

We generate the data from a GP with an RQ kernel with hyperparameters $\hat{\alpha} = 0.05$, $\hat{l} = 0.5$, and observation noise standard deviation $\hat{\sigma} = 0.1$. The dataset is shown in Appendix Figure 23. We then evaluate the LML and CLML as a function of α and compare them to the true BMA predictive likelihood of test data. For this experiment we set the lengthscale $l = \hat{l}$ to its ground truth value, and we consider two values of the observation noise standard deviation: ground truth $\sigma = \hat{\sigma} = 0.1$ and over-estimated noise $\sigma = 2 \cdot \hat{\sigma} = 0.2$. We show the results in Figure 22(b). For the ground-truth noise scale, both LML and CLML provide an adequate representation of the test likelihood, although they both peak at a lower α value than the

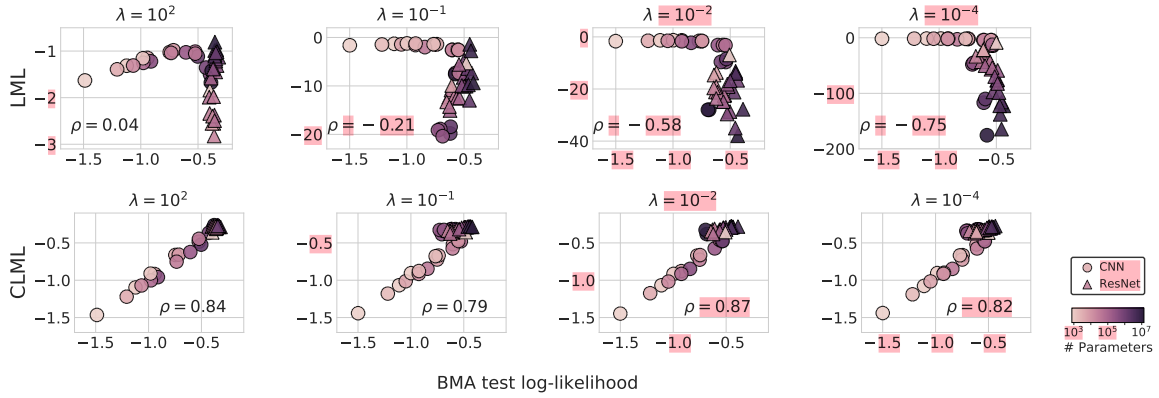


Figure 18. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision$\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test log-likelihood and the conditional log marginal likelihood (CLML). The LML almost does not correlate with the BMA test log-likelihood for high values of the prior precision, but shows a negative correlation for low values of the prior precision (vague priors). The correlation shifts around $\lambda = 10^{-1}$ as it remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test log-likelihood.

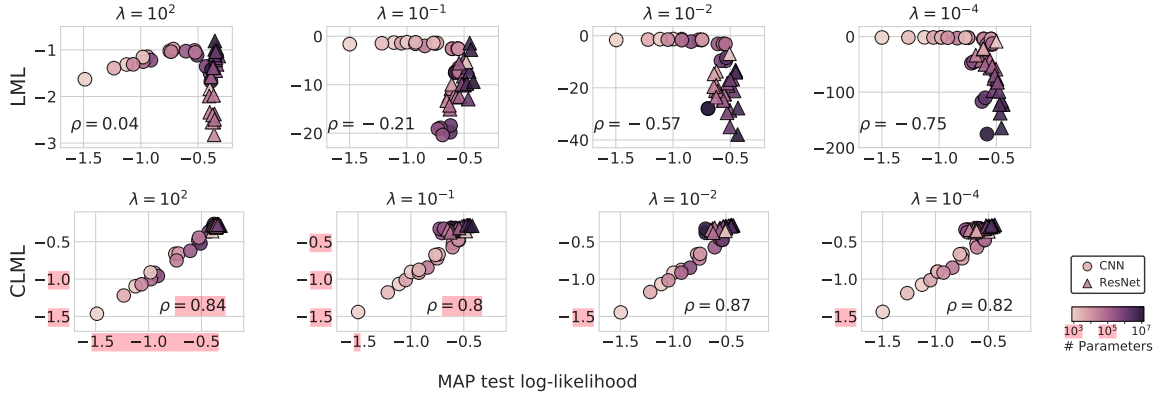


Figure 19. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the MAP test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the MAP test log-likelihood and the conditional log marginal likelihood (CLML). The LML almost does not correlate with the MAP test log-likelihood for high values of the prior precision, but shows a negative correlation for low values of the prior precision (vague priors). We can that the correlation shift occurs around $\lambda = 10^{-1}$ as the correlation remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test log-likelihood.

In Figure 22 (a), (b) we show the fit of the model with over-estimated observation noise $\sigma = 0.2$ for the α parameter chosen by maximizing the marginal likelihood and CML respectively. For the CML, we condition on $m = 45$ datapoints (the training dataset size is $n = 50$), and we average the results over 20 random orderings of the data.

In panel (c) of Figure 22 we show the learning curve averaged over 100 random orderings of the data. While for large n the $\alpha = 0.3$ model generalizes better, the small-n terms in the marginal likelihood decomposition dominate, so that marginal likelihood prefers the simpler $\alpha = 0.001$ model.

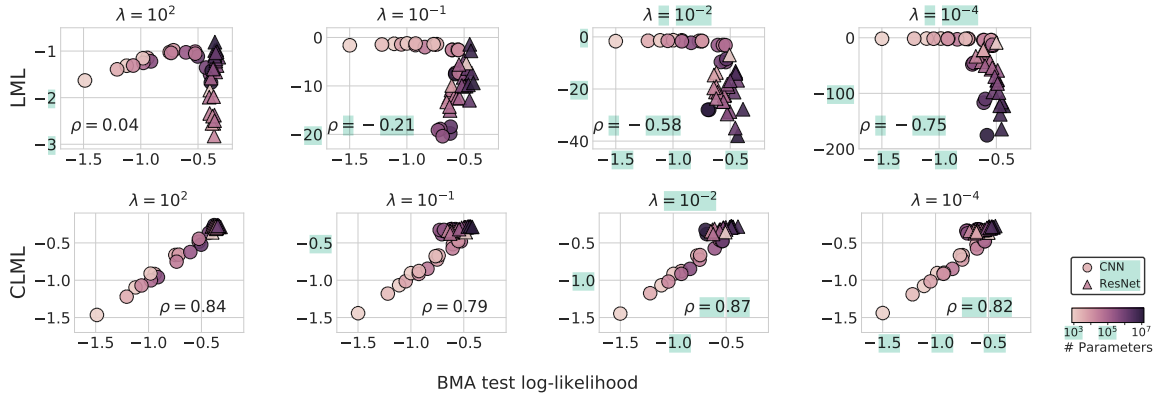


Figure 18. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision$\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the BMA test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the BMA test log-likelihood and the conditional log marginal likelihood (CLML). The LML almost does not correlate with the BMA test log-likelihood for high values of the prior precision, but shows a negative correlation for low values of the prior precision (vague priors). The correlation shifts around $\lambda = 10^{-1}$ as it remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the BMA test log-likelihood.

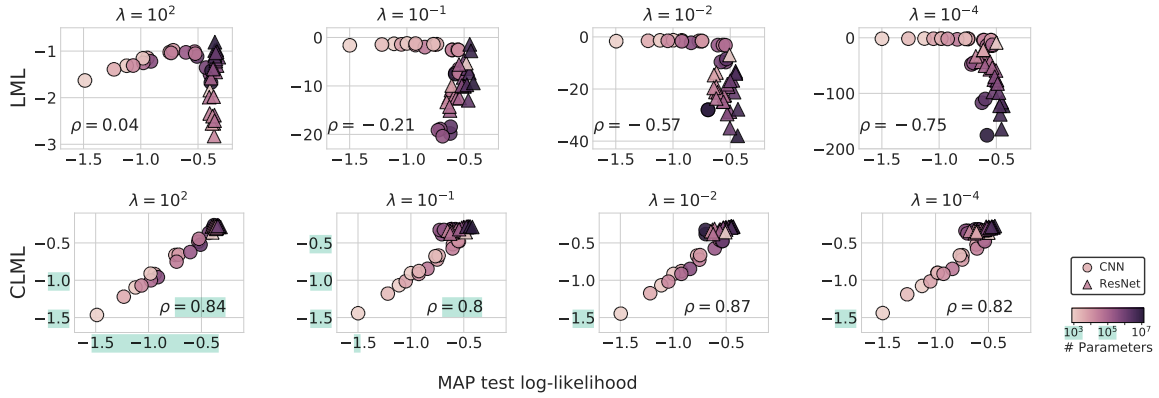
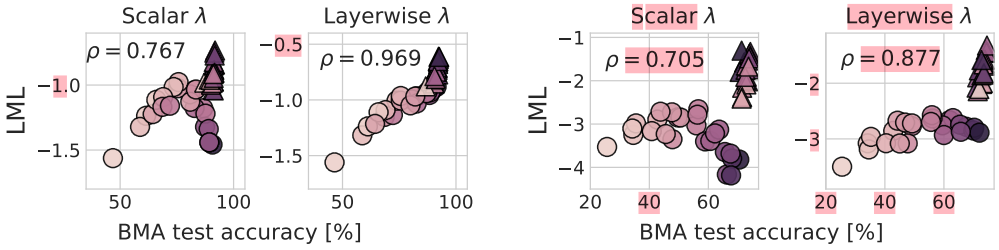


Figure 19. **Neural architecture search for CIFAR-10.** Visualization of the correlation between the model rankings according to different metrics for fixed prior precision $\lambda \in \{10^2, 10^{-1}, 10^{-2}, 10^{-4}\}$. We report the Spearman's correlation coefficient ρ in each figure. **(Top row)**: Correlation between the MAP test log-likelihood and the log marginal likelihood (LML). **(Top row)**: Correlation between the MAP test log-likelihood and the conditional log marginal likelihood (CLML). The LML almost does not correlate with the MAP test log-likelihood for high values of the prior precision, but shows a negative correlation for low values of the prior precision (vague priors). We can that the correlation shift occurs around $\lambda = 10^{-1}$ as the correlation remains positive for ResNets but becomes negative for CNNs. The CLML on the other hand is less sensitive to the value of the prior precision and consistently achieves a positive correlation with the MAP test log-likelihood.

test likelihood surface. However, for $\sigma = 0.2$ the marginal likelihood is completely misaligned with the test log-likelihood: LML peaks at $\alpha \approx 0$, and then sharply decreases, while test LL is the lowest near $\alpha = 0$, and increases with α. The CLML does a much better job of tracking the test LL curve.

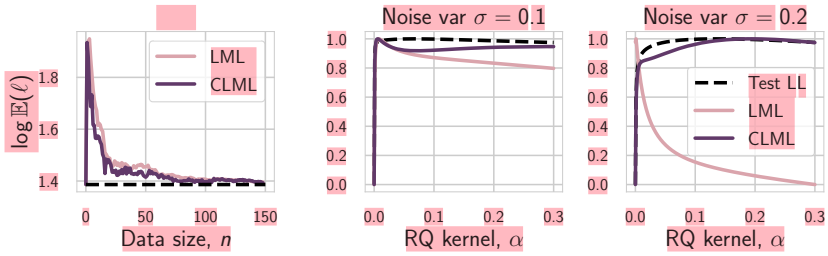
In Figure 23 (a), (b) we show the fit of the model with over-estimated observation noise $\sigma = 0.2$ for the α parameter chosen by maximizing the marginal likelihood and CML respectively. For the CML, we condition on $m = 45$ datapoints (the training dataset size is $n = 50$), and we average the results over 20 random orderings of the data.

In panel (c) of Figure 23 we show the learning curve averaged over 100 random orderings of the data. While for large n the $\alpha = 0.3$ model generalizes better, the small-n terms in the marginal likelihood decomposition dominate, so that marginal likelihood prefers the simpler $\alpha = 0.001$ model.



(a) CIFAR-10 (b) CIFAR-100

Figure 20. **Neural network hyperparameter optimization for CIFAR-10 and CIFAR-100.** Correlation between the log marginal likelihood (LML) and the BMA test accuracy for **(Left)** optimized global prior precision, and **(Right)** optimized layerwise prior precision for CIFAR-10. We report the Spearman's correlation coefficient ρ in each figure. We observe that the layerwise optimization further improves the correlation between the LML and the BMA test accuracy.



(a) Underfitting bias (b) Underfitting with the RQ kernel

Figure 21. **LML for hyper-parameter tuning in Gaussian Processes. (a):** The log-lengthscale learned by LML and CLML in a GP regression model averaged over 100 datasets generated from a GP model with a lengthscale of 4. Unlike the train likelihood, LML has a bias towards underfitting, consistently overestimating the lengthscale, particularly for small $n < 20$. **(b):** Test log-likelihood, LML and CLML as a function of the α hyper-parameter in the rational quadratic kernel. While in panel with observation noise $\sigma = 0.1$ the LML roughly captures the shape of the test likelihood curve, with $\sigma = 0.2$ the two curves are completely misaligned.

J. Deep Kernel Learning Details

J.1. UCI Regression

For the UCI regression datasets we use a DKL model with a fully-connected ReLU architecture of $[\mathcal{D}, 50, 50, 2]$, where $\mathcal{D}$ is the dimensionality of the data, and train using random subsets of the full UCI datasets ranging in size from 100 to 700 training points. We use the Bayesian Benchmarks library[1] to obtain the datasets, with a modification to ensure test data are not included in the normalization statistics. Models are trained using the closed form LML and CLML forms known for Gaussian process regression.

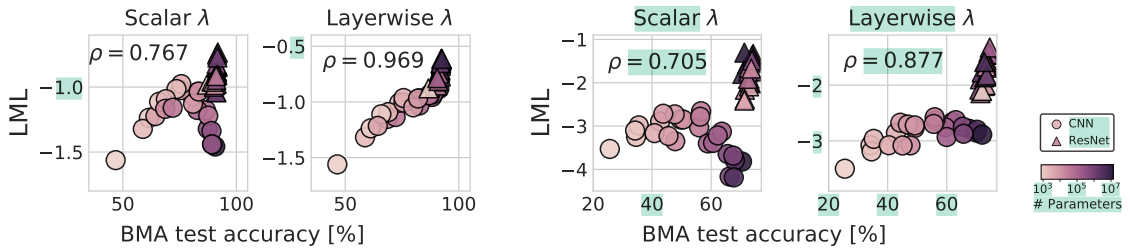
In figure 23 we show how the RMSE (normalized by dataset) varies for CLML optimization as a function of the fraction of data used used to condition the conditional marginal likelihood. As a general trend, the performance of CLML optimization increases as we use a larger fraction of the available data to condition on and a smaller fraction to compute the likelihood.

In Figure 24 we show the negative log likelihoods (normalized by dataset) of the $N = 100$ models on the UCI regression problems. While there is some variance in the relative gap in performance between CLML and LML optimization, in all cases we see that for very restricted train set sizes CLML not only produces more accurate predictions, but is more performant in terms of NLL.

J.2. DKT Transfer Learning

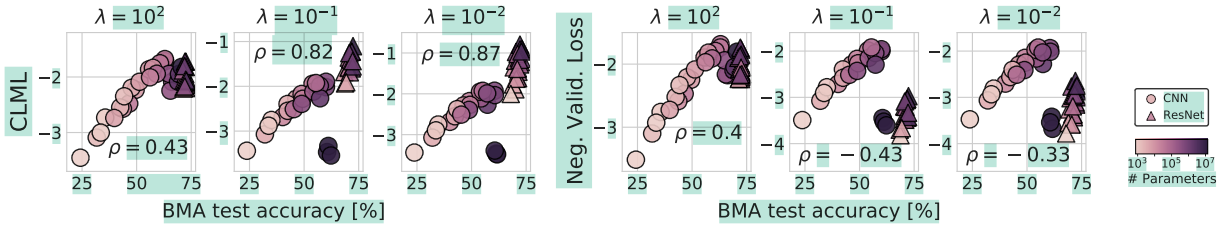
Tables 3 and 4 give the numerical results accompanying Figure 7. From these tables we see that CLML optimization with the same model configuration consistently outperforms LML optimization, and in both experiments leads to the highest

[1] https://github.com/hughsalimbeni/bayesian_benchmarks



(a) CIFAR-10 (b) CIFAR-100

Figure 20. **Neural network hyperparameter optimization for CIFAR-10 and CIFAR-100.** Correlation between the log marginal likelihood (LML) and the BMA test accuracy for **(Left)** optimized global prior precision, and **(Right)** optimized layerwise prior precision for CIFAR-10. We report the Spearman's correlation coefficient ρ in each figure. We observe that the layerwise optimization further improves the correlation between the LML and the BMA test accuracy.



(a) CLML vs BMA accuracy (b) Negative of the validation loss vs BMA accuracy

Figure 21. **Neural hyperparameter optimization for CIFAR-100.** The correlation (Spearman's ρ) between the model rankings and generalization. For panels **(a), (b)**, we consider a fixed prior precision $\lambda = 10^2$, 10^{-1}, and 10^{-3}. **(a)**: Correlation between the BMA test accuracy and the CLML. **(a)** Correlation between the BMA test accuracy and the negative of the validation loss. Similarly to the LML and the CLML, the negative of the validation loss correlates positively with the test accuracy for a high prior precision. For lower values of the prior precision, the negative validation loss correlates positively with the BMA test accuracy for CNNs but not for ResNets, therefore correlating negatively overall with the BMA test accuracy for CIFAR-100.

J. Deep Kernel Learning Details

J.1. UCI Regression

For the UCI regression datasets we use a DKL model with a fully-connected ReLU architecture of $[\mathcal{D}, 50, 50, 2]$, where $\mathcal{D}$ is the dimensionality of the data, and train using random subsets of the full UCI datasets ranging in size from 100 to 700 training points. We use the Bayesian Benchmarks library[1] to obtain the datasets, with a modification to ensure test data are not included in the normalization statistics. Models are trained using the closed form LML and CLML forms known for Gaussian process regression.

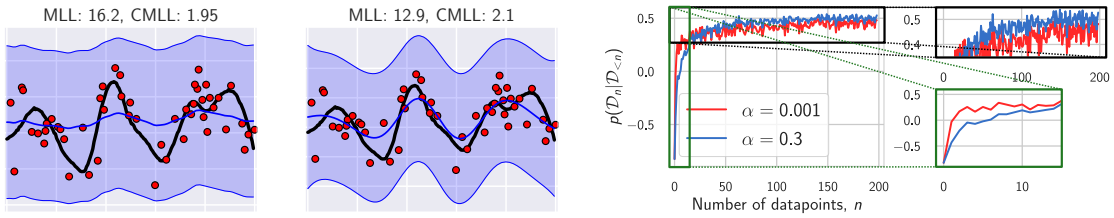
In figure 24 we show how the RMSE (normalized by dataset) varies for CLML optimization as a function of the fraction of data used used to condition the conditional marginal likelihood. As a general trend, the performance of CLML optimization increases as we use a larger fraction of the available data to condition on and a smaller fraction to compute the likelihood.

In Figure 25 we show the negative log likelihoods (normalized by dataset) of the $N = 100$ models on the UCI regression problems. While there is some variance in the relative gap in performance between CLML and LML optimization, in all cases we see that for very restricted train set sizes CLML not only produces more accurate predictions, but is more performant in terms of NLL.

J.2. DKT Transfer Learning

Tables 3 and 4 give the numerical results accompanying Figure 7. From these tables we see that CLML optimization with the same model configuration consistently outperforms LML optimization, and in both experiments leads to the highest performing model. For full experimental details see Patacchiola et al. (2020).

[1] https://github.com/hughsalimbeni/bayesian_benchmarks



(a) LML-optimized α (b) CLML-optimized α (c) Learning curves

Figure 22. **Gaussian process: RQ kernel (details).** The data fit and learning curves for the GP models with the RQ kernel with different α parameters. **(a)**: The α parameter maximizing the marginal likelihood; **(b)**: the α parameter maximizing the CML. In each panel, the red dots show the training data, the black line shows the true latent function, the blue line shows the predictive mean, and the shaded region shows the 2σ-region under the predictive distribution. **(c)**: learning curves for small α providing the best marginal likelihood, and high α providing the best generalization. While the larger α value generalizes better when the training set size is $n \geq 50$, marginal likelihood prefers small α values, as the model with a small α generalizes better on small datasets with $n \leq 15$.

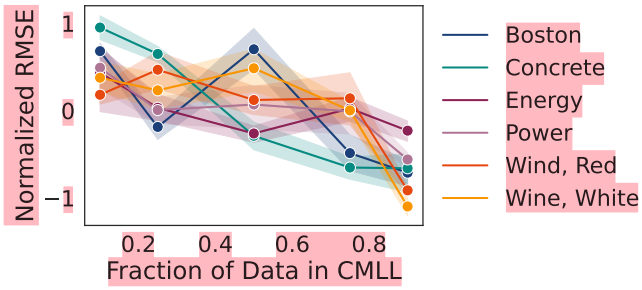


Figure 23. Effect of the parameter m in the performance of DKL on several UCI regression datasets. The general trend is such that setting m to be a larger, so that the CLML is conditioned on a larger fraction of the available training data, leads to better performance.

performing model. For full experimental details see Patacchiola et al. (2020).

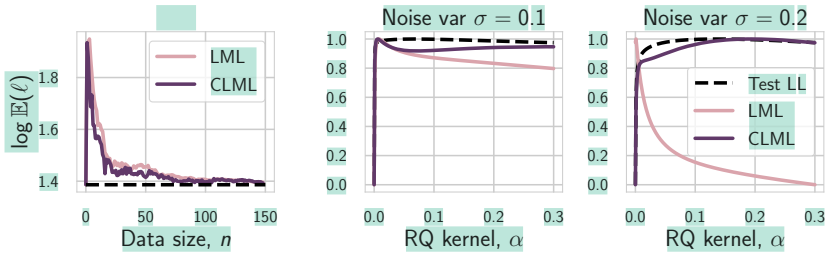
K. Choice of m for the Conditional Marginal Likelihood

The hyperparameter m — the number of datapoints that we condition on in CLML — has an important effect on the conditional marginal likelihood. Indeed, if we set $m = 0$, we recover the marginal likelihood. Setting $m = n - 1$, we recover leave-one-out cross-validation likelihood for the BMA model, assuming we average the CLML over all possible orderings of the data. Generally, we find that CLML works best for relatively large values of m. However, setting $m << n$ (for example, in the architecture search experiments in Section 6) allows us to estimate CLML without averaging over multiple orderings.

In Figure 23 we show the effect of m on the final RMSE for Deep Kernel Learning models trained with CLML. We find that larger values of m lead to better performance, but the results are relatively stable with respect to m.

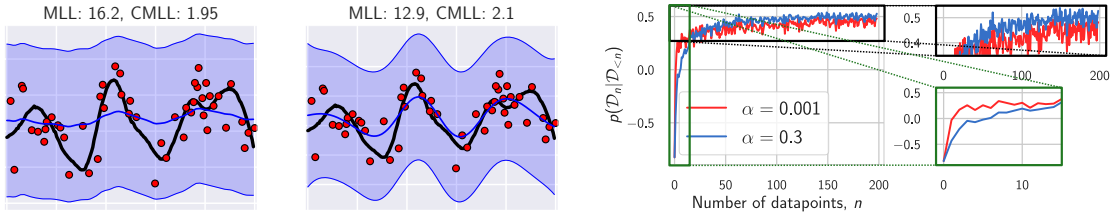
Method	Model	MSE
CLML	DKT + RBF	$\mathbf{0.066 \pm 0.08}$
	DKT + Spectral	0.076 ± 0.05
LML	DKT + RBF	0.12 ± 0.04
	DKT + Spectral	0.10 ± 0.01

Table 3. CLML and LML optimization of deep kernel transfer models on the QMUL head pose trajectory task of Patacchiola et al. (2020). In this limited data regime the focus on test performance of CLML leads to stronger performance.



(a) Underfitting bias (b) Underfitting with the RQ kernel

Figure 22. **LML for hyper-parameter tuning in Gaussian Processes. (a):** The log-lengthscale learned by LML and CLML in a GP regression model averaged over 100 datasets generated from a GP model with a lengthscale of 4. Unlike the train likelihood, LML has a bias towards underfitting, consistently overestimating the lengthscale, particularly for small $n < 20$. **(b):** Test log-likelihood, LML and CLML as a function of the α hyper-parameter in the rational quadratic kernel. While in panel with observation noise $\sigma = 0.1$ the LML roughly captures the shape of the test likelihood curve, with $\sigma = 0.2$ the two curves are completely misaligned.



(a) LML-optimized α (b) CLML-optimized α (c) Learning curves

Figure 23. **Gaussian process: RQ kernel (details).** The data fit and learning curves for the GP models with the RQ kernel with different α parameters. **(a)**: The α parameter maximizing the marginal likelihood; **(b)**: the α parameter maximizing the CML. In each panel, the red dots show the training data, the black line shows the true latent function, the blue line shows the predictive mean, and the shaded region shows the 2σ-region under the predictive distribution. **(c)**: learning curves for small α providing the best marginal likelihood, and high α providing the best generalization. While the larger α value generalizes better when the training set size is $n \geq 50$, marginal likelihood prefers small α values, as the model with a small α generalizes better on small datasets with $n \leq 15$.

K. Choice of m for the Conditional Marginal Likelihood

The hyperparameter m — the number of datapoints that we condition on in CLML — has an important effect on the conditional marginal likelihood. Indeed, if we set $m = 0$, we recover the marginal likelihood. Setting $m = n - 1$, we recover leave-one-out cross-validation likelihood for the BMA model, assuming we average the CLML over all possible orderings of the data. Generally, we find that CLML works best for relatively large values of m. However, setting $m << n$ (for example, in the architecture search experiments in Section 6) allows us to estimate CLML without averaging over multiple orderings.

In Figure 24 we show the effect of m on the final RMSE for Deep Kernel Learning models trained with CLML. We find that larger values of m lead to better performance, but the results are relatively stable with respect to m.

Method	Model	MSE
CLML	DKT + RBF	$\mathbf{0.066 \pm 0.08}$
	DKT + Spectral	0.076 ± 0.05
LML	DKT + RBF	0.12 ± 0.04
	DKT + Spectral	0.10 ± 0.01

Table 3. CLML and LML optimization of deep kernel transfer models on the QMUL head pose trajectory task of Patacchiola et al. (2020). In this limited data regime the focus on test performance of CLML leads to stronger performance.

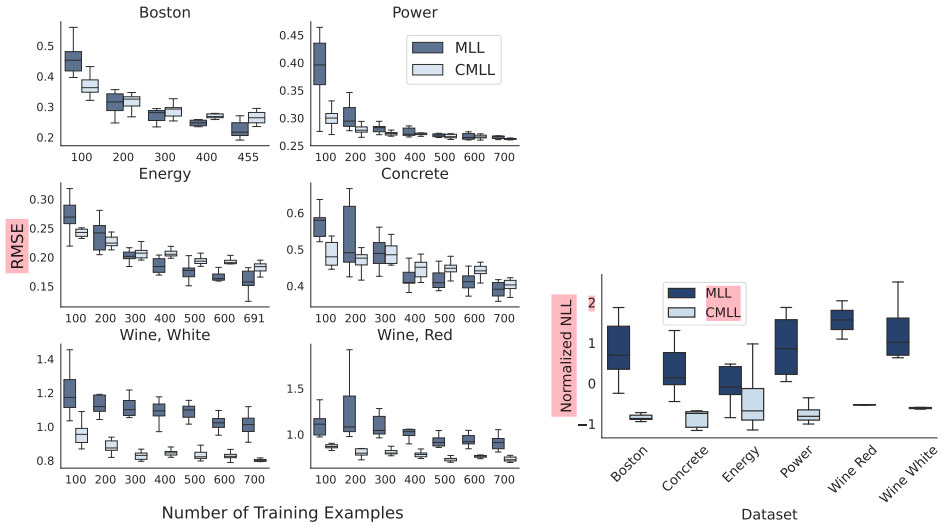


Figure 24. **Left:** RMSE and **right:** NLL values normalized by dataset for DKL models trained on UCI regression tasks when trained with $N = 300$ datapoints over 10 independent initializations. For this training set size not only does CLML optimization lead to better accuracy on the test set, but also to better test likelihoods in limited data settings.

Method	Model	Accuracy
CLML	DKT + CosSim	75.34 ± 0.35
	DKT + BNCosSim	$\mathbf{76.03 \pm 0.57}$
	DKT + Linear	75.64 ± 0.38
LML	DKT + CosSim	73.06 ± 2.36
	DKT + BNCosSim	75.06 ± 1.10
	DKT + Linear	75.97 ± 0.70

Table 4. CLML and LML optimization of deep kernel transfer models on a transfer learning task in which the training data is from the Omniglot dataset, and the test data from the EMNIST dataset. In this transfer learning setting we should be more focused on the test performance, as opposed to the alignment of our model with the training data, which is a focus more closely aligned with the biases of CLML than LML optimization.

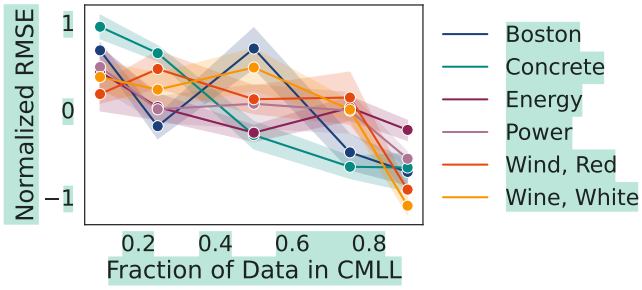


Figure 24. Effect of the parameter m in the performance of DKL on several UCI regression datasets. The general trend is such that setting m to be a larger, so that the CLML is conditioned on a larger fraction of the available training data, leads to better performance.

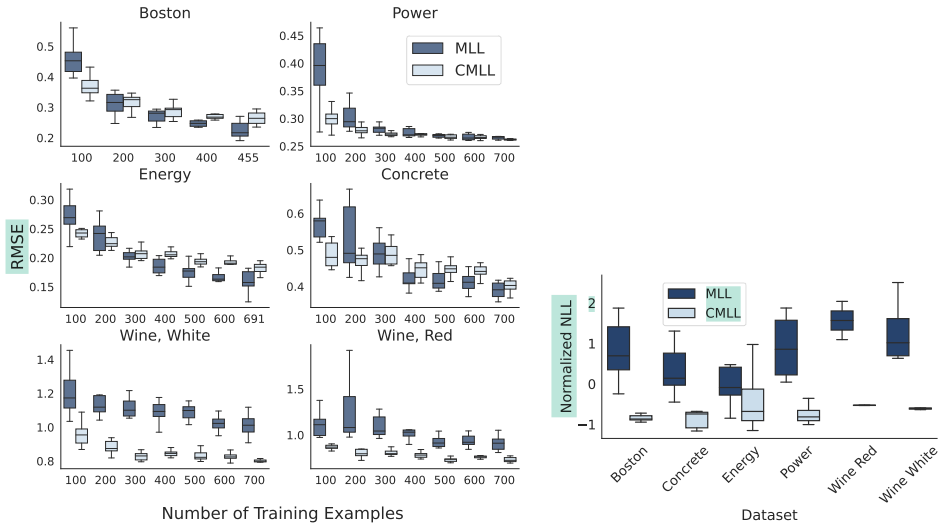


Figure 25. **Left:** RMSE and **right:** NLL values normalized by dataset for DKL models trained on UCI regression tasks when trained with $N = 300$ datapoints over 10 independent initializations. For this training set size not only does CLML optimization lead to better accuracy on the test set, but also to better test likelihoods in limited data settings.

Method	Model	Accuracy
CLML	DKT + CosSim	75.34 ± 0.35
	DKT + BNCosSim	$\mathbf{76.03 \pm 0.57}$
	DKT + Linear	75.64 ± 0.38
LML	DKT + CosSim	73.06 ± 2.36
	DKT + BNCosSim	75.06 ± 1.10
	DKT + Linear	75.97 ± 0.70

Table 4. CLML and LML optimization of deep kernel transfer models on a transfer learning task in which the training data is from the Omniglot dataset, and the test data from the EMNIST dataset. In this transfer learning setting we should be more focused on the test performance, as opposed to the alignment of our model with the training data, which is a focus more closely aligned with the biases of CLML than LML optimization.