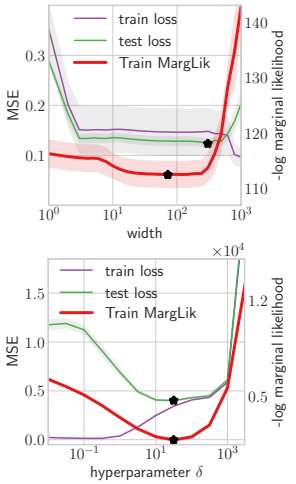
We thank the reviewers for their feedback. A big concern among all reviewers is about experimental results. We emphasize that our main contribution is to derive theoretical connections, but, as per your suggestions, we will add the following new experiments:

1. Hyperparameter tuning for discrete parameters (see top figure on the right on choosing the NN width) and on a real dataset (see bottom figure on the right for setting prior-precision on the "UCI Wine" dataset). We also have results on comparing different architectures (LeNet, AlexNet, ResNet) on CIFAR, which we will add in the paper.
2. Feature maps visualization on real data in the appendix since this takes a lot of space.
3. Comparisons with other kernels and with other BNN-GP method on a small example.



There are also a few concerns by R2 regarding originality and significance of our work. We would like to emphasize that this is the first result connecting training procedures and stationarity conditions of BNNs to GP inference. In particular, no other existing work has been able to express iterations of a VI procedure as GPs (Theorem 3). We agree that this paper takes the first step, but it is an important step.

R3 has some concerns about the GGN approximation, but these have mostly been resolved by other recent works. We have provided an explanation in the response to R3.

***R1**: what does the feature mapping look like?* - We show an example for the toy data in Fig (1b) in the paper. For real datasets, these are too big to visualize which is why we only show kernels. We will add a visualization in the appendix.
***R1**: Your NTK kernel looks very much like the correlation matrix of the output of each data example. What about comparing to other kernels or kernels in standard GPs?* - The NTK kernel is built using Jacobians, i.e., by using the first-order information, which is fundamentally different from other kernels used in GP. We will add visualizations of various kernels in the appendix to show a comparison. Our kernel can be seen as an approximation to the output correlation matrix.
***R1**: What's the influence of increasing or decreasing the number of parameters?* - Increasing the number of parameters can capture complicated information, but then the marginal likelihood penalizes for the increase in number of parameters. This trade-off is clear when we plot it with respect to the network width (see the figure on the right).
***R1**: when using GP, uncertainty should be shown.* - We have these results and will add them in the paper. The GP uncertainty is in line with that of Bayesian NN uncertainty obtained by sampling from the posterior approximation.
***R1**: how about quantitative performance compared to other models and BNN-GP relations? This may reveal the strength or weakness of the method.* - the performance of resulting GP is equal to that of a BNN, so this comparison is not necessary. Other BNN-GP methods are computationally demanding since they require computation and inversion of the kernel, which is why we are restricted to a toy problem (Fig. 2). We will try to add a more realistic example.

***R2**: Laplace approximation can be certainly interpreted as GP in some way?* - It might appear that it is easy to derive this connection explicitly, but until now there are no such results. Our derivation also extends to VI where *every iteration* can be expressed a GP. This result is nontrivial and first of its kind.
***R2**: Also the spherical Gaussian prior seems not to be crucial. Shouldn't other smooth priors work as well?* - This is correct and the method works even for nonsmooth priors such as Laplace. We will emphasize this in the paper.
***R2**: Provide more insights into algorithmic challenges such as runtime, numerics etc.* We will add this in the text.

***R3**: This paper uses neural tangent kernel (NTK) to study BNN posterior approximations.* - It appears that there is a misunderstanding here. The goal is to show that by using approximate posteriors we recover a GP. The NTK appears for Laplace, but for VI the kernel is different.
***R3**: For classification problems, the residuals do not vanish.* - This is not entirely correct. Residuals are gradients of the loss and they tend to zero as the network classification for a data example becomes better and better. See *New insights and perspectives on the natural gradient method* (Martens, 2014).
***R3**: Provide empirical evidence that posterior approximation with GGN have good performance.* - Recent works have clearly shown that GGN based VI algorithms work well; see *Practical Variational Inference for Neural Networks* (Graves, 2011), *Noisy Natural Gradient as Variational Inference* (Zhang, 2017), *Fast and scalable Bayesian deep learning by weight-perturbation in Adam* (Khan, 2018) . We will add a discussion on the accuracy of GGN referring to these papers.
***R3**: all derived GP models have data-dependent likelihood models and authors should acknowledge this limitation* - It is incorrect to say that this is a limitation of the method. Such data-dependent likelihood "approximations" are in fact very common and arise in methods such as: iterative weight least squares, expectation propagation, and even in well known variational bounds such as Jordan and Jaakkola's bound (see Bishop's book). For example, when approximating a binary likelihood, such data-dependent approximations are essential where variance is adjusted to get better approximations. This is not a limitation but an advantage that helps us to figure out important data examples, e.g. boundary points in a classification problem.