Hyperparams sensitivity

...

We've designed a comprehensive sensitivity analysis to understand how different hyperparameter choices impact the performance of the Laplace Approximation, specifically its ability to produce reliable uncertainty estimates. Our experiment is built around the **laplace** library introduced in the paper and uses the paper's own findings to guide our choices, especially in the context of limited computational resources.

Architectures: WideLeNet and ResNet18

Our investigation begins with the models themselves. We selected two distinct and well-established architectures: **WideLeNet** and **ResNet18**.

- WideLeNet is a variant of the classic LeNet-5, a foundational convolutional neural network for image recognition. By using a "wider" version, we work with a more modern and capable architecture than the original, better reflecting contemporary practices.
- ResNet18 is a deep residual network, an architecture that has become a standard-bearer in computer vision. The "Laplace Redux" paper itself uses ResNet architectures for its more complex benchmarks, such as the WILDS dataset, making our choice a direct parallel.

Using two different architectures is a crucial design choice. It allows us to check if our findings are generalizable or merely an artifact of a single model's structure. For our experiment, we trained both models on the MNIST dataset. This is a critical first step, as the post-hoc Laplace Approximation is designed to be applied to a pretrained model. We first find a Maximum a Posteriori (MAP) estimate through standard training, and then the LA builds a Gaussian approximation to the posterior around that point.

Data: In-Distribution vs. Out-Of-Distribution

A model's uncertainty is most tested when it encounters data it wasn't trained on. Therefore, evaluating on both in-distribution (ID) and out-of-distribution (OOD) data is essential.

- In-Distribution (ID): MNIST Test Set.
  Our ID data is the standard MNIST test set. This gives us a baseline performance measure in the ideal scenario where the test data comes from the same distribution as the training data.
- Out-of-Distribution (OOD): Rotated-MNIST (R-MNIST).
  For our OOD challenge, we chose Rotated-MNIST. This is a perfect choice for a sensitivity analysis and one used directly in the "Laplace Redux" paper for evaluating calibration under dataset shift (see Figure 4). Rather than using a completely different dataset, R-MNIST introduces a controlled and continuous "shift intensity". Our script evaluates the models on images rotated by a range of angles: 5, 15, 30, 45, 60, 90, 120, 160, and 180 degrees. This allows us to observe not just if performance degrades on OOD data, but how gracefully it does so as the data shifts further and further from the original distribution.

Metrics and experimental design

Simple accuracy is not enough to evaluate a Bayesian model, as a model can be accurate but dangerously overconfident. Our experiment focuses on metrics that directly measure the quality of uncertainty estimates, such as Negative Log-Likelihood (**NLL**) and Expected Calibration Error (**ECE**). NLL penalizes a model for being both incorrect and confident , while ECE directly measures if a model's confidence is reliable.

The central goal is to test sensitivity to hyperparameters. Given our limited resources, we made a strategic decision to focus on the last-layer LA by setting `subset_of_weights='last_layer'`. The paper repeatedly highlights the last-layer LA as a powerful and efficient variant. It is described as "cost-effective yet compelling" , significantly cheaper than applying the LA to all weights, and is the recommended default in the laplace library. By only treating the final layer's weights probabilistically, we dramatically reduce the size of the Hessian matrix that needs to be computed and inverted, making a wide hyperparameter sweep computationally feasible.

With that fixed, we created a grid to explore the most influential remaining hyperparameters:

- **hessian_structure**:
  We evaluate two options: `diag` and `kron`. The `diag` (diagonal) approximation is the most lightweight, assuming independence between all weights. In contrast, `kron` (Kronecker-factored / KFAC) is more expressive, modeling correlations between weights within the same layer, and is noted to provide a good trade-off between expressiveness and speed. Our experiment will directly reveal if the added complexity of KFAC provides a tangible benefit over the simpler diagonal approximation in the last-layer context.

- **prior_precision**:
  This is a fundamental Bayesian hyperparameter, corresponding to the inverse variance of the prior distribution over the weights. A low precision implies a broad prior (less regularization), while high precision implies a narrow, restrictive prior. We are testing a wide logarithmic scale of values (1e-6, 1e-4, 1e-2, 1.0, 10.0, 100.0) to see how strongly this choice affects the final calibration and OOD performance.

- **link_approx**:
  For classification, the predictive distribution is intractable and requires approximation. We compare the `probit` and `bridge` approximations. The paper suggests that the probit approximation often provides the best results , while the Laplace bridge is another method that yields a distribution over the integral solutions. Our experiment will serve to validate this recommendation in our specific setup.

- **temperature**:
  We also include temperature scaling, a common technique for post-hoc calibration. Its inclusion allows us to see how this simple method interacts with the more complex machinery of the Laplace approximation.

To ensure our results are reliable, we've used multiple random seeds ([0, 42, 123]) for training our base models. Neural network training is a stochastic process, and a single run can be an outlier. By running our entire experiment across models trained with different seeds and then averaging the results, we ensure our conclusions are robust and not due to a fluke of random initialization.

Hyperparameter analysis: prior precision

In our exploration, we can first zoom in on the most crucial hyperparameter: `prior_precision`. This single value, which controls the strength of our Bayesian regularization, has a profound and sometimes surprising impact on model calibration and robustness.

> Note: when the plots are titled Aggregated Result, results where averaged between the two different models (WideLeNet and ResNet18), as their behaviour was almost identical, thus not interesting to show both.

1. **Regularization improves In-Distribution calibration**

Our first key finding relates to the solid black line present in all the plots (here we show just the most insightful one to improve readibility), representing performance on in-distribution (ID) data. In almost every configuration, as we increase `prior_precision` from very small values, the performance on ID data either stays excellent or gets significantly better.

ECE-kron-probit-pp

This is most evident in the Aggregated ECE (Expected Calibration Error) plot for the kron/probit context. The ID ECE starts very high (indicating poor calibration) and then plummets to near-zero for `prior_precision` values of 10.0 or higher. This shows that a higher prior_precision acts as a powerful regularizer, penalizing overly complex solutions and forcing the model into a state that is much better calibrated on data it expects to see.

2. **The critical trade-off for Out-of-Distribution robustness**

While high `prior_precision` is good for ID data, it comes at a cost. This brings us to the most important insight from our entire experiment, which is best illustrated by the Aggregated NLL (Negative Log-Likelihood) plot for the kron/probit context.

NLL-kron-probit-pp

This plot reveals a crucial trade-off. Let's trace the dashed lines, which represent out-of-distribution (OOD) data with increasing rotation:

- At low precision (e.g., 10^−6), both ID and OOD NLL are high. The model is poorly regularized.
- At medium precision (around 1.0), the NLL for both ID and most OOD data reaches a minimum. This is the "sweet spot" where the model is well-regularized but still flexible.
- At high precision (e.g., 10^2), the ID NLL remains low, but the OOD NLL skyrockets.

This "U-shaped" pattern for OOD performance is a classic finding. It means that while strong regularization helps the model perfect its performance on the training distribution, it also makes the model rigid and overconfident. When faced with shifted data, this over-regularized model fails catastrophically because its learned weights are too constrained to adapt. Choosing a `prior_precision` is therefore not about maximizing ID performance, but about finding the optimal balance between ID performance and OOD robustness.

3. **The full picture depends on the configuration**

Is this U-shaped trade-off universal? Not necessarily. The final insight is that the behavior of prior_precision is deeply connected to the other hyperparameters, namely hessian_structure and link_approx.

To see this, we should compare two ECE plots side-by-side.

ECE vs. Prior Precision (kron/probit)	ECE vs. Prior Precision (diag/bridge)
kron/probit	diag/bridge

On the left (kron/probit), we see the dramatic U-shaped behavior for higly shifted OOD data. On the right (diag/bridge), the pattern is different. While the metrics still improve with higher precision, the OOD error for high-precision values tends to flatten out rather than sharply increasing.

This tells us that the combination of a more expressive Hessian (kron) and a probit link function creates a model that is highly sensitive and dynamic in its response to regularization. In contrast, the simpler diag Hessian and bridge link function lead to a more stable, perhaps less optimal, but more robust behavior against extreme regularization.

The Curious Case of the Insensitive Hyperparameter

In our experiment, one of the most valuable things we can uncover is not just how much a hyperparameter matters, but when it matters. In our exploratory plots, we noticed an interesting edge case: for the specific configuration using a Kronecker-factored Hessian (`hessian_structure='kron'`) and the Laplace Bridge predictive approximation (`link_approx='bridge'`), the choice of prior_precision appears to have virtually no effect on the model's final performance, be it NLL, ECE, or Brier score.

flat-pp

The key to this mystery lies in understanding the distinct roles of the different components. The `prior_precision` directly impacts the variance of the logit distribution that the model produces, it's our way of telling the model how confident it should be in its internal representations.

The second component, the `link_approx`, has the job of converting this distribution of logits into a final, concrete set of class probabilities.

The two link approximations we've tested, `probit` and `bridge`, do this in very different ways. The `probit` approximation has a direct and strong dependency on the variance of the logits. As we saw in other plots, changing the `prior_precision` dramatically changes the output of a probit-based model.

The Laplace Bridge, however, works by mapping the Gaussian distribution of logits to a Dirichlet distribution, which is a distribution over probability vectors. While the calculation for the parameters of this Dirichlet distribution technically depends on the logit variance (which is influenced by `prior_precision`), it seems that this dependency is extremely weak. The final output probabilities generated by the Laplace Bridge are overwhelmingly dominated by the mean of the logits, that is, the deterministic prediction from the original MAP model.

In essence, for this specific configuration, the nuance provided by the `prior_precision` gets lost in translation by the Laplace Bridge. The bridge approximation leans so heavily on the MAP model's initial guess that it effectively ignores the uncertainty information that the prior_precision was meant to control.

This is a perfect example of why sensitivity analysis is so critical. It reveals that certain combinations of methods can lead to unexpected behaviors, where the effect of one hyperparameter is effectively nullified by another.

When Models Diverge: The Importance of the Baseline

While we've seen several cases where our ResNet18 and WideLeNet models exhibit similar patterns, allowing us to summarize their behavior with an aggregated plot (like the one showed before), this is not always the case.

A prime example of this divergence is seen when using the (`hessian=diag`, `link=probit`) configuration.

comparison-pp

In the plot above, which shows the Expected Calibration Error (ECE), the two models tell different stories:

- For ResNet18 , the in-distribution (ID) ECE starts very high, around 0.4, indicating that the pre-trained model, before the full effect of the LA's regularization is applied, is poorly calibrated. As `prior_precision` increases, the ECE drops dramatically. Here, the Laplace Approximation has a powerful corrective effect, fixing a deficient baseline model.
- For WideLeNet, the ID ECE starts near-zero, indicating the pre-trained model is already very well-calibrated. As `prior_precision` changes, the LA's main role is to maintain this excellent calibration. The changes are far less dramatic because there was no fundamental problem to solve.

This divergence isn't a failure of the method but a reflection of the models themselves. The Laplace Approximation is a post-hoc method applied to an existing, pre-trained model. The characteristics of that baseline model matter:

- ResNet18, a deep and highly complex architecture, may be prone to overfitting or learning less robust features on a comparatively simple dataset like MNIST, resulting in poor initial calibration.
- WideLeNet, being a wider but shallower architecture, may have hit a sweet spot of capacity for this task, leading to a naturally well-calibrated solution.

Hence, LA is not a one-size-fits-all tool, its effect is context-dependent and heavily influenced by the quality of the initial MAP estimate. By keeping the results for this configuration separate, we highlight the LA's versatility, it can both fix poor models and preserve the quality of good ones.

Hessian Structure and Link Approximation

The "Laplace Redux" paper posits that the Kronecker-factored (`kron`) Hessian approximation offers a good trade-off between expressiveness and efficiency compared to a simple diagonal (`diag`) one. Our analysis reveals this is not a universal truth, but rather a complex reality where the optimal Hessian structure depends on a three-way interplay between regularization (`prior_precision`), the chosen link approximation (`link_approx`), and the severity of the distribution shift.

The unifying effect of strong regularization

Across all configurations, one finding is absolute: at high `prior_precision` (e.g., 100.0), the choice of Hessian structure becomes mostly irrelevant. As seen in the right-hand plots of our experiments, the performance lines for `diag` and `kron` converge. This is because a strong prior dominates the posterior estimation. The resulting Gaussian approximation of the posterior is so constrained by the narrow prior that the subtle details of the loss landscape's curvature, which the Hessian is meant to capture, become negligible. In this high-regularization regime, the simpler and more efficient `diag` approximation is sufficient. The truly interesting and complex behavior unfolds at low `prior_precision`, where the data likelihood has a much stronger influence on the posterior shape.

The Decisive Role of the Link Approximation at Low Regularization

At low `prior_precision` (e.g., 1e-06), the Hessian structure's impact is profound but is entirely mediated by the choice of `link_approx`.

1. **The bridge link**

When using the bridge link approximation at low prior_precision, we uncover a nuanced story with a clear divergence between the model's calibration (ECE) and its predictive likelihood (NLL).

bridge-hessian_ece

bridge-hessian_nll

When we evaluate the models based on their calibration, measured by the ECE, a clear and consistent pattern emerges. The kron Hessian approximation, which captures correlations between weights in the model's final layer, consistently yields a lower (better) ECE than the simpler diag approximation. This holds true for both the ResNet18 and WideLeNet architectures and across all tested degrees of distribution shift. This suggests the Laplace Bridge approximation is stable enough to leverage the richer correlation information from `kron` to produce more reliable and well-calibrated confidence estimates.

From a likelihood (NLL) perspective, the model choice seems to make a difference.

For the deeper ResNet18 model, a simpler posterior proves to be sufficient. The diag Hessian provides a better (lower) NLL for in-distribution data and for all but the most extreme distribution shifts. This suggests that for a deep architecture like ResNet18, the critical feature representations are likely learned and distributed across many layers, making the correlations between weights in just the final layer less significant for making accurate predictions. In this context, the additional correlation information captured by kron might act as noise, and ignoring these correlations becomes a beneficial simplification, leading to a more stable and effective posterior approximation for the primary prediction task.

In stark contrast, the shallower and wider WideLeNet model exhibits a fragile dependence on a correlated posterior. For in-distribution and low-shift data, the kron Hessian is unequivocally superior, with its NLL near-zero, while the diag model fails completely. This indicates that WideLeNet's architecture creates strong dependencies between weights in its final layer, making it essential to capture these correlations for good performance. However, this reliance becomes a critical vulnerability under severe distribution shift. As the input data moves further from the training distribution, the learned correlation structure becomes misleading and actively harms predictive ability. This leads the kron model's performance to collapses catastrophically. The diag model also worsen with the increase of the shift intensity, remaining consistenly worse then it's kron counterpart.

2. **The probit link**

When we switch to the probit link, the situation becomes much more nuanced and depends entirely on the degree of distribution shift for both of the metrics.

probit-hessian_ece

probit-hessian_nll

For in-distribution and low-shift data, the simpler diag Hessian is the clear winner across both metrics. It provides a significantly better NLL and, crucially, a better ECE. This outcome is likely tied to the nature of the probit function, which is highly sensitive to the variance of its input. In this low-regularization setting, the kron approximation appears to capture noisy, fine-grained curvature details. These details translate into unstable variance estimates that are detrimental to calibration and likelihood. The diag structure, by ignoring these complex correlations, provides a more robust and stable result, proving more effective when the data is close to the training distribution.

As the data shifts further into out-of-distribution (e.g., rotation angle > 60°), the performance of the two Hessian approximations diverges significantly. The diag model shows a consistent degradation in performance across both metrics as the shift increases: its NLL worsens, and its ECE also increases. In contrast, the kron model exhibits a different behavior. While its NLL remains consistently poor and largely flat, its calibration improves dramatically, with its ECE dropping sharply.
Around a 60° rotation, the kron model's ECE crosses below the diag model's, making it the better-calibrated model under severe distribution shifts. This happens because, in this high-error regime where predictions are failing, the richer curvature information from kron, even if noisy, provides a more honest and accurate picture of the model's rapidly increasing uncertainty. Although its predictions are poor (high NLL), it correctly signals its low confidence, which results in better calibration. The simpler diag structure cannot capture this complex uncertainty landscape, so while its NLL is better, its calibration continues to degrade.