We thank all the reviewers for such thoughtful and high-quality reviews as well as for the positive feedback! We will do our best to incorporate all the recommendations in the next revision of the paper.

R1, R3: Experiments on more datasets. We would like to note that we performed all experiments not only on CIFAR100 but also on CIFAR10 dataset, as mentioned in the experimental setup (line 129). All results are similar to the ones on CIFAR100 and presented in appendices. We also experimented with ResNet50 models on ImageNet, but only of standard size due to the computational limitations, therefore we did not include them in the paper. We analysed the behaviour of NLL and CNLL of the models as functions of ensemble size (as in section 4) and the results are the same as the ones in the paper. We will add the results in the appendix.

R1: More thorough related work. We kindly thank the reviewer for the provided references. We had to reduce the related work section due to space limitation but will revise it and make it more broad and thorough in the next version of the paper. Below we comment on the specific related topics mentioned by R1. References are numbered according to R1.

Relation to different ensemble technique [2]. In our work, we consider simple ensembles of neural networks independently trained from random initializations, as they are widely used in practice. Different ensembling modifications, e.g. proposed in [1, 2], go beyond the scope of our work, however, we regard this direction as very interesting.

Relation to Bayesian NNs. Despite the asymptotic nature of our theoretical results, we mainly consider finite ensembles in our experiments. The Bayesian (and other limiting) perspective on ensembling and its connection to our results for $n \to \infty$ is a promising direction for future research.

On model diversity in ensembles. This issue was partially addressed in [6, 3], where it is shown that random initialization allows to explore different modes in function space, which explains why deep ensembles trained with just random initialization perform well in practice. Moreover, as noted in [6], simple diversity inducing techniques, like bagging, may even deteriorate performance in uncertainty estimation.

R1, R2: Other metrics/benchmarks for uncertainty estimation. Considering other metrics/benchmarks is a direction for future research since it requires a separate thorough study, as different metrics have various specifics.

R1: Timing analysis. For a single VGG on CIFAR-100, for network sizes 0.125 / 0.25 / 0.5 / 1 / 2 / 4 / 8 (in standard budgets), testing takes 6.8 / 9.5 / 20 / 33 / 63 / 111 / 227 seconds (batch size 1024), while one training epoch takes 7.8 / 8 / 11 / 16 / 27 / 42 / 80 seconds (batch size 64). For example, for budget $4S$, with a single network / memory split of 4 networks, testing takes 111 / 132 sec, while one training epoch takes 42 / 64 seconds. So training and using memory split is slower than a single network, but only moderately, not four times slower.

R3: Why do we use power law? From the start, we had two main motivations: 1) the theoretical one — theoretical results presented in the paper show the asymptotic power-law behavior of (C)NLL, 2) the practical one — plots of (C)NLL do look similar to power laws in practice. In our experiments, power laws allow both accurate *interpolation* and *extrapolation*. Interpolation: we can approximate (C)NLL with a power law *on a wide range of arguments*. Extrapolation: (C)NLL fitted on a *short* segment approximates well the *full* range of arguments. Hence, we argue that (C)NLL follows a power law and not another function.

R3: The conditions for derivation in equation 2. The only condition on the distribution of ensemble predictions p^* required for the derivation of equation 2 (besides i.i.d.) is separability from zero, i.e. $p^* \in [\epsilon, 1]$, just due to irregular behavior of logarithm near zero. Given that, we can obtain our main theoretical result stated in Proposition 1 — asymptotic power-law behavior of the ensemble NLL. The rigorous derivation is provided in Appendix A.1.

R3: How to use predictions with power laws in practice? Let's say we have a budget B and want to find an optimal memory split (MS). We consider MSs with number of networks $n = 1, 2, 4, 8, \ldots$. If we do not use power-law predictions we can train MSs one by one (one network of size B, then two networks of size $B/2$ and so on) while the quality of the MS starts to decrease. In this case we need to train n networks of size B/n for $n = 1, \ldots, n^* + 1$, where n^* is a number of networks in an optimal MS. If we use power-law predictions in the same manner we did in the paper (see section 7) we need to train $\min(n, 6)$ networks of size B/n for $n = 1, \ldots, n^* + 1$ and then use predictions. After finding n^* we also need to train lacking networks for the optimal MS. As a result, if $n^* \geq 4$, power-law predictions allow training fewer networks, and the higher n^* the higher the gain.

R4: Why calibration removes the double-descent behavior? The accurate answer to this question requires a more thorough study of double descent. Our preliminary experiments show that model overfits in terms of NLL easier than in terms of accuracy or CNLL, therefore double descent of NLL can be observed more often.