# On Power Laws in Deep Ensembles

**Ekaterina Lobacheva[1], Nadezhda Chirkova[1], Maxim Kodryan[1], Dmitry Vetrov[1,2]**
[1]Samsung-HSE Laboratory, National Research University Higher School of Economics
[2]Samsung AI Center Moscow
Moscow, Russia
`{elobacheva,nchirkova,mkodryan,dvetrov}@hse.ru`

## Abstract

Ensembles of deep neural networks are known to achieve state-of-the-art performance in uncertainty estimation and lead to accuracy improvement. In this work, we focus on a classification problem and investigate the behavior of both non-calibrated and calibrated negative log-likelihood (CNLL) of a deep ensemble as a function of the ensemble size and the member network size. We indicate the conditions under which CNLL follows a power law w. r. t. ensemble size or member network size, and analyze the dynamics of the parameters of the discovered power laws. Our important practical finding is that one large network may perform worse than an ensemble of several medium-size networks with the same total number of parameters (we call this ensemble a memory split). Using the detected power law-like dependencies, we can predict (1) the possible gain from the ensembling of networks with given structure, (2) the optimal memory split given a memory budget, based on a relatively small number of trained networks.

## 1   Introduction

Neural networks provide state-of-the-art results in a variety of machine learning tasks, however, several neural network's aspects complicate their usage in practice, including overconfidence [18], vulnerability to adversarial attacks [30], and overfitting [29]. One of the ways to compensate these drawbacks is using deep ensembles, i. e. the ensembles of neural networks trained from different random initialization [18]. In addition to increasing the task-specific metric, e. g. accuracy, the deep ensembles are known to improve the quality of uncertainty estimation, compared to a single network. There is yet no consensus on how to measure the quality of uncertainty estimation. Ashukha et al. [2] consider a wide range of possible metrics and show that the calibrated negative log-likelihood (CNLL) is the most reliable one because it avoids the majority of pitfalls revealed in the same work.

Increasing the size $n$ of the deep ensemble, i. e. the number of networks in the ensemble, is known to improve the performance [2]. The same effect holds for increasing the size $s$ of a neural network, i. e. the number of its parameters. Recent works [3, 22] show that even in an extremely overparameterized regime, increasing $s$ leads to a higher quality. These works also mention a curious effect of non-monotonicity of the test error w. r. t. the network size, called double descent behaviour.

In figure 1, left, we may observe the saturation and stabilization of quality with the growth of both the ensemble size $n$ and the network size $s$. The goal of this work is to study the asymptotic properties of CNLL of deep ensembles as a function of $n$ and $s$. We investigate under which conditions and w. r. t. which dimensions the CNLL follows a power law for deep ensembles in practice. In addition to the horizontal and vertical cuts of the diagram shown in figure 1, left, we also study its diagonal direction, which corresponds to the increase of the total parameter count.

The power-law behaviour of deep ensembles has previously been touched in the literature. Geiger et al. [8] consider simple shallow architectures and reason about the power-law behaviour of the test
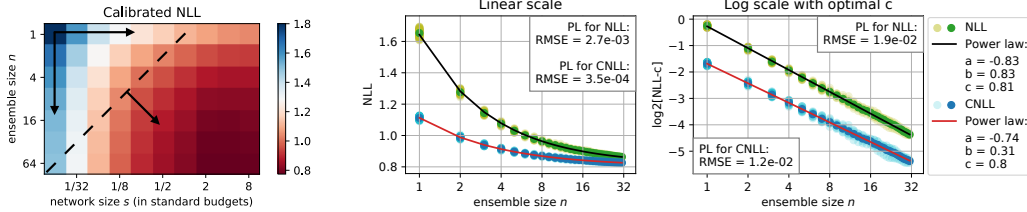
Figure 1: Non-calibrated NLL and CNLL of VGG on CIFAR-100. Left: the $(n, s)$-plane for the CNLL. Middle and right: non-calibrated $\mathrm{NLL}_n$ and $\mathrm{CNLL}_n$ can be closely approximated with a power law (VGG of the commonly used size as an example) .

error of a deep ensemble as a function of $n$ when $n \to \infty$, and of a single network as a function of $s$ when $s \to \infty$. Kaplan et al. [13], Rosenfeld et al. [26] investigate the behaviour of *single* networks of modern architectures and empirically show that their NLL and test error follow power laws w. r. t. the network size $s$. In this work, we perform a broad empirical study of power laws in deep ensembles, relying on the practical setting with properly regularized, commonly used deep neural network architectures. Our main contributions are as follows:

1. for the practically important scenario with NLL calibration, we derive the conditions under which CNLL of an ensemble follows a power law as a function of $n$ when $n \to \infty$;

2. we empirically show that, in practice, the following dependencies can be closely approximated with a power law on the *whole* considered range of their arguments: (a) CNLL of an ensemble as a function of the ensemble size $n \geqslant 1$; (b) CNLL of a single network as a function of the network size $s$; (c) CNLL of an ensemble as a function of the total parameter count;

3. based on the discovered power laws, we make several practically important conclusions regarding the use of deep ensembles in practice, e. g. using a large single network may be less beneficial than using a so-called memory split — an ensemble of several medium-size networks of the same total parameter count;

4. we show that using the discovered power laws for $n \geqslant 1$, and having a small number of trained networks, we can predict the CNLL of the large ensembles and the optimal memory split for a given memory budget.

**Definitions and notation.** In this work, we treat a power law as a family of functions $\mathrm{PL}_m = c + bm^a$, $m = 1, 2, 3, \ldots$; $a < 0$, $b \in \mathbb{R}$, $c \in \mathbb{R}$ are the parameters of the power law. Parameter $c = \lim_{m \to \infty}(c + bm^a) = \lim_{m \to \infty} \mathrm{PL}_m \overset{\text{def}}{=} \mathrm{PL}_\infty$ reflects the asymptote of the power law. Parameter $b = c - \mathrm{PL}_1 = \mathrm{PL}_\infty - \mathrm{PL}_1$ reflects the difference between the starting point of the power law and its asymptote. Parameter $a$ reflects the speed of approaching the asymptote. In the rest of the work, $(\mathrm{C})\mathrm{NLL}_m$ denotes (C)NLL as a function of $m$.

## 2 Theoretical view

The primary goal of this work is to perform the empirical study of the conditions under which NLL and CNLL of deep ensembles follow a power law. Before diving into a discussion about our empirical findings, we first provide a theoretical motivation for anticipating power laws in deep ensembles, and discuss the applicability of this theoretical reasoning to the practically important scenario with calibration.

We begin with a theoretical analysis of the non-calibrated NLL of a deep ensemble as a function of ensemble size $n$. Assume that an ensemble consists of $n$ models that return independent identically distributed probabilities $p^*_{\mathrm{obj},i} \in [0, 1], i = 1, \ldots, n$ of the correct class for a single object from the dataset $\mathcal{D}$. Hereinafter, operator $*$ denotes retrieving the prediction for the correct class. We introduce

the *model-average* NLL of an ensemble of size $n$ for the *given object*:

$$\text{NLL}_n^{\text{obj}} = -\mathbb{E}\log\left(\frac{1}{n}\sum_{i=1}^{n}p_{\text{obj},i}^*\right). \tag{1}$$

The expectation in (1) is taken over all possible models that may constitute the ensemble (e. g. random initializations). The following proposition describes the asymptotic power-law behavior of $\text{NLL}_n^{\text{obj}}$ as a function of the ensemble size.

**Proposition 1** *Consider an ensemble of $n$ models, each producing independent and identically distributed probabilities of the correct class for a given object: $p_{\text{obj},i}^* \in [\epsilon_{\text{obj}}, 1]$, $\epsilon_{\text{obj}} > 0$, $i = 1,\ldots,n$. Let $\mu_{\text{obj}} = \mathbb{E}p_{\text{obj},i}^*$ and $\sigma_{\text{obj}}^2 = \mathbb{D}p_{\text{obj},i}^*$ be, respectively, the mean and variance of the distribution of probabilities. Then the model-average NLL of the ensemble for a single object can be decomposed as follows:*

$$\text{NLL}_n^{\text{obj}} = \text{NLL}_\infty^{\text{obj}} + \frac{1}{n}\frac{\sigma_{\text{obj}}^2}{2\mu_{\text{obj}}^2} + \mathcal{O}\left(\frac{1}{n^2}\right). \tag{2}$$

*where $\text{NLL}_\infty^{\text{obj}} = -\log(\mu_{\text{obj}})$ is the "infinite" ensemble NLL for the given object.*

The proof is based on the Taylor expansions for the moments of functions of random variables, we provide it in Appendix A.1. The assumption about the lower limit of model predictions $\epsilon_{\text{obj}} > 0$ is necessary for the accurate derivation of the asymptotic in (2). We argue, however, that this condition is fulfilled in practice as real softmax outputs of neural networks are always positive and separated from zero.

The model-average NLL of an ensemble of size $n$ on the whole dataset, $\text{NLL}_n$, can be obtained via summing $\text{NLL}_n^{\text{obj}}$ over objects, which implies that $\text{NLL}_n$ also behaves as $c + bn^{-1}$, where $c, b > 0$ are constants w. r. t. $n$, as $n \to \infty$. However, for the finite-range $n$, the dependency in $\text{NLL}_n$ may be more complex.

Ashukha et al. [2] emphasize that the comparison of the NLLs of different models with suboptimal softmax temperature may lead to an arbitrary ranking of the models, so the comparison should only be performed after *calibration*, i. e. with optimally selected temperature $\tau$. The model-average CNLL of an ensemble of size $n$, measured on the whole dataset $\mathcal{D}$, is defined as follows:

$$\text{CNLL}_n = \mathbb{E}\min_{\tau>0}\left\{-\sum_{\text{obj}\in\mathcal{D}}\log\bar{p}_{\text{obj},n}^*(\tau)\right\}, \tag{3}$$

where the expectation is also taken over models, and $\bar{p}_{\text{obj},n}(\tau) \in [0,1]^K$ is the distribution over $K$ classes output by the ensemble of $n$ networks with softmax temperature $\tau$. Ashukha et al. [2] obtain this distribution by averaging predictions $p_{\text{obj},i} \in [0,1]^K$ of the member networks $i = 1,\ldots,n$ for a given object and applying the temperature $\tau > 0$ on top of the ensemble: $\bar{p}_{\text{obj},n}(\tau) = \text{softmax}\{(\log(\frac{1}{n}\sum_{i=1}^{n}p_{\text{obj},i}))/\tau\}$. This is a native way of calibrating, in a sense that we plug the ensemble into a standard procedure of calibrating an arbitrary model. We refer to the described calibration procedure as applying temperature *after* averaging. In our work, we also consider another way of calibrating, namely applying temperature *before* averaging: $\bar{p}_{\text{obj},n}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\text{softmax}\{\log(p_{\text{obj},i})/\tau\}$. The two calibration procedures perform similarly in practice, in most of the cases the second one performs slightly better (see Appendix C.1).

The following series of derivations helps to connect the non-calibrated and calibrated NLLs. If we fix some $\tau > 0$ and apply it *before* averaging, $\bar{p}_{\text{obj},n}(\tau)$ fits the form of the ensemble in the right-hand side of equation (1), and according to Proposition 1, we obtain that the model-average NLL of an $n$-size ensemble with fixed temperature $\tau$, $\text{NLL}_n(\tau)$, follows a power law w. r. t. $n$ as $n \to \infty$. Applying $\tau$ *after* averaging complicates the derivation, but the same result is generally still valid, see Appendix A.2. However, the parameter $b$ of the power law may become negative for certain values of $\tau$. In contrast, when we apply $\tau$ before averaging, $b$ always remains positive, see eq. (2).

Minimizing $\text{NLL}_n(\tau)$ w. r. t. $\tau$ results in a lower envelope of the (asymptotic) power laws:

$$\text{LE-NLL}_n = \min_{\tau>0}\text{NLL}_n(\tau), \quad \text{NLL}_n(\tau) \stackrel{n\to\infty}{\sim} \text{PL}_n. \tag{4}$$

3

The lower envelope of power laws also follows an (asymptotic) power law. Consider for simplicity a finite set of temperatures $\{\tau_1, \ldots, \tau_T\}$, which is the conventional practical case. As each of $\text{NLL}_n(\tau_t), t = 1, \ldots, T$ converges to its asymptote $c(\tau_t)$, there exists an optimal temperature $\tau_{t^*}$ corresponding to the lowest $c(\tau_{t^*})$. The above implies that starting from some point $n$, LE-NLL$_n$ will coincide with $\text{NLL}_n(\tau_{t^*})$ and hence follow its power law. We refer to Appendix A.3 for further discussion on continuous temperature.

Substituting the definition of $\text{NLL}_n(\tau)$ into (4) results in:

$$\text{LE-NLL}_n = \min_{\tau > 0} \mathbb{E}\left\{ - \sum_{\text{obj} \in \mathcal{D}} \log \bar{p}^*_{\text{obj},n}(\tau) \right\}, \tag{5}$$

from which we obtain that the only difference between LE-NLL$_n$ and CNLL$_n$ is the order of the minimum operation and the expectation. Although this results in another calibration procedure than the commonly used one, we show in Appendix D that the difference between the values of LE-NLL$_n$ and CNLL$_n$ is negligible in practice. Conceptually, applying expectation inside the minimum is also a reasonable setting: in this case, when choosing the optimal $\tau$, we use the more reliable estimate of the NLL of the $n$-size ensemble with temperature $\tau$. This setting is not generally considered in practice, since it requires training several ensembles and, as a result, is more computationally expensive. In the experiments we follow the definition of CNLL (3) to consider the most practical scenario.

To sum up, in this section we derived an asymptotic power law for LE-NLL that may be treated as another definition of CNLL, and that closely approximates the commonly used CNLL in practice.

## 3 Experimental setup

We conduct our experiments with convolutional neural networks, WideResNet [33] and VGG16 [27], on CIFAR-10 [16] and CIFAR-100 [17] datasets. We consider a wide range of network sizes $s$ by varying the width factor $w$: for VGG / WideResNet, we use convolutional layers with $[w, 2w, 4w, 8w]$ / $[w, 2w, 4w]$ filters, and fully-connected layers with $8w$ / $4w$ neurons. For VGG / WideResNet, we consider $2 \leqslant w \leqslant 181$ / $5 \leqslant w \leqslant 453$; $w = 64 / 160$ corresponds to a standard, commonly used, configuration with $s_{\text{standard}} = 15.3\text{M}$ / $36.8\text{M}$ parameters. These sizes are later referred to as the standard budgets. For each network size, we tune hyperparameters (weight decay and dropout) using grid search. We train all networks for 200 epochs with SGD with an annealing learning schedule and a batch size of 128. We aim to follow the practical scenario in the experiments, so we use the definition CNLL (3), not LE-NLL (4). Following [2], we use the "test-time cross-validation" to compute the CNLL. We apply the temperature before averaging, the motivation for this is given in section 4. More details are given in Appendix B.

For each network size $s$, we train at least $\ell = \max\{N, 8s_{\text{standard}}/s\}$ networks, $N = 64 / 12$ for VGG / WideReNet. For each $(n, s)$ pair, given the pool of $\ell$ trained networks of size $s$, we construct $\lfloor \frac{\ell}{n} \rfloor$ ensembles of $n$ distinct networks. The NLLs of these ensembles have some variance, so in all experiments, we average NLL over $\lfloor \frac{\ell}{n} \rfloor$ runs. We use these values to approximate NLL with a power law along the different directions of the $(n, s)$-plane. For this, we only consider points that were averaged over at least three runs.

**Approximating sequences with power laws.** Given an arbitrary sequence $\{\hat{y}_m\}$, $m = 1, \ldots, M$, we approximate it with a power law $\text{PL}_m = c + bm^a$. In the rest of the work, we use the hat notation $\hat{y}_m$ to denote the observed data, while the value without hat, $y_m$, denotes $y$ as a function of $m$. To fit the parameters $a$, $b$, $c$, we solve the following optimization task using BFGS:

$$(a, b, c) = \operatorname*{argmin}_{a,b,c} \frac{1}{M} \sum_{m=1}^{M} \left( \log_2(\hat{y}_m - c) - \log_2(bm^a) \right)^2. \tag{6}$$

We use the logarithmic scale to pay more attention to the small differences between values $\hat{y}_m$ for large $m$. For a fixed $c$, optimizing the given loss is equivalent to fitting the linear regression model with one factor $\log_2 m$ in the space $\log_2 m$ — $\log_2(y_m - c)$ (see fig. 1, right as an example).

# 4 NLL as a function of ensemble size

In this section, we would like to answer the question, whether the NLL as the function of ensemble size can be described by a power law in practice. We consider both calibrated NLL and NLL with a fixed temperature. To answer the stated question, we fit the parameters $a$, $b$, $c$ of the power law on the points $\widehat{\mathrm{NLL}}_n(\tau)$ or $\widehat{\mathrm{CNLL}}_n$, $n = 1, 2, 3, \ldots$, using the method described in section 3, and analyze the resulting parameters and the quality of the approximation.

As we show in Appendix C.2, when the temperature is applied *after* averaging, $\mathrm{NLL}_n(\tau)$ is, in some cases, an increasing function of $n$. As for CNLL, we found settings when $\mathrm{CNLL}_n$ with the temperature applied *after* averaging is not a convex function for small $n$, and as a result, cannot be described by a power law. In the rest of the work, we apply temperature *before* averaging, as in this case, both $\mathrm{NLL}_n(\tau)$ and $\mathrm{CNLL}_n$ can be closely approximated with power laws in all considered cases.

**NLL with fixed temperature.** For all considered dataset–architecture pairs, and for all temperatures, $\widehat{\mathrm{NLL}}_n(\tau)$ with fixed $\tau$ can be closely approximated with a power law. Figure 1, middle and right shows an example approximation for VGG of the commonly used size with the temperature equal to one. Figure 2 shows the dynamics of the parameters $a$, $b$, $c$ of power laws, approximating the NLL with a fixed temperature of ensembles of different network sizes and for different temperatures, for VGG on a CIFAR-100 dataset. The rightmost plot reports the quality of approximation measured with RMSE in the *log*-space. We note that even the highest RMSE in the *log*-space corresponds to the low RMSE in the *linear* space (the RMSE in the *linear* space is less than 0.006 for all lines in figure 2).

In theory, starting from large enough $n$, $\mathrm{NLL}_n(\tau)$ follows a power law with parameter $a$ equal to -1, and for small $n$, more than one terms in eq. (2) are significant, resulting in a complex dependency $\mathrm{NLL}_n(\tau)$. In practice, we observe the power-law behaviour for the whole considered graph $\mathrm{NLL}_n(\tau)$, $n \geqslant 1$, but with $a$ slightly larger than $-1$. This result is consistent for all considered dataset–architecture pairs, see Appendix E.1.

When the temperature grows, the general behaviour is that $a$ approaches -1 more and more tightly. This behaviour breaks for the ensembles of small networks (blue lines). The reason is that the number of trained small networks is large, and the NLL for large ensembles with high temperature is noisy in log-scale, so the approximation of NLL with power law is slightly worse than for other settings, as confirmed in the rightmost plot of fig. 2. Nevertheless, these approximations are still very close to the data, we present the corresponding plots in Appendix E.2.

Parameter $b = \mathrm{NLL}_1(\tau) - \mathrm{NLL}_\infty(\tau)$ reflects the potential gain from the ensembling of the networks with the given network size and the given temperature. For a particular network size, the gain is higher for low temperatures, since networks with low temperatures are overconfident in their predictions, and ensembling reduces this overconfidence. With high temperatures, the predictions of both single network and ensemble get closer to the uniform distribution over classes, and $b$ approaches zero.

Parameter $c$ approximates the quality of the "infinite"-size ensemble, $\mathrm{NLL}_\infty(\tau)$. For each network size, there is an optimal temperature, which may be either higher or lower than one, depending on the dataset–architecture combination (see Appendix E.1 for more examples). This shows that even large ensembles need calibration. Moreover, the optimal temperature increases, when the network size grows. Therefore, not only large single networks are more confident in their predictions than small single networks, but the same holds even for large ensembles. Higher optimal temperature reduces their confidence. We notice that for given network size, the optimal temperature converges when $n \to \infty$, we show the corresponding plots in Appendix F.

**NLL with calibration.** When the temperature is applied before averaging, $\widehat{\mathrm{CNLL}}_n$ can be closely approximated with a power law for all considered dataset–architecture pairs, see Appendix G. Figure 3 shows how the resulting parameters of the power law change when the network size increases for different settings. The rightmost plot reports the quality of the approximation.

In figure 3, we observe that for WideResNet, parameter $b$ decreases, as $s$ becomes large, and $c$ starts growing for large $s$. For VGG, this effect also occurs in a light form but is almost invisible at the plot. This suggests that large networks gain less from the ensembling, and therefore the ensembles of larger networks are less effective than the ensembles of smaller networks. We suppose, the described
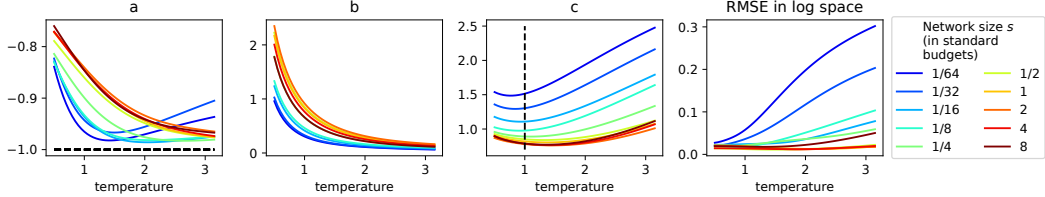
5

Figure 2: Parameters of power laws and the quality of the approximation for $\mathrm{NLL}_n(\tau)$ with a fixed temperature $\tau$ for VGG on CIFAR-100.
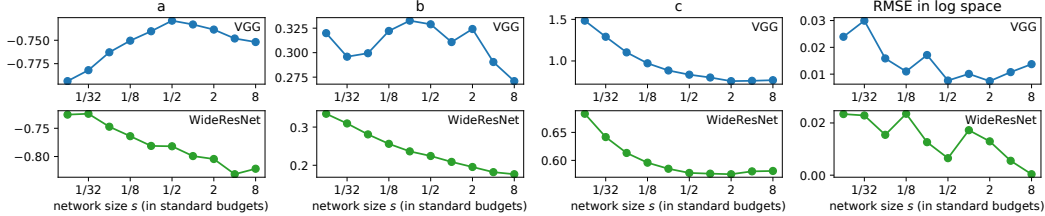


Figure 3: Parameters of power laws and the quality of the approximation for $\mathrm{CNLL}_n$ for different network sizes $s$. VGG and WideResNet on CIFAR-100.

effect is a consequence of under-regularization (the large networks need more careful hyperparameter tuning and regularization), because we also observed the described effect in a more strong form for the networks with all regularization turned off, see Appendix H. However, the described effect might also be a consequence of the decreased diversity of wider networks [23], and needs further investigation.

## 5 NLL as a function of network size

In this section, we analyze the behaviour of the NLL of the ensemble of a fixed size $n$ as a function of the member network size $s$. We consider both non-calibrated and calibrated NLL, and analyze separately cases $n = 1$ and $n > 1$. Geiger et al. [8] reason about a power law of accuracy for $n = 1$ when $s \to \infty$, considering shallow fully-connected networks on the MNIST dataset. We would like to check, whether $\widehat{(\mathrm{C})\mathrm{NLL}}_s$ can be approximated with a power law on the whole reasonable range of $s$ in practice.

**Single network.** Figure 4, left shows the NLL with $\tau = 1$ and the CNLL of a single VGG on the CIFAR-100 dataset as a function of the network size. We observe the double descent behaviour [22, 3] of the non-calibrated NLL, which could not be approximated with a power law for the considered range of $s$. The calibration removes the double-descent behaviour, and allows a close power-law approximation, as confirmed in the middle plot of figure 4. Interestingly, parameter $a$ is close to $-0.5$, which coincides with the results of Geiger et al. [8] derived for the test error. The results for other dataset–architecture pairs are given in Appendix I.

Nakkiran et al. [22] observe the double descent behaviour of *accuracy* as a function of the network size for highly overfitted networks, when training networks without regularization, with label noise, and for much more epochs than is usually needed in practice. In our practical setting, accuracy and CNLL are the monotonic functions of the network size, while for the non-calibrated NLL, the double descent behaviour is observed. Ashukha et al. [2] point out that accuracy and CNLL usually correlate, so we hypothesize that the double descent *may be* observed for CNLL in the same scenarios when it is observed for accuracy, while the non-calibrated NLL exhibits the double descent at the earlier epochs in these scenarios. To sum up, our results support the conclusions of [2] that the comparison of the NLL of the models of different *sizes* should only be performed with an optimal temperature.

**Ensemble.** As can be seen from figure 4, right, for ensemble sizes $n > 1$, CNLL starts increasing at some network size $s$. This agrees with the behaviour of parameter $c$ of the power law for CNLL
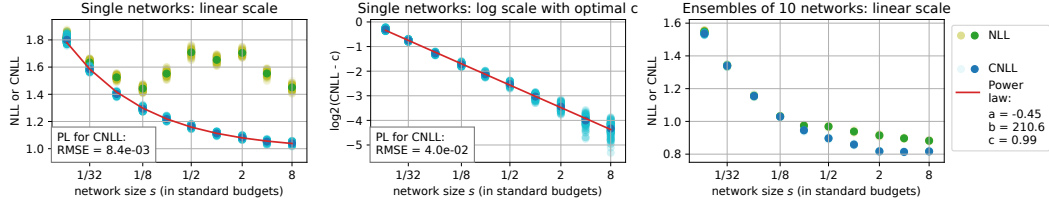
6

Figure 4: Non-calibrated $\mathrm{NLL}_s$ and $\mathrm{CNLL}_s$ for VGG on CIFAR-100. Left and middle: for a single network, $\mathrm{NLL}_s$ exhibits double descent, while $\mathrm{CNLL}_s$ can be closely approximated with a power law. Right: $\mathrm{NLL}_s$ and $\mathrm{CNLL}_s$ of an ensemble of several networks may be non-monotonic functions.
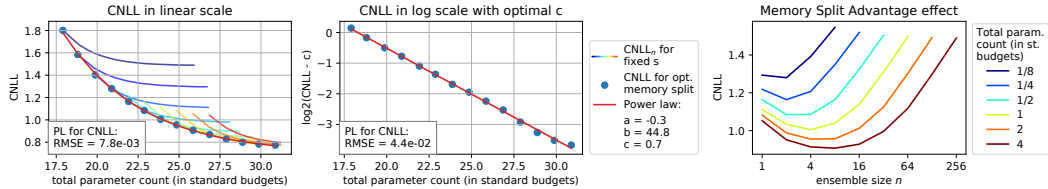


Figure 5: Left and middle: $\mathrm{CNLL}_B$ for VGG on CIFAR-100 can be closely approximated with a power law. $\mathrm{CNLL}_B$ is a lower envelope of $\mathrm{CNLL}_n$ for different network sizes $s$. Right: Memory Split Advantage effect, VGG on CIFAR-100. For different memory budgets $B$, the optimal CNLL is achieved at $n > 1$.

shown in figure 3 and was discussed in section 4. Because of this behaviour, we do not perform experiments on approximating $\mathrm{CNLL}_s$ for $n > 1$ with a power law.

## 6 NLL as a function of the total parameter count

In the previous sections, we analyzed the vertical and horizontal cuts of the $(n, s)$-plane shown in figure 1, left. In this section, we analyze the diagonal cuts of this space. One direction of diagonal corresponds to the fixed total parameter count, later referred to as a memory budget, and the orthogonal direction reflects the increasing budget.

We firstly investigate CNLL as a function of the memory budget. In figure 5, left, we plot sequences $\widehat{\mathrm{CNNL}}_n$ for different network sizes $s$, aligning plots by the total parameter count. CNLL as a function of the memory budget is then introduced as the lower envelope of the described plots. As in the previous sections, we approximate this function with the power law, and observe, that the approximation is tight, the corresponding visualization is given in figure 5, middle. The same result for other dataset-architecture pairs is shown in Appendix J.

Another, practically important, effect is that the lower envelope may be reached at $n > 1$. In other words, for a fixed memory budget, a single network may perform worse than an ensemble of several medium-size networks of the same total parameter count, called a memory split in the subsequent discussion. We refer to the described effect itself as a Memory Split Advantage effect (MSA effect). We further illustrate the MSA effect in figure 5, right, where each line corresponds to a particular memory budget, the x-axis denotes the number of networks in the memory split, and the lowest CNLL, denoted at the y-axis, is achieved at $n > 1$ for all lines. We consistently observe the MSA effect for different settings and metrics, i. e. CNLL and accuracy, for a wide range of budgets, see Appendix J. We note that the MSA-effect holds even for budgets smaller than the standard budget. We also show in appendix J that using the memory split with the relatively small number of networks is only moderately slower than using a single wide network, in both training and testing stages. We describe the memory split advantage effect in more details in [4].
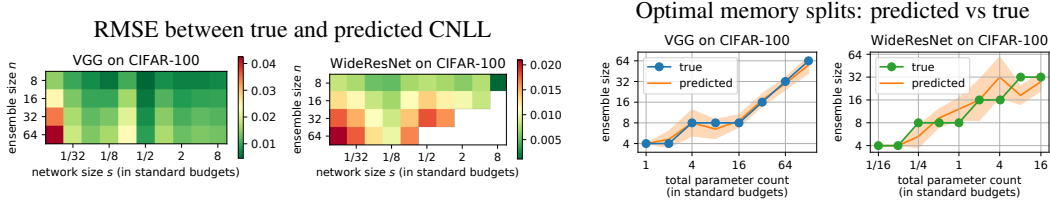
7

Figure 6: Predictions based on $\mathrm{CNLL}_n$ power laws for VGG and WideResNet on CIFAR-100. Predictions are made for large $n$ based on $n = 1..4$. Left pair: RMSE between true and predicted CNLL. Right pair: predicted optimal memory splits vs true ones. Mean $\pm$ standard deviation is shown for predictions.

## 7 Prediction based on power laws

One of the advantages of a power law is that, given a few starting points $y_1, \ldots, y_m$ satisfying the power law, one can exactly predict values $y_i$ for any $i \gg m$. In this section, we check whether the power laws discovered in section 4 are stable enough to allow accurate predictions.

We use the CNLL of the ensembles of sizes $1 - 4$ as starting points, and predict the CNLL of larger ensembles. We firstly conduct the experiment using the values of starting points, obtained by averaging over a large number of runs. In this case, the CNLL of large ensembles may be predicted with high precision, see Appendix K. Secondly, we conduct the experiment in the practical setting, when the values of starting points were obtained using only 6 trained networks (using 6 networks allows the more stable estimation of CNLL of ensembles of sizes $1-3$). The two left plots of figure 6 report the error of the prediction for the different ensemble sizes and network sizes of VGG and WideResNet on the CIFAR-100 dataset. The plots for other settings are given in Appendix K. The experiment was repeated 10 times for VGG and 5 times for WideResNet with the independent sets of networks, and we report the average error. The error is $1 - 2$ orders smaller than the value of CNLL, and based on this, we conclude that the discovered power laws allow quite accurate predictions.

In section 6, we introduced memory splitting, a simple yet effective method of improving the quality of the network, given the memory budget $B$. Using the obtained predictions for CNLL, we can now predict the optimal memory split (OMS) for a fixed $B$ by selecting the optimum at a specific diagonal of the predicted $(n, s)$-plane, see Appendix K for more details. We show the results for the practical setting with 6 given networks in figure 6, right. The plots depict the number of networks $n$ in the true and predicted OMS; the network size can be uniquely determined by $B$ and $n$. In most cases, the discovered power laws predict either the exact or the neighboring split. If we predict the neighboring split, the difference in CNLL between the true and predicted splits is negligible, i.e. of the same order as the errors presented in figure 6, left.

To sum up, we observe that the discovered power laws not only *interpolate* $\widehat{\mathrm{CNLL}}_n$ on the *whole* considered range of $n$, but also is able to *extrapolate* this sequence, i.e. CNLL fitted on a *short* segment of $n$ approximates well the *full* range, providing an argument for using particularly power laws and not other functions.

## 8 Related Work

**Deep ensembles and overparameterization.** The two main approaches to improve deep neural networks accuracy are ensembling and increasing network size. While a bunch of works report the quantitative influence of the above-mentioned techniques on model quality [7, 12, 18, 22, 24, 25], few investigate the qualitative side of the effect. Some recent works [5, 8, 9] consider a simplified or narrowed setup to tackle it. For instance, Geiger et al. [8] similarly discover the power laws in test error w. r. t. model and ensemble size for simple binary classification with hinge loss, and give a heuristic argument supporting their findings. We provide an extensive theoretical and empirical justification of similar claims for the calibrated NLL using modern architectures and datasets. Other layers of works on studying neural network ensembles and overparameterized models include but not limited to the

Bayesian perspective [10, 31, 32], ensembles diversity improvement techniques [14, 20, 28, 34], neural tangent kernel (NTK) view on overparameterized neural networks [1, 11, 19], etc.

**Power laws for predictions.** A few recent works also empirically discover power laws with respect to data and model size and use them to extrapolate the performance on small models/datasets to larger scales [13, 26]. Their findings even allow estimating the optimal compute budget allocation given limited resources. However, these studies do not account for the ensembling of models and the calibration of NLL.

**MSA-effect.** Concurrently with our work, Kondratyuk et al. [15] investigate a similar effect for budgets measured in FLOPs. Earlier, an MSA-like effect has also been noted in [6, 21]. However, the mentioned works did not consider the proper regularization of networks of different sizes and did not propose the method for predicting the OMS, while both aspects are important in practice.

## 9  Conclusion

In this work, we investigated the power-law behaviour of CNLL of deep ensembles as a function of ensemble size $n$ and network size $s$ and observed the following power laws. Firstly, with a minor modification of the calibration procedure, CNLL as a function of $n$ follows a power law on the wide finite range of $n$, starting from $n = 1$, but with the power parameter slightly higher than the one derived theoretically. Secondly, the CNLL of a single network follows a power law as a function of the network size $s$ on the whole reasonable range of network sizes, with the power parameter approximately the same as derived. Thirdly, the CNLL also follows a power law as a function of the total parameter count (memory budget). The discovered power laws allow predicting the quality of large ensembles based on the quality of the smaller ensembles consisting of networks with the same architecture. The practically important finding is that for a given memory budget, the number of networks in the optimal memory split is usually much higher than one, and can be predicted using the discovered power laws. Our source code is available at `https://github.com/nadiinchi/power_laws_deep_ensembles`.

## Broader Impact

In this work, we provide an empirical and theoretical study of existing models (namely, deep ensembles); we propose neither new technologies nor architectures, thus we are not aware of its specific ethical or future societal impact. We, however, would like to point out a few benefits gained from our findings, such as optimization of resource consumption when training neural networks and contribution to the overall understanding of neural models. As far as we are concerned, no negative consequences may follow from our research.

## References

[1] Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. (2019). On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8141–8150.

[2] Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. (2020). Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*.

[3] Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Science*, 116(32):15849–15854.

[4] Chirkova, N., Lobacheva, E., and Vetrov, D. (2020). Deep ensembles on a fixed memory budget: One wide network or several thinner ones?

[5] d'Ascoli, S., Refinetti, M., Biroli, G., and Krzakala, F. (2020). Double trouble in double descent: Bias and variance (s) in the lazy regime. *arXiv preprint arXiv:2003.01054*.

[6] Dutt, A., Pellerin, D., and Quénot, G. (2018). Coupled ensembles of neural networks. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6.

[7] Fort, S., Hu, H., and Lakshminarayanan, B. (2019). Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*.

[8] Geiger, M., Jacot, A., Spigler, S., Gabriel, F., Sagun, L., d'Ascoli, S., Biroli, G., Hongler, C., and Wyart, M. (2020). Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401.

[9] Geiger, M., Spigler, S., d'Ascoli, S., Sagun, L., Baity-Jesi, M., Biroli, G., and Wyart, M. (2019). Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100(1):012115.

[10] He, B., Lakshminarayanan, B., and Teh, Y. W. (2020). Bayesian deep ensembles via the neural tangent kernel. *arXiv preprint arXiv:2007.05864*.

[11] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580.

[12] Ju, C., Bibaut, A., and van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818.

[13] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

[14] Kim, W., Goyal, B., Chawla, K., Lee, J., and Kwon, K. (2018). Attention-based ensemble for deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 736–751.

[15] Kondratyuk, D., Tan, M., Brown, M., and Gong, B. (2020). When ensembling smaller models is more efficient than single large models.

[16] Krizhevsky, A., Nair, V., and Hinton, G. CIFAR-10 (canadian institute for advanced research).

[17] Krizhevsky, A., Nair, V., and Hinton, G. CIFAR-100 (canadian institute for advanced research).

[18] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*.

[19] Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8572–8583.

[20] Lee, S., Prakash, S. P. S., Cogswell, M., Ranjan, V., Crandall, D., and Batra, D. (2016). Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pages 2119–2127.

[21] Li, H., Ng, J. Y.-H., and Natsev, P. (2019). Ensemblenet: End-to-end optimization of multi-headed models.

[22] Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2020). Deep double descent: Where bigger models and more data hurt. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[23] Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., and Mitliagkas, I. (2018). A modern take on the bias-variance tradeoff in neural networks. *arXiv preprint arXiv:1810.08591*.

[24] Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. (2018). Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*.

[25] Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*.

[26] Rosenfeld, J. S., Rosenfeld, A., Belinkov, Y., and Shavit, N. (2019). A constructive prediction of the generalization error across scales. *arXiv preprint arXiv:1909.12673*.

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

[28] Sinha, S., Bharadhwaj, H., Goyal, A., Larochelle, H., Garg, A., and Shkurti, F. (2020). Dibs: Diversity inducing information bottleneck in model ensembles. *arXiv preprint arXiv:2003.04514*.

[29] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

[30] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *International Conference on Learning Representations*.

[31] Wilson, A. G. (2020). The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*.

[32] Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*.

[33] Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *British Machine Vision Conference*.

[34] Zaidi, S., Zela, A., Elsken, T., Holmes, C., Hutter, F., and Teh, Y. W. (2020). Neural ensemble search for performant and calibrated predictions. *arXiv preprint arXiv:2006.08573*.

## A Theory

### A.1 Proof of Proposition 1

In this section, we prove Proposition 1 about the asymptotic power law behavior of the model-average NLL of the ensemble for a single object, $\mathrm{NLL}_n^{\mathrm{obj}}$, as a function of the ensemble size $n$.

**Proof.** In what follows, by abuse of notation, the subscript $\mathrm{obj}$ is omitted. Consider $n$ independent identically distributed random variables $p_i^* \in [\epsilon, 1]$, $\epsilon > 0$ with mean $\mu = \mathbb{E}p_i^*$ and variance $\sigma^2 = \mathbb{D}p_i^*$. Denote $\bar{p}_n^* = \frac{1}{n}\sum_{i=1}^n p_i^*$. The cornerstone of the proof is doing Taylor expansion of logarithm around $\mu$ and obtaining the following expression:

$$\mathrm{NLL}_n^{\mathrm{obj}} = -\mathbb{E}\log(\bar{p}_n^*) = \mathbb{E}\left[-\log(\mu) + \frac{\mu - \bar{p}_n^*}{\mu} + \frac{(\mu - \bar{p}_n^*)^2}{2\mu^2} + R_2(\mu - \bar{p}_n^*)\right] =$$

$$= -\log(\mu) + \frac{1}{n}\frac{\sigma^2}{2\mu^2} + \mathbb{E}\left[R_2(\mu - \bar{p}_n^*)\right], \quad (7)$$

where $R_m(\mu - \bar{p}_n^*) = -\log\left(\frac{\bar{p}_n^*}{\mu}\right) - \sum_{k=1}^m \frac{(\mu - \bar{p}_n^*)^k}{k\mu^k}$ is the remainder term. First we show how to bound $\mathbb{E}\left[R_m(\mu - \bar{p}_n^*)\right]$ w.r.t. $n$ for general $m$ and then use this bound to derive the $\mathcal{O}\left(\frac{1}{n^2}\right)$ asymptotic for $\mathbb{E}\left[R_2(\mu - \bar{p}_n^*)\right]$.

Let us fix some $0 < \varepsilon < \mu$ and split the expectation of the remainder into two terms:

$$\mathbb{E}\left[R_m(\mu - \bar{p}_n^*)\right] = \mathbb{E}\left[R_m(\mu - \bar{p}_n^*) \cdot \mathbb{1}[|\bar{p}_n^* - \mu| > \varepsilon]\right] + \mathbb{E}\left[R_m(\mu - \bar{p}_n^*) \cdot \mathbb{1}[|\bar{p}_n^* - \mu| \leq \varepsilon]\right]. \quad (8)$$

Applying the Hoeffding's inequality to $\bar{p}_n^*$ allows us to exponentially bound the probability of deviation from $\mu$ by $\varepsilon$:

$$P(|\bar{p}_n^* - \mu| > \varepsilon) \leq 2\exp\left(-2n\varepsilon^2\right). \quad (9)$$

Taking inequality (9) into account, we can bound the first term in (8) as follows:

$$\left|\mathbb{E}\left[R_m(\mu - \bar{p}_n^*) \cdot \mathbb{1}[|\bar{p}_n^* - \mu| > \varepsilon]\right]\right| \leq 2M\exp\left(-2n\varepsilon^2\right), \quad (10)$$

where $M = \max_{\bar{p}_n^* \in [\epsilon, 1]} |R_m(\mu - \bar{p}_n^*)|$.

In the second term of (8), as $|\bar{p}_n^* - \mu| \leq \varepsilon < \mu$, the remainder term can be expanded as a Taylor series $R_m(\mu - \bar{p}_n^*) = \sum_{k=m+1}^\infty \frac{(\mu - \bar{p}_n^*)^k}{k\mu^k}$ and thus bounded as follows:

$$|R_m(\mu - \bar{p}_n^*)| \leq \sum_{k=m+1}^\infty \frac{|\mu - \bar{p}_n^*|^k}{k\mu^k} \leq \sum_{k=m+1}^\infty \frac{\varepsilon^k}{k\mu^k}. \quad (11)$$

From that we derive the following bound on the second term in (8):

$$\left|\mathbb{E}\left[R_m(\mu - \bar{p}_n^*) \cdot \mathbb{1}[|\bar{p}_n^* - \mu| \leq \varepsilon]\right]\right| \leq \sum_{k=m+1}^\infty \frac{\varepsilon^k}{k\mu^k} = \mathcal{O}\left(\varepsilon^{m+1}\right). \quad (12)$$

If we fix any $0 < \delta < 1$ and choose the following sequence of $\varepsilon_n = n^{-\frac{1-\delta}{2}}$,[1] the first bound (10) becomes $2M\exp\left(-2n^\delta\right)$, i.e. exponential in $n$, and the second bound (12) becomes $\mathcal{O}\left(n^{-\frac{(m+1)(1-\delta)}{2}}\right)$. By taking $\delta = \frac{1}{5}$, we obtain that $\mathbb{E}\left[R_4(\mu - \bar{p}_n^*)\right] = \mathcal{O}\left(\frac{1}{n^2}\right)$.

To conclude the proof, we note that

$$\mathbb{E}\left[R_2(\mu - \bar{p}_n^*)\right] = \frac{\mathbb{E}(\mu - \bar{p}_n^*)^3}{3\mu^3} + \frac{\mathbb{E}(\mu - \bar{p}_n^*)^4}{4\mu^4} + \mathbb{E}\left[R_4(\mu - \bar{p}_n^*)\right] \quad (13)$$

and $\mathbb{E}(\mu - \bar{p}_n^*)^3 = \frac{\mathbb{E}(\mu - p_i^*)^3}{n^2} = \mathcal{O}\left(\frac{1}{n^2}\right)$, $\mathbb{E}(\mu - \bar{p}_n^*)^4 = \frac{3\sigma^4}{n^2} + \mathcal{O}\left(\frac{1}{n^3}\right) = \mathcal{O}\left(\frac{1}{n^2}\right)$. ∎

### A.2 Power law in NLL with temperature applied after averaging

Here we consider the power-law behaviour of the calibrated NLL of the ensemble, when the temperature is applied after averaging. Recall that $K$ is the number of classes and operator $*$ denotes retrieving the prediction for the correct class. Denote $\bar{p}_n = \frac{1}{n}\sum_{i=1}^n p_i$, $\mu = \mathbb{E}p_i$. Then the single-object model-average NLL has the following form (suppose, without loss of generality, that the first class is the correct one):

$$\mathrm{NLL}_n^{\mathrm{obj}}(\tau) = -\mathbb{E}\log\left(\mathrm{softmax}\{(\log(\bar{p}_n))/\tau\}^*\right) = \mathbb{E}\left[-\gamma\log(\bar{p}_{n1}) + \log\left(\sum_{k=1}^K \bar{p}_{nk}^\gamma\right)\right], \quad (14)$$

---

[1] As $\delta < 1$, $\varepsilon_n \to 0$ monotonically and from some $n$ the condition $0 < \varepsilon_n < \mu$ is fulfilled.

where $\gamma = \frac{1}{\tau}$. By applying a similar Taylor expansions trick as in Proposition 1 to the function $f(p) = -\gamma \log(p_1) + \log\left(\sum_{k=1}^{K} p_k^\gamma\right)$ under expectation in (14), we arrive at:

$$\mathrm{NLL}_n^{\mathrm{obj}}(\tau) \approx f(\mu) + \frac{1}{2n} \sum_{k,k'=1}^{K} cov(p_{ik}, p_{ik'}) \left.\frac{\partial^2 f}{\partial p_k \partial p_{k'}}\right|_{p=\mu}, \tag{15}$$

where

$$\frac{\partial^2 f}{\partial p_k \partial p_{k'}} = \frac{\gamma}{p_1^2}[k=1][k'=1] - \frac{\gamma^2 p_k^{\gamma-1} p_{k'}^{\gamma-1}}{\left(\sum_{s=1}^{K} p_s^\gamma\right)^2} + [k=k']\frac{\gamma(\gamma-1)p_k^{\gamma-2}}{\sum_{s=1}^{K} p_s^\gamma}. \tag{16}$$

We can see that the first term in (16) is non-negative, while the second one is, on the contrary, always negative. High values of temperature (i. e. $\gamma < 1$) make the last term negative as well, which, in turn, may lead to negative $b$ coefficient in the power law of the right-hand part of (15). We observe this effect in practice: at certain values of temperature applied after averaging, the NLL starts *increasing* as a function of $n$, see Appendix C.2. We could not apply a similar technique, as was provided in section A.1, to derive the exact asymptote in (15), when the temperature is applied after averaging.

We note, however, that when the temperature $\tau$ is applied *before* averaging, i. e. fixed[2] $\tau$ is used to compute the final member networks predictions: $p_i(\tau) = \mathrm{softmax}\{\log(p_i)/\tau\}$, the expression for $\mathrm{NLL}_n^{\mathrm{obj}}(\tau)$ fits the form of (1):

$$\mathrm{NLL}_n^{\mathrm{obj}}(\tau) = -\mathbb{E}\log\left(\frac{1}{n}\sum_{i=1}^{n} p_i^*(\tau)\right), \tag{17}$$

and hence Proposition 1 is applicable. The power law parameter $b = \frac{\sigma^2}{2\mu^2}$ in this case is always positive.

### A.3 Lower envelope of power laws with continuous temperature

In this subsection, we show that even when the set of temperatures is uncountable, the lower envelope of power laws asymptotically follows the power law. This argument generalizes our discussion at the end of section 2.

**Proposition 2** *Consider a compact set $\mathcal{T}$ and two continuous mappings $c : \mathcal{T} \to \mathbb{R}$ and $b : \mathcal{T} \to \mathbb{R}$. Let each value $\tau \in \mathcal{T}$ correspond to a certain power law w. r. t. $n$:*

$$\mathrm{PL}_n(\tau) = c(\tau) + \frac{b(\tau)}{n}. \tag{18}$$

*Then the lower envelope $\mathrm{LE}_n = \min_{\tau \in \mathcal{T}} \mathrm{PL}_n(\tau)$ of the power laws (18) follows a power law asymptotically.*

**Proof.** Let $\tau_n \in \underset{\tau \in \mathcal{T}}{\mathrm{Argmin}}\, \mathrm{PL}_n(\tau)$ be the value which minimizes $\mathrm{PL}_n(\tau)$ at given $n$. Then the lower envelope of power laws (18) can be defined as

$$\mathrm{LE}_n = \mathrm{PL}_n(\tau_n). \tag{19}$$

We need to show that $\exists\, c^*, b^* \in \mathbb{R}$:

$$\mathrm{LE}_n = c^* + \frac{b^*}{n} + o\left(\frac{1}{n}\right). \tag{20}$$

By the definition of $\tau_n$, the following inequalities hold:

$$\begin{cases} c(\tau_n) + \frac{b(\tau_n)}{n} \leq c(\tau_{n+1}) + \frac{b(\tau_{n+1})}{n} \\ c(\tau_{n+1}) + \frac{b(\tau_{n+1})}{n+1} \leq c(\tau_n) + \frac{b(\tau_n)}{n+1} \end{cases} \implies \begin{cases} c(\tau_{n+1}) \leq c(\tau_n) \\ b(\tau_n) \leq b(\tau_{n+1}). \end{cases} \tag{21}$$

The first of the right inequalities in (21) can be obtained via multiplication of the left inequalities by $n$ and $n+1$, respectively, and summation. The second inequality is obtained after substituting the first one into the right-hand part of the first left inequality.

Now, as $\{\tau_n\} \subset \mathcal{T}$, which is a compact, there exists a subsequence $\tau_{n_k} \to \tau^* \in \mathcal{T}$ which implies that $b(\tau_n) \to b(\tau^*)$, $c(\tau_n) \to c(\tau^*)$ monotonically due to continuity of mappings $b(\tau)$, $c(\tau)$ and the right inequalities in (21). Finally, consider the difference between $\mathrm{LE}_n$ and $\mathrm{PL}_n(\tau^*)$:

$$0 \leq \mathrm{PL}_n(\tau^*) - \mathrm{LE}_n = c(\tau^*) - c(\tau_n) + \frac{b(\tau^*) - b(\tau_n)}{n} \leq \frac{b(\tau^*) - b(\tau_n)}{n}. \tag{22}$$

As $b(\tau^*) - b(\tau_n) = o(1)$, we come to (20) with $c^* = c(\tau^*)$, $b^* = b(\tau^*)$. ∎

As was shown in section 2, $\mathrm{NLL}_n(\tau)$ follows a power law asymptotically for any fixed $\tau > 0$. Due to continuity of a softmax function w. r. t. $\tau$, we could deduce that the parameters $b$ and $c$ of the respective power law are also continuous functions of temperature. Finally, appropriately choosing the temperatures set $\mathcal{T}$ (namely, separate from zero — the singularity point) allows to conclude that deviation of $\mathrm{NLL}_n(\tau)$ from its power law is $o(\frac{1}{n})$ uniformly for all $\tau \in \mathcal{T}$ and hence Proposition 2 is applicable to the lower envelope of $\mathrm{NLL}_n(\tau)$ as well.

---

[2]We use the same value of $\tau$ for all member networks.

# B   Experimental details

**Data.**   We conduct experiments on CIFAR-100 and CIFAR-10 datasets, each containing 50000 training and 10000 testing examples. For tuning hyperparameters, we randomly select 5000 training examples as a validation set. After choosing optimal hyperparameters, we retrain the models on the full training dataset. We use a standard data augmentation scheme: zero-padding with 4 pixels on each side, random cropping to produce $32 \times 32$ images, and horizontal mirroring with probability 0.5.

**Details.**   We consider two architectures: VGG and WideResNet. We use the implementation provided at `https://github.com/timgaripov/dnn-mode-connectivity`.To obtain networks of different sizes, we vary the width factor of the networks: for VGG / WideResNet, we use convolutional layers with $[w, 2w, 4w, 8w]$ / $[w, 2w, 4w]$ filters, and fully-connected layers with $8w$ / $4w$ neurons. For VGG / WideResNet, we consider $2 \leqslant w \leqslant 181$ / $5 \leqslant w \leqslant 453$; $w = 64$ / $160$ corresponds to a standard, commonly used, configuration with $s_{\text{standard}}$ = 15.3M / 36.8M parameters. These sizes are referred to as the standard budgets. For VGG, we use weight decay, and binary dropout for fully-connected layers. For WideResNet, we use weight decay and batch normalization. For each considered width factor, we tune the hyperparameters using grid search with the following grids: learning rate — $\{0.005, 0.05\}$ / $\{0.01, 0.1\}$ for VGG / WideResNet, weight decay — $\{10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}\}$, dropout rate — $\{0, 0.25, 0.5\}$. We train all models for 200 epochs using SGD with momentum of 0.9 and the following learning rate schedule: constant (100 epochs) – linearly annealing (80 epochs) – constant (20 epochs). The final learning rate is 100 times smaller than the initial one. We use the batch size of 128.

In several experiments presented in the Appendix, we train networks without regularization, with all hyperparameters being the same for all network sizes. By this, we mean that we set weight decay and dropout rate to zero, do not use data augmentation, and use an initial learning rate 10 times smaller than in the reference implementation to ensure that the training converges for all considered models.

**Computing infrastructure.**   VGG networks were trained on NVIDIA Teals P100 GPU. Training one network of the standard size / smallest considered size / largest considered size took approximately 1 hour / 20 minutes / 4.5 hours. WideResNet networks were trained on NVIDIA Tesla V100 GPU. Training one network of the standard size / smallest considered size / largest considered size took approximately 5.5 hours / 50 minutes / 32 hours.

**Approximating sequences with power laws.**   To approximate a sequence $\hat{y}_m$, $m \geqslant 1$ with a power law, we solve optimization task (6) using BFGS and performing computations with the double-precision floating point numbers (float64) to avoid overflow when computing the logarithms of nearly zero values. If we consider a uniform grid for $m$ in the linear scale, then in the logarithmic scale the density of the points in the area with large $m$ is much higher than in the area with small $m$. To account for this variable density, we weight terms corresponding to different values of $m$ in (6): $w_m = \frac{1}{m}$. When we report the quality of approximation measured with RMSE, we use uniform weights $w_m = \frac{1}{M}$. The described heuristic is utilized for approximating $\widehat{(\text{C})\text{NLL}}_n$, since we have a uniform grid for $n$ in the linear scale, and is not utilized for approximating $\widehat{\text{CNLL}}_s$ and $\widehat{\text{CNLL}}_B$, since for these sequences we have a uniform grid for $s$ / $B$ in the logarithmic scale.

**Test-time cross-validation.**   Ashukha et al. [2] utilize a so-called *test-time cross-validation* to obtain an unbiased, low-variance estimate of the CNLL using the publicly available test set. The test-time cross-validation implies that the test set is randomly split into two equal parts five times. For each split, one part is used to find the optimal temperature, and the other one — to measure the CNLL, and vice versa. Finally, the CNLL is averaged over ten measurements.

# C   Calibration of ensembles: applying temperature before or after averaging

## C.1   Difference in CNLL

In section 2, we introduced two ways of applying temperature to the ensemble, namely before and after averaging. In this subsection, we empirically compare these two calibration procedures. Figure 7 shows the results for VGG on CIFAR-100, for setting with and without regularization. The difference between CNLL values is low in all the cases, hence the two procedures perform similarly. In most of the cases, particularly in practically important cases of ensembling networks of medium and large sizes, calibration with applying temperature before averaging performs slightly better (there are a lot of green pixels in the heatmaps). The results for other dataset-architecture pairs are similar. To sum up, the procedure with applying temperature before averaging can be used in practice instead of the standard one, with temperature applied after averaging, without loss in the quality.
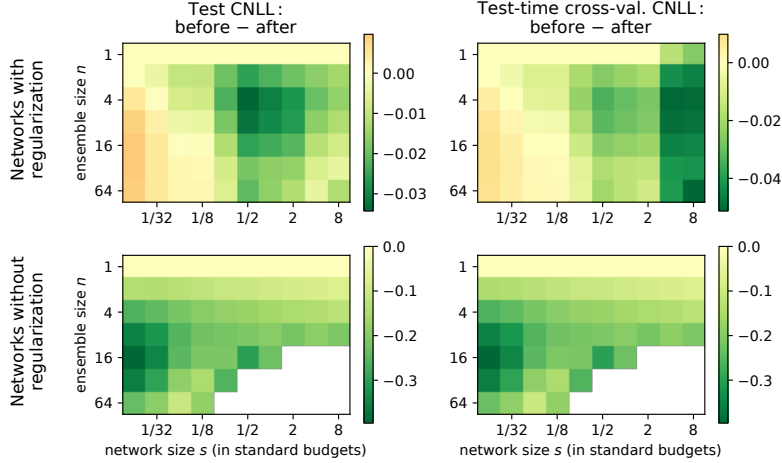
Figure 7: Difference between CNLLs computed with applying temperature *before* and *after* averaging for different values of ensemble size $n$ and network size $s$, for VGG on CIFAR-100. Left: the optimal temperature for CNLL is chosen using the whole test set. Right: CNLL is computed using test-time cross-validation.
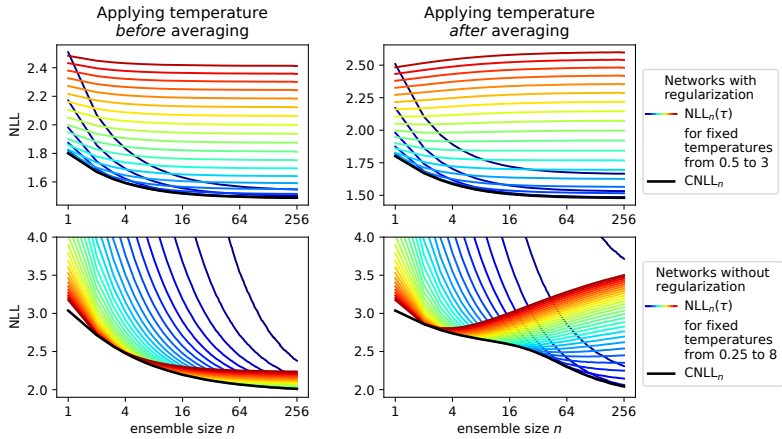


Figure 8: $\mathrm{NLL}_n(\tau)$ for different values of $\tau$ and $\mathrm{CNLL}_n$. VGG on CIFAR-100, the smallest considered network size (1/64 of standard budget). The blue / red color corresponds to the low / high temperatures.

## C.2   Dynamics of NLL with fixed temperature and CNLL

In this subsection, we illustrate the behaviour of $\mathrm{NLL}_n(\tau)$ and $\mathrm{CNLL}_n$ for both ways of applying temperature. Figure 8 shows the results for VGG on CIFAR-100, for setting with and without regularization, for the smallest considered network size. When the temperature is applied before averaging, in practice, both $\mathrm{NLL}_n(\tau)$ and $\mathrm{CNLL}_n$ follow a power law as functions of $n$, with positive parameter $b$, as can be seen from figure 8, left column. When the temperature is applied after averaging, $\mathrm{NLL}_n(\tau)$ starts growing w.r.t. $n$ for high values of $\tau$, as can be seen from figure 8, right. As a result, $\mathrm{CNLL}_n$ in this case may be non-convex, see the bottom right plot. We observed the effect of $\mathrm{NLL}_n(\tau)$ increase in a wide range of settings, while the effect of $\mathrm{CNLL}_n$ non-convexity was observed only for small unregularized networks. In the most cases, $\mathrm{CNLL}_n$ with temperature applied after averaging also follows a power law.

## D   Comparison of CNLL and LE-NLL

In this section, we empirically show that moving the minimum operation outside the expectation in equation (3) does not change the value of CNLL a lot in practice. In other words, we compare the values of CNLL (3), commonly used in practice, and LE-NLL (4), utilized in section 2, for different dataset–architecture pairs. We
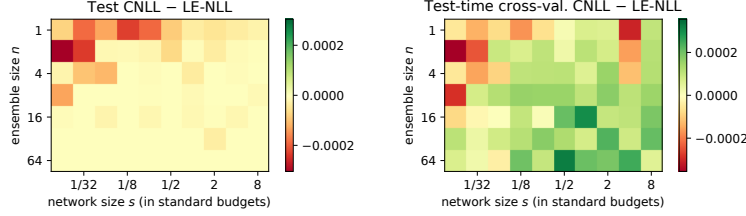
Figure 9: Difference between CNLL (3) and LE-NLL (4) for different values of ensemble size $n$ and network size $s$, for VGG on CIFAR-100. Left: the optimal temperature for CNLL is chosen using the whole test set. Right: CNLL is computed using test-time cross-validation.
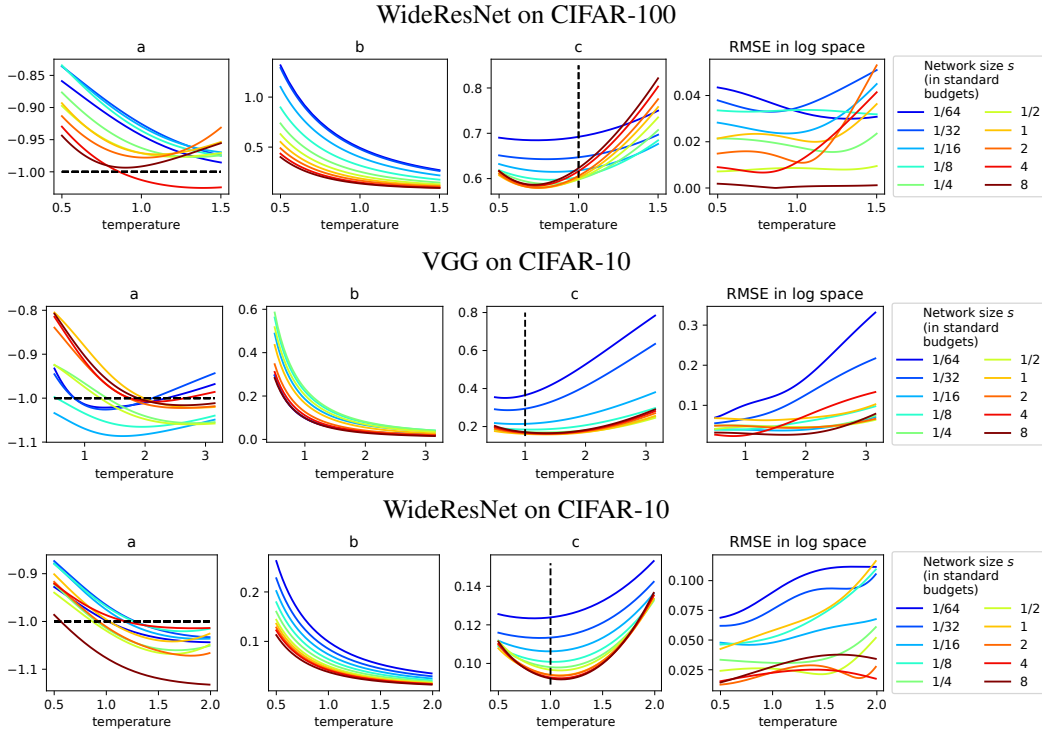


Figure 10: Parameters of power laws and the quality of approximation for $\text{NLL}_n(\tau)$ with fixed temperature $\tau$.

consider two scenarios for computing CNLL: (a) when the optimal temperature is chosen using the whole test set, as it is the case for LE-NLL, and (b) when the test-time cross-validation is utilized [2]. In figure 9 we depict the difference between CNLL and LE-NLL for different values of ensemble size $n$ and network size $s$, for VGG on CIFAR-100. We observe that the difference is negligible, compared to the values of LE-NLL. The relative difference for all values of $n$ and $s$ is bounded by $0.018\%/0.29\%$ for scenarios (a) and (b) respectively. For WideResNet on CIFAR-100, the relative difference is bounded by $0.081\%/0.61\%$, for VGG on CIFAR-10 — $0.054\%/0.46\%$, for WideResNet on CIFAR-10 — $0.023\%/0.45\%$ for scenarios (a) and (b) respectively. In all the the experiments in the paper, we use CNLL computed using test-time cross-validation.
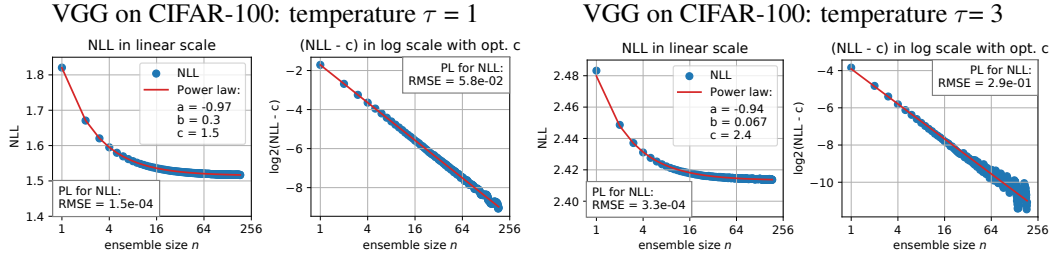
16

Figure 11: Power law approximation for $\mathrm{NLL}_n(\tau)$ for VGG of small size (1/64 of standard budget) on CIFAR-100, with the standard and high values of temperature. Approximations in both linear space and log-space are presented.

# E    NLL with fixed temperature as a function of ensemble size

## E.1    Power law approximation for different dataset–architecture pairs

In this subsection, we show that $\mathrm{NLL}_n(\tau)$ can be closely approximated with a power law as a function of $n$, on the whole considered range $n \geqslant 1$, for all values of $\tau > 0$, and for different dataset–architecture pairs. Figure 10 supplements figure 2 for other dataset–architecture pairs, and shows the quality of approximation of $\mathrm{NLL}_n(\tau)$ with power laws as well as the dynamics of power law parameters $a$, $b$, $c$.

For a particular network size, parameter $a$ is generally a decreasing function of temperature $\tau$. The described effect does not hold for the ensembles of large WideResNet networks, because for them we only consider small ensembles due to the resource limitations, so the approximation is performed for only a few points and is slightly unstable. The described effect also does not hold for the ensembles of small VGG networks, since applying high temperatures leads to the noise in $\mathrm{NLL}_n(\tau)$ for large $n$ and makes the approximation slightly worse than for other settings, as confirmed in the rightmost plot. This approximation is still very close, see appendix E.2.

Parameter $b$ decreases when the temperature grows, since lower temperatures result in more contrast predictions, and ensembling smooths them. Parameter $c$ is a non-monotonic function of $\tau$, and its optimum reflects the optimal temperature for the "infinite"-size ensemble. The optimal temperature may be greater or less than one, depending on the dataset–architecture combination.

## E.2    Power law approximation for small networks with high temperature

Figure 11 visualizes the power law approximation of $\mathrm{NLL}_n(\tau)$ for VGG of the smallest considered width, with a standard temperature $\tau = 1$ and a high temperature $\tau = 3$. In log-space, we observe the visible fluctuations of $\mathrm{NLL}_n(\tau = 3)$ for large ensembles. These fluctuations result in a small bias in the estimation of power law parameters, particularly parameter $a$. However, in the linear space, the fluctuations do not have a significant effect, and the approximation is close to the data.

The reason for the fluctuations is that for high values of temperature and for large ensemble sizes $n$, the difference $\mathrm{NLL}_n(\tau) - c$ for optimal $c$ is extremely small, while the precision of the computation is limited. Particularly, the minimum value of $\log_2(\mathrm{NLL}_n(\tau) - c)$ is approximately equal to $-11$ for $\tau = 3$ and to $-9$ for $\tau = 1$, see the y-axes of figure 11. We observe the fluctuations for the ensembles of small networks, since for small $s$ we consider the largest values of $n$.

# F    Convergence of temperature

To show that the optimal temperature $\tau$ converges when ensemble size $n$ increases, we plot optimal $\tau$ vs. $n$ for different dataset–architecture pairs in figure 12. We average the optimal temperature over runs, i.e. different trained ensembles, and over folds in test-time cross-validation.

# G    Power law approximation of CNLL as a function of ensemble size

Figure 13 supplements figure 3, to show that $\mathrm{CNLL}_n$ with the temperature applied before averaging can be closely approximated with a power law on the whole considered range $n \geqslant 1$, for all considered dataset–architecture pairs. The rightmost plot reports the quality of approximation in the log-space. We notice that in the linear space, the RMSE is less than $5.9 \cdot 10^{-4}$ for all blue points, corresponding to VGG, and less $4.7 \cdot 10^{-4}$ for
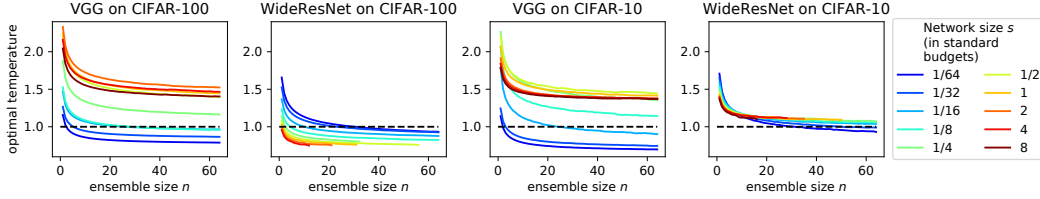
Figure 12: Optimal temperatures for different dataset–architecture pairs.
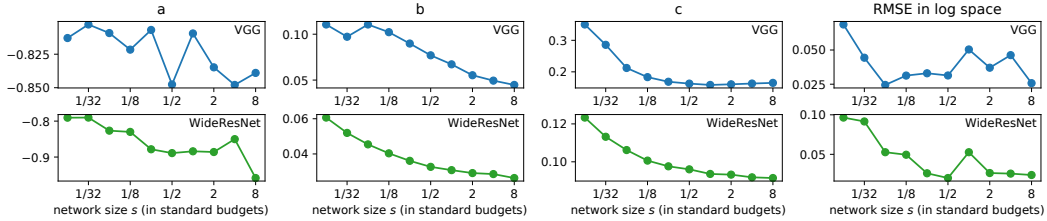


Figure 13: Parameters of power laws and the quality of approximation for $\mathrm{CNLL}_n$ for different network sizes $s$. VGG and WideResNet on CIFAR-10.

all green points, corresponding to WideResNet. The first three plots in each row visualize the dynamics of power law parameters $a$, $b$, $c$ as network size increases; these dynamics were discussed in section 4.

## H  CNLL of the ensemble of unregularized networks

In this section, we analyse the CNLL of the ensembles of unregularized networks as a function of network size. Turning off regularization, i.e. weight decay and dropout, results in the clearly observed effect that for large values of $n$, $\mathrm{CNLL}_s$ of the ensemble of size $n$ has optimum at some network size $s$. We firstly visualize this effect in figure 14 showing the $(n, s)$ plane of CNLL for the setting without regularization, for VGG and WideResNet on CIFAR-100. For higher $n$ (e.g. $n > 5$), the horizontal cuts are non-monotonic w.r.t. $s$. Secondly, we analyse the CNLL of the "infinite"-size ensemble, using power laws discovered in the paper. Figure 15 visualizes the parameters of the power laws approximating $\mathrm{CNLL}_n$ for different values of $s$, for VGG and WideresNet on CIFAR-100. We can clearly observe that parameter $b$, that reflects the possible gain of ensembling the networks of size $s$, decreases as $s$ increases, while the parameter $c$, that reflects the CNLL of the "infinite"-size ensemble, is non-monotonic, i.e. achieves optimum at some $s$ in the middle of the considered range of $s$. The similar effect was observed in figure 3 for the regularized setting, but in a less visible form. We notice that the decrease of parameter $b$ in the setting without regularization is much larger, than in the standard setting with regularization, and the effect of non-monotonicity of $c$ is also stronger.

The described effect of the non-monotonicity of $\mathrm{CNLL}_s$ for large ensembles may be a consequence of under-regularization of the large networks, or a consequence of the decreased diversity of the large networks [23], and needs further investigation.
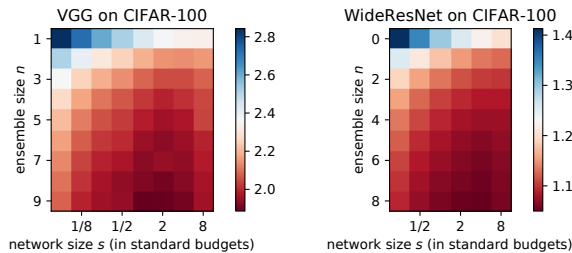


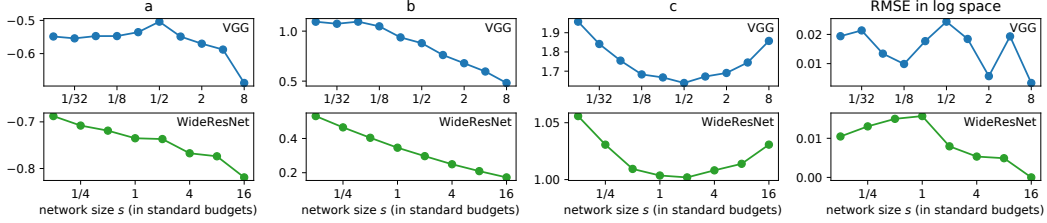Figure 14: The $(n, s)$-plane of CNLL for WideResNet and VGG on CIFAR-100 for the setting without regularization.

Figure 15: Parameters of power laws and the quality of approximation for $\text{CNLL}_n$ for different network sizes $s$. VGG and WideResNet on CIFAR-100, setting without regularization.
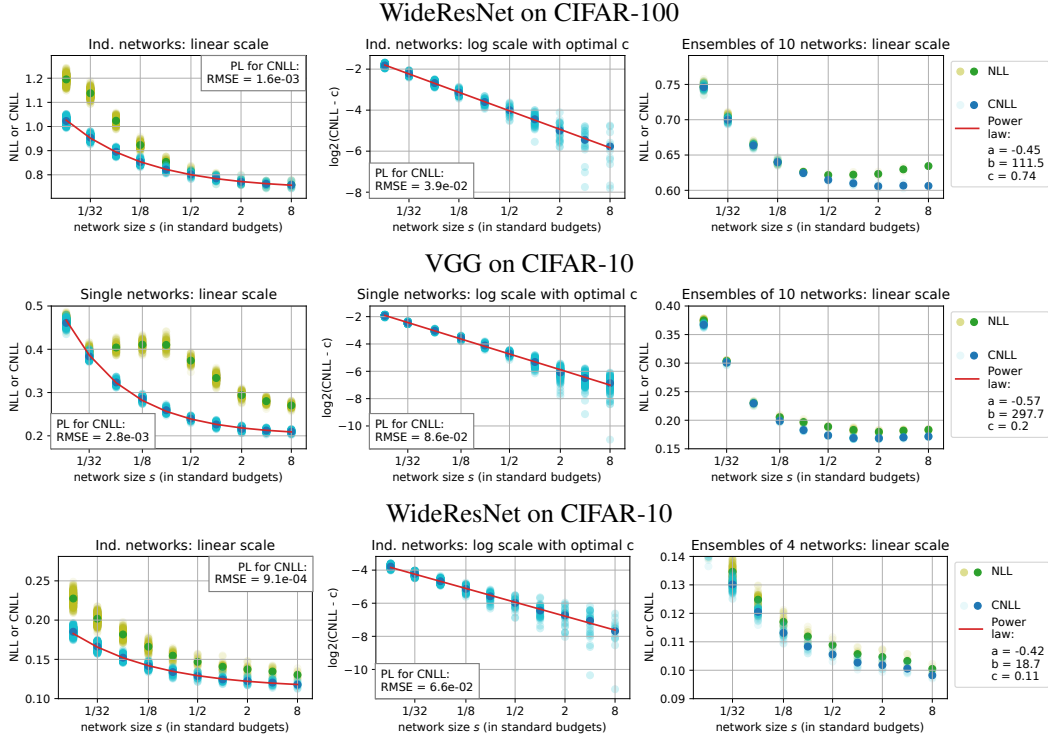


Figure 16: Non-calibrated $\text{NLL}_s$ and $\text{CNLL}_s$. Left and middle: for a single network, $\text{NLL}_s$ may exhibit double descent, while $\text{CNLL}_s$ can be closely approximated with a power law. Right: $\text{NLL}_s$ and $\text{CNLL}_s$ of an ensemble of several networks may be non-monotonic functions.

## I    Power law approximation of NLL as a function of network size

Figure 16 supplements figure 4, to show that $\text{CNLL}_s$ can be closely approximated with a power law on the whole considered range of network sizes $s$, for all considered dataset–architecture pairs, with parameter $a$ close to $-0.5$. For the non-calibrated NLL of VGG on CIFAR-10, we observe the same double descend behaviour as for VGG on CIFAR-100.

## J    Power law approximation of NLL as a function of the memory budget. MSA effect

Figure 17 supplements figure 5, to show that $\text{CNLL}_B$ can be closely approximated with a power law on the whole considered range of memory budgets $B$, for all considered dataset–architecture pairs. The right column of the plots visualizes the MSA effect: for each memory budget $B$ (each line), the optimum of CNLL is achieved at abscissa $n > 1$. The MSA effect also holds for accuracy for a wide range of budgets, including budgets less than the standard one, see figure 18. The optimal memory split is usually achieved for accuracy at the same $n$ or at the smaller $n$ than for CNLL.
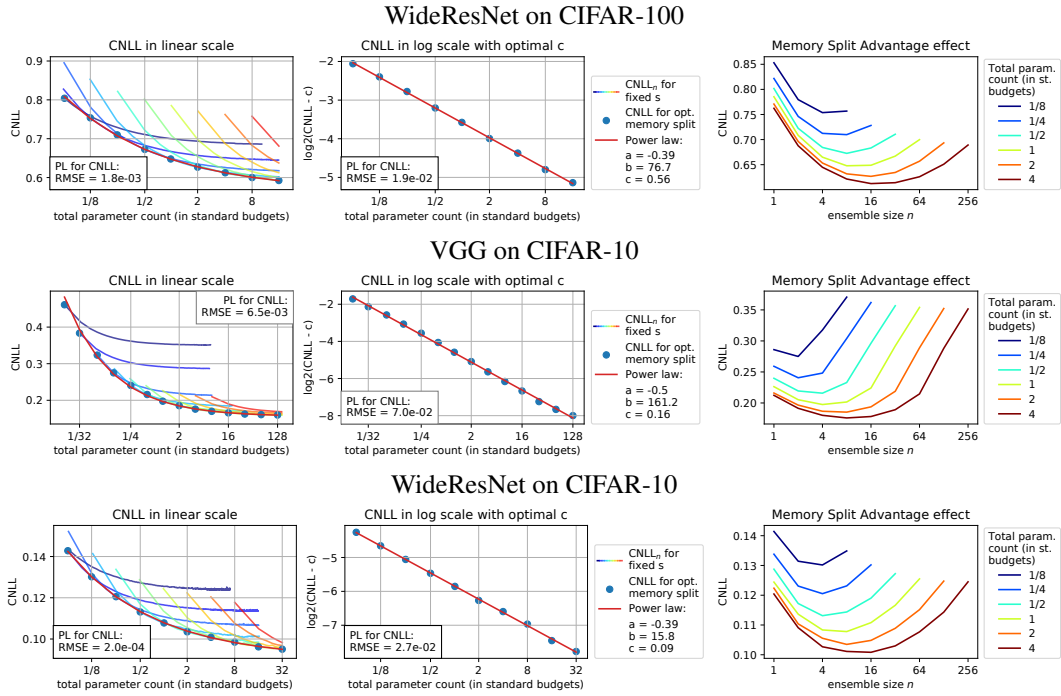
Figure 17: Left and middle: $CNLL_B$ for different dataset–architecture pairs can be closely approximated with a power law. $CNLL_B$ is a lower envelope of $CNLL_n$ for different network sizes $s$. Right: MSA effect: for different memory budgets $B$, the optimal CNLL is achieved at $n > 1$.
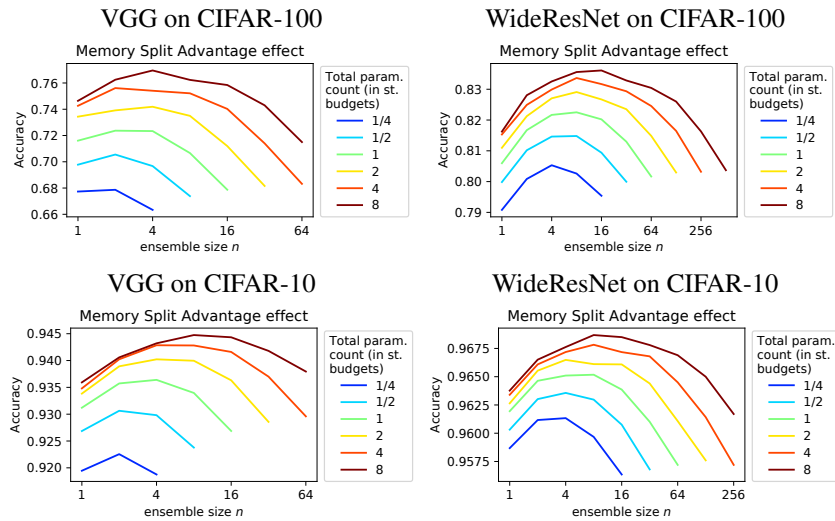


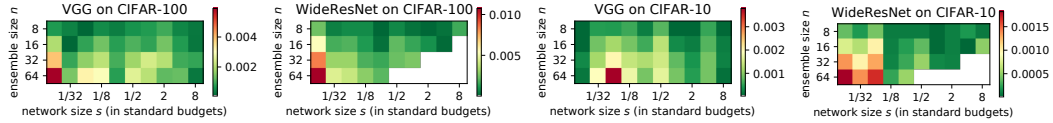Figure 18: MSA effect for accuracy: for different memory budgets $B$, the optimal accuracy is achieved at $n > 1$.

Figure 19: Predictions based on $\mathrm{CNLL}_n$ power laws. Difference between true and predicted CNLL. Predictions are made for large $n$ based on $n = 1..4$ using all trained networks of each size, i. e. with averaging $\mathrm{CNLL}_n$ for $n = 1..4$ over a large number of runs.
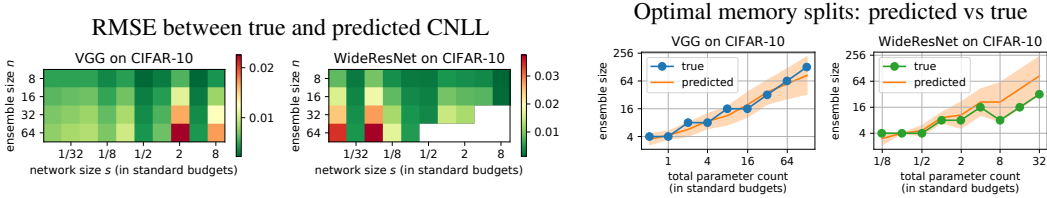


Figure 20: Predictions based on $\mathrm{CNLL}_n$ power laws for VGG and WideResNet on CIFAR-10. Predictions are made for large $n$ based on $n = 1..4$ using 6 trained networks of each size. Left pair: RMSE between true and predicted CNLL. Right pair: predicted optimal memory splits vs true ones. Mean $\pm$ standard deviation is shown for predictions.

**The timing analysis of the memory splitting procedure.** One might wonder, how much slower is training and prediction with the ensemble of several medium-size networks compared to a single large network. In table 1, we list the training and prediction time for a single VGG on CIFAR-100, for different network sizes. We conduct this experiment on GPU conducting training / prediction with several networks sequentially, using the training batch size of 64 and the testing batch size of 1024. We observe that using the memory split with the relatively small number of networks (which is the case in practice) is only moderately slower than using a single wide network. 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. Please note that these numbers are given for the case when the networks in the memory split are trained / validated *sequentially*, while the memory split allows *parallel* training / validation.

| Network size (in standard budgets) | 0.125 | 0.25 | 0.5 | 1 | 2 | 4 | 8 |
|---|---|---|---|---|---|---|---|
| Time of 1 training epoch | 7.8 | 8 | 11 | 16 | 27 | 42 | 80 |
| Prediction time | 6.8 | 9.5 | 20 | 33 | 63 | 111 | 227 |

Table 1: Training and prediction time for a single VGG of different sizes on CIFAR-100.

# K    Predictions based on power laws

In this section, we expand the discussion on using the power laws observed in the paper for predicting the CNLL of large ensembles, and optimal memory splits for different memory budgets. For all predictions, we use the values of $\mathrm{CNLL}_n$ for $n = 1..4$ for different values of $s$ as given data.

We firstly conduct an experiment in which $\mathrm{CNLL}_n$ for $n = 1..4$ is averaged over all available $\lfloor \frac{\ell}{n} \rfloor$ runs, see section 3 for details on the number of available runs. In this experiment, we predict $\mathrm{CNLL}_n$ for $n > 4$. Figure 19 reports the RMSE between the true and predicted $\mathrm{CNLL}_n$. The predictions are highly precise with the error 2–3 orders smaller than the values being predicted. This experiment provides an evidence that the power laws discussed in the paper can be used for prediction. However, the described setting is not practically applicable due to the use of a large number of networks.

The second experiment is practically oriented: we use only 6 networks of each network size $s$, and average $\mathrm{CNLL}_n$ for $n = 1..4$ using only these 6 networks. The number 6 is chosen to provide more stable CNLL estimates for $n = 1..3$. We average the errors in predictions over 10 / 5 runs for VGG / WideResNet. Figure 20 supplements figure 6 and shows the results for CIFAR-10. When predicting $\mathrm{CNLL}_n$ for $n > 4$ (see the left pairs of plots in figures 6 and 20), we obtain the error of 1-2 orders smaller than the values being predicted. The

---
**Algorithm 1** Finding optimal memory split using power laws
---
**Input:** budget $B$ (the number of parameters);
**Output:** the optimal memory split of budget $B$;
 1: $\text{CNLL}_* = +\infty, \quad n_* = \text{None}; \quad k = -1;$
 2: **repeat**
 3:     $k \mathrel{+}= 1, n = 2^k;$
 4:     Train $\min(n, 6)$ networks of size $B/n$;
 5:     **if** $n > 4$ **then**
 6:         Fit the parameters of power law on sequence $\{\widehat{\text{CNLL}}_i\}_{i=1}^4$;
 7:         Predict $\text{CNLL}_n$ using power law;
 8:     **else**
 9:         Compute $\text{CNLL}_n$ using trained networks;
10:     **if** $\text{CNLL}_n < \text{CNLL}_*$ **then**
11:         $\text{CNLL}_* = \text{CNLL}_n; \quad n_* = n;$
12:         found $=$ False; // updated minimum
13:     **else**
14:         found $=$ True; // passed minimum
15: **until** found
16: Train the ensemble of $n_*$ networks of size $B/n_*$.
---

accurate prediction of $\text{CNLL}_n$ allows predicting the optimal memory splits for different memory budgets $B$, see the right pairs of plots in figures 6 and 20.

**Finding optimal memory splits in practice.** Let's say we have a budget $B$ and want to find an optimal memory split (MS). Algorithm 1 describes the procedure of finding the optimal MS using the discovered power laws. With this algorithm, we need to train $\min(n, 6)$ networks of size $B/n$ for $n = 1, \ldots, n^* + 1$ where $n^*$ is a number of networks in an optimal MS, and after finding $n^*$, we also need to train lacking networks for the optimal MS. 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 $B/2$, etc.) while the quality of the MS starts to degrade. In this case we need to train $n$ networks of size $B/n$ for $n = 1, \ldots, n^* + 1$. As a result, if $n^* \geqslant 4$, power-law predictions allow training fewer networks, and the higher $n^*$ the higher the gain.

# L    Additional experiments with ImageNet

Ashukha et al. [2] released the weights of the ensembles of ResNet50 networks (commonly used size) trained on ImageNet. We used their data to empirically confirm the power-law behaviour of NLL and CNLL as functions of the ensemble size. Figure 21 shows that the resulting power law approximations fit the data well, supporting the results presented in section 4.
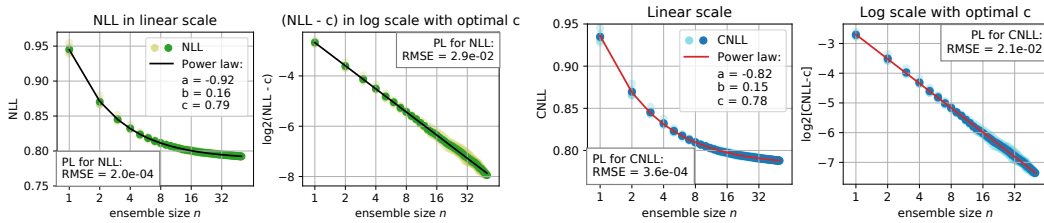


Figure 21: Non-calibrated NLL and CNLL of the ResNet50 of the commonly used size on ImageNet. Both $\text{NLL}_n$ and $\text{CNLL}_n$ can be closely approximated with a power law.