Initial Guessing Bias: How Untrained Networks Favor Some Classes

Emanuele Francazi [1 2] **Aurelien Lucchi** [3] **Marco Baity-Jesi** [2]

Abstract

Understanding and controlling biasing effects in neural networks is crucial for ensuring accurate and fair model performance. In the context of classification problems, we derive a theoretical analysis demonstrating that the structure of a deep neural network (DNN) can condition the model to assign all predictions to the same class, even before the beginning of training, and in the absence of explicit biases. We prove that, besides dataset properties, the presence of this phenomenon, which we call *Initial Guessing Bias* (IGB), is influenced by model choices including dataset preprocessing methods, and architectural decisions, such as activation functions, max-pooling layers, and network depth. Our analysis of IGB provides valuable information for selecting network architectures and initializing models. We also highlight theoretical consequences, such as a breakdown of node-permutation symmetry, a violation of self-averaging and non-trivial effects that depth has on IGB.

1. Introduction

In the field of deep learning, the art of designing neural networks often traverses a terrain where empirical practices overshadow theoretical foundations. The design of DNNs involves a complex array of decisions, each of which can significantly influence the network's performance and learning dynamics (Liu et al., 2017; Khanday & Sofi, 2021). Choices such as data standardization, activation functions, and initial weight configurations, pivotal for network performance, are typically guided by heuristic methods due to limited theoretical insights. A deeper theoretical understanding is crucial for developing more predictable and robust models. Additionally, the ethical and fairness implications of biased model predictions have become crucial in responsible machine learning development (Fuchs, 2018; Parraga et al., 2022; Siddique et al., 2023; Louppe et al., 2017).

In this paper, we study how different choices in architecture design and data pre-processing influence the predictions of neural networks at initialization. This leads us to the discovery of a previously unexplored phenomenon, which we illustrate in Fig. 1, where ***the initial predictions made by untrained neural networks are biased. We name this phenomenon Initial Guessing Bias (IGB).*** IGB challenges naive assumptions about DNNs and informs crucial design decisions. Our study elucidates how, beyond effects that may be induced by the structure of the data itself, the design of the model plays a crucial role in shaping the initial predictive bias of the model. Specifically, our work makes the following contributions:

- **Identification and formalization of IGB:** We are the first to observe and formally articulate the concept of IGB. Its relevance lies in:
  - Showing that a model can be biased toward specific predictions, before it even saw the data it will be trained on.
  - Guiding critical design choices in terms of architecture, initialization, and data standardization.
  - Revealing a symmetry breaking and a violation of self-averaging, which are common working hypotheses.
  - Influencing the initial phase of learning dynamics, whose behaviour is affected by the level of IGB.
- **Demonstration of IGB's robustness:** Through a comprehensive analysis, we establish the robustness of IGB across various settings:
  - We demonstrate the dependence of IGB on activation function choices and outline general rules for identifying IGB-inducing functions.
  - We uncover the role of max pooling in generating and intensifying IGB.
  - We show how data preprocessing procedures are crucial in activating and amplifying IGB.

[1]Physics Department, EPFL, Switzerland [2]SIAM Department, Eawag, Switzerland [3]Department of Mathematics and Computer Science, University of Basel, Switzerland. Correspondence to: Emanuele Francazi <emanuele.francazi@epfl.ch>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

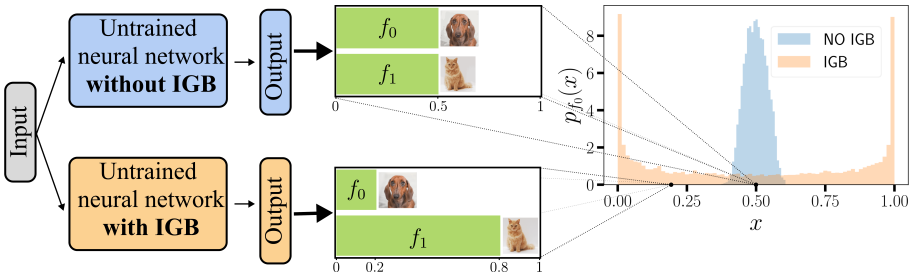

Figure 1. Initial Guessing Bias (IGB). Consider a task where we classify a binary dataset using an untrained network. Does it assign half of the examples to each class, or does it privilege one class? The answer depends on the model design. In the top-left, we classify a binary dataset with an untrained network *without IGB*. This model will generally assign half of the examples to each class (histogram on the top-center). In the bottom-left, we classify the same dataset using an untrained network *with IGB*. In this case, most of the guesses will usually go to one of the two classes (histogram on the bottom-center). As an example, we take the `dog/cat` classes (label 0 / label 1) from CIFAR10, and pass them through an untrained CNN with 2 layers, each followed by pooling. The non-IGB model uses $\tanh$ activations and average pooling, the IGB model uses ReLU and max pooling. We show in the center-right the distribution over different initializations, $p_{f_0}(x)$, of the fraction f_0 of times that each model guessed `dog` (equivalently, $f_1 = 1 - f_0$ indicates the fraction of images guessed as `cat`). While for the non-IGB models, f_0 is most often 50%, with IGB it most often is either 0% or 100%.

- We find that while network depth does not initiate IGB, it amplifies the bias when present.
- We develop a theory that analytically describes IGB in MLPs with random data, encompassing the settings mentioned above.
- We provide empirical evidence of the emergence of IGB in a broader range of practical scenarios, including real data, and a wide spectrum of architectures (e.g., CNNs, ResNets, Vision Transformers), demonstrating the prevalence of IGB.
- Besides the theoretical setting employing untrained architectures, IGB is also present in the context of transfer learning, where only the head of the model is randomly initialized while the backbone is pre-trained.

2. Related work

2.1. Bias effects

Bias in machine learning models is a multifaceted issue (Mehrabi et al., 2021). While the term 'bias' often carries negative connotations, particularly when it leads to performance degradation (Francazi et al., 2023; Engstrom et al., 2020) or fairness concerns (Torralba & Efros, 2011), it is important to recognize that not all forms of bias are inherently detrimental (Hagendorff & Fabi, 2023; Pot et al., 2021). When controlled, biases can be essential to the learning process of a good model. The key lies in the ability to regulate these effects to avoid adverse outcomes.

Most research in this domain has predominantly concentrated on biases stemming from dataset characteristics (Barocas & Selbst, 2016), such as class imbalance (Francazi et al., 2023; Ye et al., 2021; Panigrahi & Zhu, 2024), or arising due to specific algorithmic choices (Pessach & Shmueli, 2023). These forms of bias, while critical, represent only a part of the broader spectrum. The impact of biases resulting directly from model design decisions—such as network architecture, activation functions, and initialization strategies—remains comparatively unexplored.

Our work contributes to this less-charted territory by examining how certain design choices in DNNs can introduce biases at the very onset of the training process. This perspective not only broadens our understanding of bias in machine learning but also emphasizes the need for a more comprehensive approach to model development that considers the potential effects of every design decision.

2.2. Properties of DNNs at initialization

Recently, the study of DNNs at the initialization stage has garnered increasing attention, given that the network's initial state can significantly influence the training process. For example, the initial distribution of the weights can determine an amplification/decay of the signal coming from the input, or even limit the depth to which signals can propagate through random neural networks (Schoenholz et al., 2016; Hanin & Rolnick, 2018; Glorot & Bengio, 2010; Saxe et al., 2013; Orvieto et al., 2021; Noci et al., 2022). The initial state of a DNN also significantly impacts generalization performance (Ramasinghe et al., 2023). While the study of the initial state of neural networks has received increasing attention over the past few years, our own study focuses on the initial bias and therefore differs from past work in some important key aspects which we detail in App. B.

3. Preliminaries

Definition IGB We now articulate the concept of IGB and its practical implications. For clarity, we primarily focus on

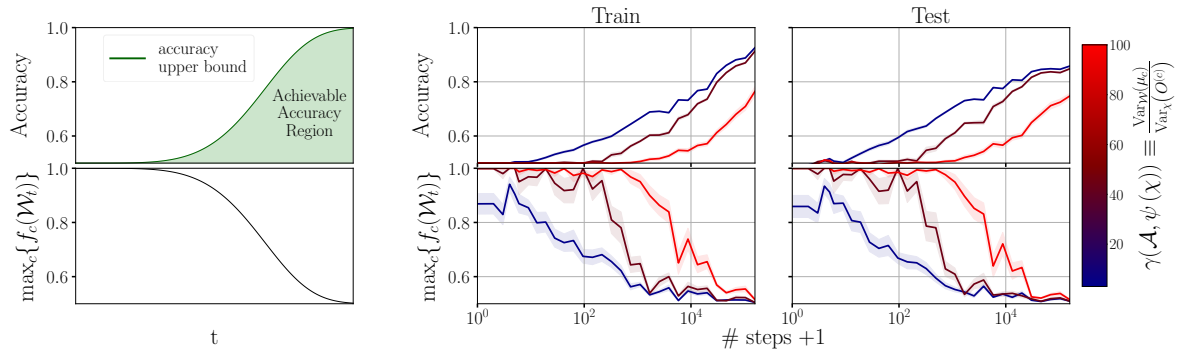

Figure 2. **Left**: IGB and Performance Bound: Diagram of the accessible performance range conditioned on the behavior of $\max_c\{f_c(\mathcal{W}_t)\}$ in a balanced binary dataset, in accordance with Eq. (1). **Right**: Comparison of the trend of $\max_c\{f_c(\mathcal{W}_t)\}$ with that of accuracy during the learning dynamics, varying with the level of guessing bias at initialization (IGB). Particularly, $\gamma(\mathcal{A},\psi(\chi)) \in \mathbb{R}^+$ (colormap on the right) provides a measure of the level of IGB: the higher the value of $\gamma(\mathcal{A},\psi(\chi))$, the higher the level of IGB (see Sec. 4 for more details). The curves show a consistent pattern with the diagram on the left. They also demonstrate that the time for IGB absorption increases with the level of IGB itself. The simulations were conducted on an MLP-mixer using a binary dataset (dogs *vs* cats from CIFAR) as input; more details on the setting and additional experiments with more architectures/datasets are provided in App. I.2.

binary classification (so $N_C = 2$ classes). Our findings also apply to multiclass classification, as detailed in App. H.

We start with an informal definition of the phenomenon of IGB. Consider a dataset χ and an architecture $\mathcal{A}$, parameterized by a set of weights. Specifically, we denote the configuration of the weight set at time t as $\mathcal{W}_t$, with $\mathcal{W} \equiv \mathcal{W}_0$. An observable that characterizes the IGB phenomenon is the fraction of points classified, by the untrained model, as class c, and denoted by $f_c(\mathcal{W})$.

Definition 3.1 (IGB, informal)**.** We say there is an Initial Guessing Bias if there is an imbalance in the initial fractions of points, captured by $f_c(\mathcal{W})$, assigned to a class.

Beyond initialization As we will show in this paper, IGB is a pervasive phenomenon emerging in untrained models. Therefore, a natural question arises: what are the consequences of IGB in terms of training and generalization? Fully addressing this question and determining whether IGB has beneficial or detrimental effects on network training is complex as the answer might vary with the problem (data, architecture, optimizer, choices of hyperparameter, and so on might all have confounding effects). A comprehensive study of this kind is beyond the scope of this work. However, we can start with a simple observation; a discrepancy between the fractions of guesses, $f_c(\mathcal{W}_t)$, and the actual class proportions in the dataset, sets a limit on the achievable performance. For instance, considering a balanced dataset with N_C classes, we can derive the following upper bound on the accuracy (Fig. 2 (left)):

$$\text{Accuracy}(t) \leq 1 - \left(\max_c\{f_c(\mathcal{W}_t)\} - \frac{1}{N_C}\right). \quad (1)$$

Therefore, in the presence of strong IGB in balanced datasets, it is necessary for IGB to be absorbed during the learning process for the model to produce effective predictions. Note however, that while in this specific case IGB slows down training, it might instead be beneficial in other situations, *e.g.* imbalanced training.

Sec. 5 will show how IGB can be activated and amplified in various ways, such as by acting on the architectural design or data pre-processing procedures. In Fig. 2 (right), simulations on an MLP-mixer (Tolstikhin et al., 2021) for various levels of IGB are presented. Specifically, to maintain a constant architecture for comparison, we adjusted the IGB level by modifying the dataset standardization (for more details see App. I.2). The experiment reveals that the time required for IGB absorption (bottom plots), and consequently the improvement in performance (upper plots), increases with the level of IGB. This increase in absorption time correlated with the level of IGB is consistently observed in further experiments on different architectures, as detailed in App. I.2.

Methods for controlling IGB These experiments highlight the importance of thoroughly understanding the phenomenon to formulate effective control strategies. The analysis we present offers several practical methods to regulate—either increase or decrease—IGB based on practitioners' needs. In particular, the main design choices outlined are:

- **The choice of activation function**: Thm. 5.1, and App. F analyse how the choice of the activation function determines the emergence of IGB. This offers practical guidelines on adapting activation functions to either mitigate or induce IGB. Detailed strategies are described in App.F.3 and exemplified in Fig.F.1, show-

ing that simply adding an offset to the ReLU function can significantly reduce IGB.

- **The choice of data standardization**: Thm. 5.2 describe how the choice of data pre-processing can be exploited to tune IGB. Beyond the theoretical derivation in App. G.1, we use these results in our experiments to adjust the level of IGB and compare different cases (see Fig. 2 Fig. I.4 and Fig. I.5).
- **The choice of pooling layer**: Thm. 5.2, and App. G.2 demonstrate how the max pool, depending on the chosen kernel size, can regulate the level of IGB, with larger kernel sizes amplifying the phenomenon.
- **The depth of the network**: Thm. 5.2, and App. G.3.1 illustrate that while increasing the network's depth does not cause IGB to emerge, it can amplify IGB if it is already present.

Similarly, many other design choices can be analyzed using the framework presented to determine control strategies for the level of IGB. For example, the choice of the temperature of the SoftMax function (see App. E.2.1 for more details).

4. Emergence of IGB: a first insight into the phenomenon

Setting and main notation Multi-Layer Perceptron (MLP) submodules are widely integrated into various architectures (He et al., 2016; Vaswani et al., 2017; Dosovitskiy et al., 2020). Moreover, recent advances, such as the MLP-mixer (Tolstikhin et al., 2021), highlight MLPs' capability to achieve competitive performance, suggesting their untapped potential, especially in large dataset applications. Our theoretical analysis centers on understanding the intricacies of MLP architectures. Formally, we consider an MLP as a set of D inputs $\{\boldsymbol{\xi}^{(a)}\}_{a=1}^{D}$ that propagate through a set of L hidden layers, until they reach the output layer, composed of a set of two nodes (one for each class), $\left\{O^{(c)}\right\}_{c=0,1}$ (see App. A for more details). Each input is classified by selecting the class c with the largest output value.

Our main notation is described in Fig. A.1–right, with data and weights modeled as follows:

- $\xi_b^{(a)} \sim \mathcal{N}(0,1)$ for each input component.
- $w_{ij}^{(l)} \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{N_{l-1}}\right)$ for the weights, with all biases set to 0 (see Eq. (21)). N_l indicates the number of nodes of the l^{th} hidden layer, while σ_w^2 is a constant that does not scale with the layer size (*e.g.* the gain value).

The choice of random unstructured data as input is common in the literature (Pennington & Worah, 2018; Koehler et al., 2021; Loureiro et al., 2021; Mignacco et al., 2020) as it simplifies theoretical analyses. In our case, this setting has the additional benefit of isolating the analysis from sources of IGB that are potentially embedded in the data structure. To support this assertion, we present empirical results using real data in App. I.1. These results illustrate that correlations within the data can exacerbate the impact of IGB. We follow the common practice of initializing bias parameters to zero, as noted in seminal works (Glorot & Bengio, 2010; He et al., 2015). By considering unstructured data *i.i.d.* across different classes and DNNs with null biases, we create a highly symmetric setting, making the presence of predictive bias more counter-intuitive. Extending our analysis to include non-null bias parameters is straightforward and discussed in App. G.3.2. Similarly, App. D.4 discusses how the analysis can be extended to include datasets with classes that are not identically distributed.

We also note that the Gaussian initialization employed in our analysis is the standard Kaiming initialization (He et al., 2015) but the analysis can be adapted to other initialization schemes. More specifically, our analysis applies to any set of independent weights $\{w_{ij}^{(l)}\}$ drawn from a centered distribution with variance $\mathcal{O}(1/N_l)$.

Two kinds of averages In our analysis, the random variables (*r.v.*s) we consider are essentially functions of two independent sources of randomness: the dataset, χ, and the set of initialized weights, $\mathcal{W}$. We denote the cumulative distribution function (*c.d.f.*) of a *r.v.* X as $F_X(x)$ and its probability density function (*p.d.f.*) as $p_X(x)$. For variables dependent on both sources of randomness, we specify the active source in the notation. For instance, given a function of two independent sets of random variables, $X(\mathcal{W},\chi)$:

$$F_X^{(\chi)}(x) = \mathbb{P}\left(X < x|\mathcal{W}\right)\ ,\ p_X^{(\chi)}(x') = \frac{d}{dx}F_X^{(\chi)}(x)\Big|_{x=x'}, \tag{2}$$

represents the *c.d.f.* and *p.d.f.* of X for a fixed set of weights $\mathcal{W}$. As we need to average over both the dataset and the weights, we use the concise notations

$$\langle x\rangle \equiv \mathbb{E}_\chi\left(x \mid \mathcal{W}\right) \quad \text{and} \quad \bar{x} \equiv \mathbb{E}_{\mathcal{W}}\left(x\right),$$

to signify the expectations over these two distinct sources of randomness (refer to App. A for more details).

Node Symmetry Breaking: the foundation of IGB We study how untrained neural networks assign data points to different classes before they start learning. In the absence of explicit bias weights, our intuition might suggest an even partitioning between classes. Instead, we demonstrate that the model design choices can induce an imbalance effect, resulting in a large fraction of data points being classified as a single class. The key quantity in our analysis is the

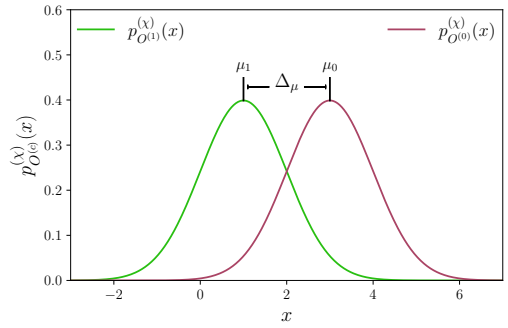

Figure 3. Illustration of the key quantities used in the analysis: 1) The green and purple curves represent the distributions of the two output nodes for a fixed set of network weights, $\mathcal{W}$, and 2) the mean of the distributions are denoted by μ_c.

distribution (over $\mathcal{W}$) of the fraction $f_c(\mathcal{W})$ of datapoints classified as class $c \in \{0, 1\}$. We have $f_c = 1/2$ in the absence of IGB, and a different value otherwise.

Since our model's guess is assigned to the class with the largest output value, $O^{(c)}$, in the limit of infinite datapoints, the *Law of large numbers* implies that

$$\lim_{D\to\infty} f_c(\mathcal{W}) = \mathbb{P}\left(O^{(c)} > O^{(1-c)} \mid \mathcal{W}\right) = \int_0^\infty p^{(\chi)}_{\Delta_{O^{(c)}}}(x)\, dx\,, \text{ with } \Delta_{O^{(c)}} \equiv O^{(c)} - O^{(1-c)}. \tag{3}$$

Without loss of generality, we will often use class 0 as a representative class, but the same discussion applies to any class. For class 0, Eq. (3) simplifies to

$$\lim_{D\to\infty} f_0(\mathcal{W}) = \mathbb{P}(\Delta_O > 0 \mid \mathcal{W}) = \int_0^\infty p^{(\chi)}_{\Delta_O}(x)\, dx\,, \tag{4}$$

with $\Delta_O \equiv O^{(0)} - O^{(1)}$. For the sake of compactness, we define $\mu_c \equiv \langle O^{(c)} \rangle$ and $\Delta_\mu \equiv \mu_0 - \mu_1$ as the difference between the distributions mean values. While f_c is a convenient and interpretable metric related to performance, our analysis can incorporate alternative measures that, for instance, allow for the analysis of the confidence level in the network's assignments, as shown in App. E.2.
Eq. (4) connects f_0 (*i.e.* the observable we are interested in) to the nodes variables $\{O^{(c)}\}$ (*i.e.* the set of variables we analyze through our investigation). The value of f_0 depends on how often the output related to class 0 has a higher value of the output than that of class 1. More precisely, Eq. (4) shows that f_0 is determined by comparing the output distributions $p^{(\chi)}_{O^{(c)}}(x)$ related to each class c. We illustrate this in Fig. 3. In the example of the figure, the output distribution related to class 0 is centered around higher values than that of class 1. Therefore, we will have $f_0 > f_1$, indicating the presence of IGB. In App. D.1 we show that, for MLPs, $p^{(\chi)}_{O^{(c)}}(x)$, is asymptotically a Gaussian whose center, μ_c, is itself a *r.v.*, which is drawn from a Normal distribution $p_{\mu_c}(x)$ that has a wide support:

Theorem 4.1 (Informal)**.** *Consider a Gaussian distributed dataset processed through an MLP with L hidden layers and weights initialized according to the Kaiming normal scheme (with null bias weights). In the limit of infinite width, the distribution of an output node $O^{(c)}$, at initialization, converges to:*

$$p^{(\chi)}_{O^{(c)}}(x) \xrightarrow{|\mathcal{W}|\to\infty} \mathcal{N}\left(x; \mu_c, Var_\chi\left(O^{(c)}\right)\right), \tag{5}$$

where $|\mathcal{W}|$ indicates the cardinality of the set $\mathcal{W}$ and, for compactness, we denoted by $|\mathcal{W}| \to \infty$ the limit where the number of neurons, N_l, of each hidden layer $l \in \{0, \dots, L\}$ tends to infinity.
Moreover, while the variance of the distribution, $Var_\chi\left(O^{(c)}\right)$, converges with high probability (w.h.p.) to a deterministic value, the center of this distribution, μ_c, is itself a r.v., varying from node to node and converging in distribution to:

$$p_{\mu_c}(x) \xrightarrow{|\mathcal{W}|\to\infty} \mathcal{N}(x; 0, Var_{\mathcal{W}}(\mu_c)). \tag{6}$$

In other words, the outputs of different classes are distributed according to *p.d.f.*s each centered on a different value: as μ_c are *r.v.*s, varying across output nodes, they are not all identically distributed. This difference results in a breakdown of permutation symmetry of nodes belonging to the same hidden layer (NSB). As we will explain now, this asymmetry is directly related to the emergence of IGB.
In fact, the study of IGB can be conceptually summarized as a comparison between the fluctuations of $p_{\mu_c}(x)$, which defines the distance between the Gaussians in Fig. 3, and those of $p^{(\chi)}_{O^{(c)}}(x)$, which define how wide each of these Gaussians is. We now illustrate the two extreme cases to underline our point. Starting from Eq. (3), we will discuss how the integral on the *r.h.s.* varies in these two scenarios:

- **Absence of IGB (Fig. 4 (left))**:
  If the fluctuations of $p_{\mu_c}(x)$ are much smaller than the ones of $p^{(\chi)}_{O^{(c)}}(x)$, we will have two Gaussian *r.v.*s centered almost on the same point, therefore, $\mathbb{P}\left(O^{(0)} > O^{(1)} \mid \mathcal{W}\right) = \mathbb{P}(\Delta_O > 0 \mid \mathcal{W}) \simeq 1/2$.
  Indeed, the difference between two Gaussian *r.v.*s is itself a Gaussian *r.v.*, centered at the difference between the mean values of the original distributions, Δ_μ. If the fluctuations of Δ_μ are much smaller than those of Δ_O, we will typically have that $p^{(\chi)}_{O^{(c)}}(x)$ is a symmetric distribution centered very close to the origin. Therefore $\mathbb{P}(\Delta_O > 0 \mid \mathcal{W}) \simeq 1/2$ *i.e.*, from Eq. (3), the fraction of points assigned to both classes is equal to 1/2.

- **Strong IGB (Fig. 4 (right))**:
  If, instead, the scale of $p_{\mu_c}(x)$ fluctuations is much bigger than that of $p^{(\chi)}_{O^{(c)}}(x)$ we will typically fall in the opposing scenario where the two Gaussian distributions, $p^{(\chi)}_{O^{(c)}}(x)$, are well separated. We can assume, without loss of generality, that $\mu_0 > \mu_1$. In this case we will have $\mathbb{P}\left(O^{(0)} > O^{(1)} \mid \mathcal{W}\right) = \mathbb{P}\left(\Delta_O > 0 \mid \mathcal{W}\right) \simeq 1$.

This difference between the last two scenarios suggests the following formal definition for IGB (we write it for a generic number N_C of classes).

Definition 4.2 (IGB, formal)**.** Given an architecture $\mathcal{A}$ and a preprocessed dataset $\psi(\chi)$, we have an absence of IGB if and only if, *w.h.p.*,

$$\lim_{D\to\infty} f_c(\mathcal{W}) = \frac{1}{N_C},\ \forall c \in \{0,\dots,N_C-1\}. \quad (7)$$

We instead have a presence of IGB if, in the limit of infinite data points ($D \to \infty$), we observe a disproportion between the values $\{f_c\}$ for different classes.

Note that, when D is finite, there can be finite-size fluctuations which move $f_c(\mathcal{W})$ from its asymptotic value.

Consistently with the intuition presented in Sec. 4, a possible measure of the level of IGB is given by the ratio of variances:

$$\gamma(\mathcal{A},\psi(\chi)) \equiv \frac{\mathrm{Var}_{\mathcal{W}}(\mu_c)}{\mathrm{Var}_{\chi}\left(O^{(c)}\right)}. \quad (8)$$

In the absence of IGB, $\lim_{D\to\infty}\gamma(\mathcal{A},\psi(\chi)) = 0$. The value of γ increases with the level of IGB, providing a quantifiable metric to assess the extent of imbalance in initial guess fractions across classes. In particular, we can use γ to describe the limit of large IGB.

Definition 4.3 (Strong IGB)**.** Given a setting $(\mathcal{A},\psi(\chi))$, a model exhibits *strong IGB* for such a setting if

$$\gamma(\mathcal{A},\psi(\chi)) = \infty. \quad (9)$$

Note that, Eq. (4) implies that, in the strong IGB limit,

$$\gamma(\mathcal{A},\psi(\chi)) = \infty \Longrightarrow p_{f_0}(x) = \tfrac{1}{2}\delta(x) + \tfrac{1}{2}\delta(x-1), \quad (10)$$

indicating that in each experiment, the dataset is completely classified as either belonging to class 0 or class 1.

5. IGB: emergence and amplification

By leveraging Theorem 4.1, we are able to theoretically estimate the *p.d.f.* of f_0, building upon the distributions defined in Eq. (5) and Eq. (6). These estimations, described in App. E.1 and App. G.3.1, enable us to systematically evaluate $p_{f_0}(x)$ across a spectrum of model designs.
The key steps in the derivation are:

(A) Deriving Eq. (5), which leads to the *p.d.f.* of Δ_O:

$$p^{(\chi)}_{\Delta_O}(x) = \mathcal{N}\left(x;\Delta_\mu, 2\mathrm{Var}_\chi\left(O^{(c)}\right)\right). \quad (11)$$

(B) Substituting Eq. (11) into Eq. (4) allows us to invert the equation linking f_0 to Δ_μ, thereby expressing Δ_μ as a function of f_0.

(C) Deriving Eq. (6), which leads to the *p.d.f.* of Δ_μ:

$$p_{\Delta_\mu}(y) = \mathcal{N}\left(y;0,2\mathrm{Var}_{\mathcal{W}}(\mu_c)\right). \quad (12)$$

(D) Starting from Eq. (12) and the relationship $\Delta_\mu(f_0)$ from point (B), we obtain $p_{f_0}(x)$ through a change of variables, that is, by applying the formula:

$$p_{f_0}(x)\,dx = p_{\Delta_\mu}(y(x))\,dy. \quad (13)$$

This analysis not only identifies the architectural elements critical to the emergence of IGB but also quantifies their influence.[1]

5.1. Single hidden layer

Let us begin by considering the case with $L = 1$; in this scenario, the element of non-linearity in the network (such as the activation function, pooling layer, etc.) can give rise to IGB. In our analysis we consider three different setups to highlight the different behaviors induced by non-linearity elements. Fig. 5 shows $p_{f_0}(x)$ (both empirical histogram and theoretical curves) for the following cases:

- **Linear**: $p_{f_0}(x)$ asymptotically converges to a delta distribution peaked on $f_0 = 1/2$.[2] The theoretical curve shown in the plot takes into account the finite dataset size effects[3] (since in real simulations $D < \infty$) as discussed in App. D.

[1] We note that, to derive $p_{f_0}(x)$, we work in the infinite-width limit. This is however not a necessary condition for the appearance of IGB (see *e.g.* experiments with finite-width models in App. I), but rather an assumption that allows us to derive the theoretical curves.

[2] By asymptotically, we mean in the $D \to \infty$ limit.

[3] To take finite dataset-size effects into account, we substitute the population data distribution, with the empirical one, defined over the finite set of D points.

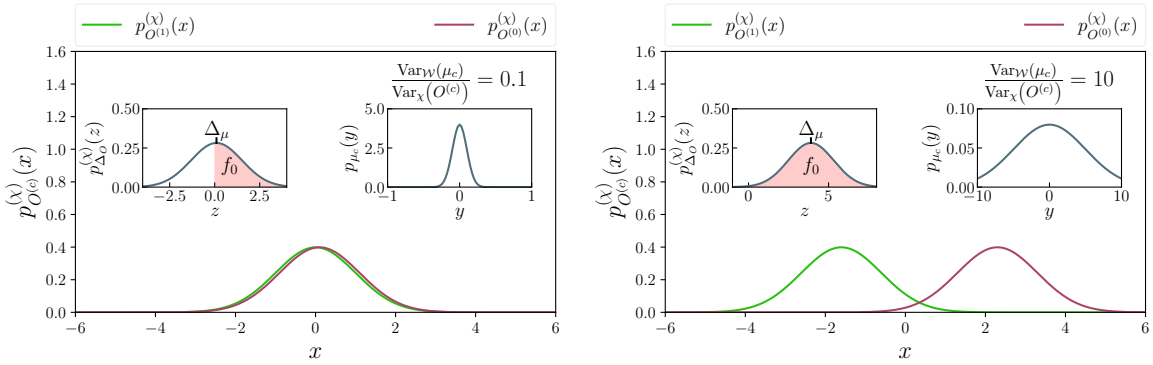

Figure 4. Comparison of two extreme scenarios: no IGB on the left, and strong IGB on the right. If the centers of the distributions, μ_c, have small fluctuations compared to the ones of the distributions $p^{(\chi)}_{O^{(c)}}(x)$, the two distributions almost completely overlap, resulting in a similar probability that one output node exceeds the other (left). If, instead, the centers are typically much further apart than the fluctuations scale of the distributions $p^{(\chi)}_{O^{(c)}}(x)$, the values drawn from one distribution exceed the other one with high probability (right). Each plot contains two inset plots. The inset plot in the upper left represents the distribution of the difference of the *r.v.*s shown in the main plot, (Δ_O). Note that, fixing the set $\mathcal{W}$ in a given experiment, and assuming a dataset big enough, (4) holds (the probability mass of the *r.h.s.* is depicted with a red area bounded by the distribution and the integration extremes). The inset plot in the upper right shows instead $p_{\mu_c}(x)$ to give an idea of the fluctuations of μ_c for the two cases, measured also by the variances ratio reported above the inset plot.

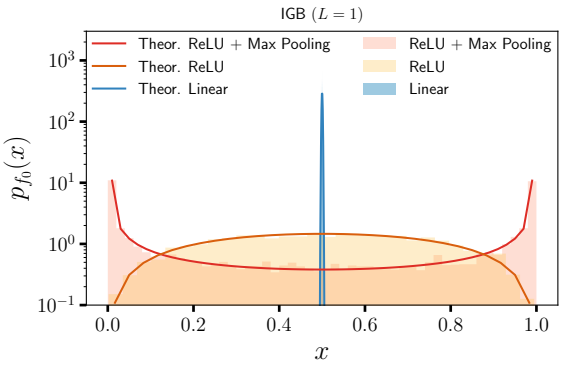

Figure 5. The distribution $p_{f_0}(x)$ in a single-hidden-layer perceptron, for different choices of activation functions and with/without max pooling.

- **ReLU**: In this case, $p_{f_0}(x)$ does not concentrate at $f_0 = 1/2$ and stays asymptotically wide (we detail this in App. D, Fig. D.1), so f_0 will, *w.h.p.*, be away from 1/2. The mode of the distribution remains, as for the linear case, at $f_0 = 1/2$. Yet, in this case, the fluctuations from the peak are not due to finite size effects and remain finite in the limit of infinite data points.

- **ReLU + max pool**: With respect to the previous case we add a pooling layer defined as

$$\rho_l^{(1;m)} = \rho^{(m)}\left(\left\{g\left(h_j^{(1)}\right)\right\}_{j\in S_l^m}\right), \qquad (14)$$

where, in our notation, $\rho^{(m)}(\cdot)$ indicates a pooling function with kernel size equals to m, S_l^m indicate the l^{th} subgroup of m nodes. In particular, for the max pooling

$$\rho_l^{(1;m)} \equiv \max_{j\in S_l^m}\left\{\max(0, g_j^{(1)})\right\}. \qquad (15)$$

As in the case of the ReLU, we have a wide distribution that does not concentrate in the limit of infinite data. The difference with the previous case is that now the distribution is peaked at the extremes of the support (we will elaborate more on this in Sec. 5.2); in this case, it is very likely that the untrained network will classify most of the dataset as belonging to only one of the two classes.

Note how, for the output layer, permutation symmetry in nodes corresponds to symmetry between classes. The latter is always preserved at the ensemble level,[4] in fact in all three cases $\overline{f_0(\mathcal{W})} = 1/2$, where $\overline{f_0(\mathcal{W})} \equiv \int p_{\mathcal{W}}\left(\tilde{\mathcal{W}}\right) f_0\left(\tilde{\mathcal{W}}\right) d\tilde{\mathcal{W}}$ indicate the average of $f_0(\mathcal{W})$ over the ensemble of weight initializations. However, while in the linear case, the symmetry between classes is also conserved on the single element of the ensemble,[5] in the other two cases the single realization diverges, *w.h.p.* from the mean over the ensemble. NSB thus results in a breaking of the symmetry between classes and a consequent self-averaging breakdown for the observable f_0. This difference between the two estimates is not due to finite network size effects; our predictions (theoretical curves in Fig. 5) are, in fact, formally exact in the limit of infinite size.

[4] Here, "ensemble" refers to all possible random initializations specified by the chosen scheme (e.g., Kaiming Normal).

[5] In this case, we have $\lim_{D\to\infty} f_0(\mathcal{W}) = \overline{f_0(\mathcal{W})} = 1/2$.

Which activations cause IGB and which do not Our results can be extended to a generic activation function. In particular, we can clarify (App. F) what is the fundamental attribute of the activated nodes, *i.e.* passed through the activation function and the pooling layer, $\{\rho_i^{(1;m)}\}_i$, that triggers NSB and consequently IGB:

Theorem 5.1 (Informal)**.** *Consider a Gaussian distributed dataset processed through a single hidden layer perceptron, in the limit of infinite width hidden layer, and whose weights are initialized according to the Kaiming normal scheme (with null bias weights). Then, we will have an absence of IGB in the network if, and only if, for the generic i^{th} node of the hidden layer:*

$$\left\langle \rho_i^{(1;m)} \right\rangle = 0 \ , \quad \forall i. \tag{16}$$

Or conversely, IGB will emerge if, and only if

$$\left\langle \rho_i^{(1;m)} \right\rangle \neq 0 \ , \quad \forall i. \tag{17}$$

Thus, the results we obtained on IGB apply to any kind of activation function: activations without IGB align with the description for linear activations, while those with IGB qualitatively resemble ReLU.

The effect of data pre-processing Condition (17) relies on data averages $\langle \dots \rangle$, which means IGB can be controlled by data standardization. Although our analysis until now centered on data around 0 (otherwise, we would not be able to attribute IGB to the mentioned architectural elements), other standardization methods (*e.g.*, inputs in [0,1]) will induce IGB. In App. G.1, we focus on this aspect and demonstrate how the choice of input standardization can amplify IGB. Thus, in the experiments shown in Fig. 2 and App. I.2, we use standardization as way to tune IGB without changing the architecture design (which would instead result in incomparable training curves).

As a general guideline, antisymmetric activations exhibit no IGB when the data is centered around zero inputs. Condition (17) also suggests that activations can be redefined to gain or lose their IGB property. For instance, in App. F.3, we demonstrate that appropriately shifting ReLU functions can eliminate IGB.

5.2. Conditions for strong IGB

We are now interested in the conditions that lead an untrained model to assign *all* the examples of a dataset to the same class (Strong IGB). There are in fact specific preprocessing and architecture choices that can amplify IGB (for example, in Fig. 5, IGB is exacerbated through max pooling). We can show that standardization, max pooling and network depth can all lead to *strong IGB*.

Theorem 5.2 (Conditions for strong IGB, informal)**.** *Consider a centered Gaussian distributed dataset that undergoes preprocessing through the standardization*

$$\psi\left(\boldsymbol{\xi}^{(\boldsymbol{a})}\right) = \boldsymbol{\xi}^{(\boldsymbol{a})} + \boldsymbol{K},$$

where $\boldsymbol{K}$ is an offset with the same dimensionality of the input. The data is then processed through a Multi-Layer Perceptron (MLP) with L hidden layers. This MLP employs ReLU nonlinearities and max pooling layers with a kernel size of m, and the weights are initialized using the Kaiming normal scheme (with zero biases).
In the limit of infinite width for hidden layers, the following scenarios are observed:

I. Shifting the center of the input distribution by a vector with the norm $|\boldsymbol{K}|$, even when $L = 1$ and $m = 1$, results in:

$$\lim_{|\boldsymbol{K}|\to\infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty. \tag{18}$$

II. With an increase in the kernel size of the pooling layer, even for a single hidden layer ($L = 1$), and in the absence of standardization offset ($|\boldsymbol{K}| = 0$), one has:

$$\lim_{m\to\infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty. \tag{19}$$

III. In the infinite-depth limit, even in the absence of pooling layers ($m = 1$) and standardization ($|\boldsymbol{K}| = 0$):

$$\lim_{L\to\infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty. \tag{20}$$

The proofs for points I., II., and III. are detailed in Apps. G.1, G.2, and G.3, respectively. It is important to note a key distinction among the three sources of IGB amplification we analyzed. Our analysis reveals that while the choice of data standardization and pooling can both amplify and initiate IGB (for instance, leading to the emergence of IGB in models where it would not appear otherwise), increasing network depth can only amplify existing IGB in the model, but cannot initiate its onset.

5.3. Experimental results

5.3.1. Robustness of IGB in real settings

Beyond the theoretical analysis presented, in App. I we present experiments demonstrating the presence of IGB in broader settings. In particular, we explore real data and vari-

ous architectures including CNNs, ResNets, MLP-mixers, and Vision Transformers. These experiments demonstrate the presence of IGB in settings commonly used in practice. They also highlight that the use of structured data accentuates IGB, combining the influence of model design with that originating from dataset correlations. The experiments analyze binary as well as multi-class datasets, both of which are covered by our theory.[6]

5.3.2. Effects of IGB on Training Dynamics

We also provide further experiments on the effect of IGB on learning dynamics with real data (both binary and multi-class) and different architectures (ResNet, MLP-mixer, and Vision Transformers), showing results qualitatively analogous to those shown in Fig. 2. We consider the case of balanced datasets, where the presence of IGB translates into a discrepancy between the fractions of class labels and the corresponding fractions of model predictions. This mismatch poses a natural limit on the performance we can reach, as explained in Sec. 3. In the presence of IGB in these settings, it becomes crucial to reabsorb the predictive bias during the dynamics to achieve good performance. It then becomes important to determine the time required by the dynamics to reabsorb the bias. Our experiments show how this reabsorbing time is critically dependent on the level of IGB; with high levels of IGB, convergence can become dramatically slow. It can also be observed how the improvement in performance over the dynamics proceeds consistently with the absorption of Guessing Bias. Figs. 2, I.4, and I.5 show this by tracking both accuracy and the maximum among the class fraction of guesses, used as a measure of IGB. The comparison of these two trends shows that during the dynamics, the level of guessing bias reduces while the performance improves accordingly.

5.3.3. Effects of IGB on Pre-Trained Models

We show that in the context of transfer learning, IGB can persist in pre-trained models, where only the final layer(s) of the architecture are untrained. Experiments outlined in App. I.3 (see also Fig. 6) show the presence of IGB in these models and its potential amplification, consistent with our analysis. Acknowledging the presence of IGB in this context is relevant, especially in few-shot learning (Wang et al., 2020), where the fine-tuning dynamics do not involve a large number of steps/epochs.

6. Discussion

We examined the classification bias in untrained neural networks, uncovering a phenomenon named IGB that arises from architectural choices that break permutation symmetry among hidden and output nodes within the same layer. The conditions that trigger IGB are related to model design, including data preprocessing, network depth, the selection of activation function and pooling. Factors such as max pooling, offsetted data standardization or network depth can exacerbate IGB to the extent that all dataset elements are assigned to a single class.

While our analysis is based on random *i.i.d.* data, we anticipate the presence of IGB in real datasets due to correlated data patches, increasing the likelihood of similar classifications. Hence, we expect stronger IGB in real datasets (as empirically confirmed in App. I.1). Moreover, the hypothesis of unstructured data is instrumental in highlighting the effects induced solely by model design.

Due to its data and architecture-dependent nature, IGB remains a complex phenomenon that requires further investigation to fully understand its implications. A comprehensive exploration of its effects across different scenarios remains an important area for future research. Our experimental results in App. I.2 demonstrate that IGB alters the training dynamics when using gradient-based methods. This alteration can sometimes be detrimental to training, but it can also be beneficial. For example, one could exploit IGB to combat data imbalance, or differences in the gradients related to different classes (Francazi et al., 2023). Furthermore, many works set the dynamics in the small learning rate regime (Francazi et al., 2023; Sarao Mannelli et al., 2020; Tarmoun et al., 2021); also in this context, the presence of IGB could turn out to be relevant, since the dynamics are more bound to the initial state.

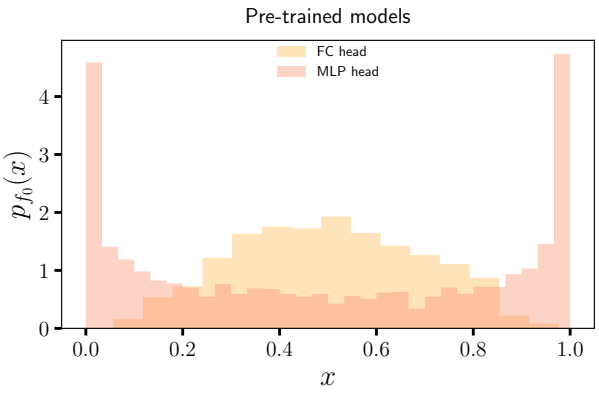

Figure 6. The distribution $p_{f_0}(x)$ on pre-trained EfficientNetV2 with the binary dataset (`C10`). The difference lies in the untrained head: a fully connected layer (`PT-ENetV2-FC`) versus a MLP (`PT-ENetV2-MLP`). More details in App. I.4. Increasing head depth amplifies IGB, consistently with our analysis.

[6] While we focus in our discussion on the binary case for the sake of clarity, refer to App. H

Impact statement

Our work identifies a source of prediction bias appearing at initialization in DNNs, and reveals that a model can be biased toward specific predictions, before it even saw the data it will be trained on. We also show that its presence has an effect on the early learning dynamics. This has, for example, an effect on hyperparameter tuning, which is part of model selection. Therefore, the choice of the final model could be influenced by biases.

By informing model selection, data preparation, and initial conditions, our results have the potential to enhance the training of machine learning models. Therefore our study not only advances theoretical knowledge but also promotes more considerate practices in the design of DNNs. It emphasizes the need to balance performance with fairness considerations.

Acknowledgements

This work was supported by the Swiss National Foundation, SNF grant # 196902.

References

Agarwala, A., Pennington, J., Dauphin, Y., and Schoenholz, S. Temperature check: theory and practice for training models with softmax-cross-entropy losses. *arXiv preprint arXiv:2010.07344*, 2020.

Barocas, S. and Selbst, A. D. Big data's disparate impact. *California law review*, pp. 671–732, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deng, L. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

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

Dotsenko, V. *An introduction to the theory of spin glasses and neural networks*, volume 55. World Scientific, 1994.

Dubey, S. R., Singh, S. K., and Chaudhuri, B. B. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 2022.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Steinhardt, J., and Madry, A. Identifying statistical bias in dataset replication. In *International Conference on Machine Learning*, pp. 2922–2932. PMLR, 2020.

Francazi, E., Baity-Jesi, M., and Lucchi, A. A theoretical analysis of the learning dynamics under class imbalance. In *International Conference on Machine Learning*, pp. 10285–10322. PMLR, 2023.

Fuchs, D. J. The dangers of human-like bias in machine-learning algorithms. *Missouri S&T's Peer to Peer*, 2(1):1, 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Gnedenko, B. V., Kolmogorov, A. N., Doob, J. L., and Hsu, P.-L. *Limit distributions for sums of independent random variables*, volume 233. Addison-wesley Reading, MA, 1968.

Gumbel, E. J. *Statistics of extremes*. Columbia university press, 1958.

Hagendorff, T. and Fabi, S. Why we need biased ai: How including cognitive biases can enhance ai systems. *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–14, 2023.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *nature*, 405(6789):947–951, 2000.

Han, X., Zhang, Z., Ding, N., Gu, Y., Liu, X., Huo, Y., Qiu, J., Yao, Y., Zhang, A., Zhang, L., et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

Hanin, B. and Rolnick, D. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, 31, 2018.

Hartarsky, I., Baity-Jesi, M., Ravasio, R., Billoire, A., and Biroli, G. Maximum-energy records in glassy energy landscapes. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(9):093302, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE*

conference on computer vision and pattern recognition, pp. 770–778, 2016.

Hutson, M. Who should stop unethical AI. *The New Yorker*, 2021.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Kang, K., Oh, J.-H., Kwon, C., and Park, Y. Generalization in a two-layer neural network. *Physical Review E*, 48(6): 4805, 1993.

Keller, J. B. and Kuske, R. Rate of convergence to a stable law. *SIAM Journal on Applied Mathematics*, 61(4):1308–1323, 2001.

Khanday, N. Y. and Sofi, S. A. Taxonomy, state-of-the-art, challenges and applications of visual understanding: A review. *Computer Science Review*, 40:100374, 2021.

Koehler, F., Zhou, L., Sutherland, D. J., and Srebro, N. Uniform convergence of interpolators: Gaussian width, norm bounds and benign overfitting. *Advances in Neural Information Processing Systems*, 34:20657–20668, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

Louppe, G., Kagan, M., and Cranmer, K. Learning to pivot with adversarial networks. *Advances in neural information processing systems*, 30, 2017.

Loureiro, B., Sicuro, G., Gerbelot, C., Pacco, A., Krzakala, F., and Zdeborová, L. Learning gaussian mixtures with generalized linear models: Precise asymptotics in high-dimensions. *Advances in Neural Information Processing Systems*, 34:10144–10157, 2021.

Matthews, A. G. d. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

Mézard, M., Parisi, G., and Virasoro, M. A. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

Mignacco, F., Krzakala, F., Urbani, P., and Zdeborová, L. Dynamical mean-field theory for stochastic gradient descent in gaussian mixture classification. *Advances in Neural Information Processing Systems*, 33:9540–9550, 2020.

Noci, L., Anagnostidis, S., Biggio, L., Orvieto, A., Singh, S. P., and Lucchi, A. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems*, 35: 27198–27211, 2022.

Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Oostwal, E., Straat, M., and Biehl, M. Hidden unit specialization in layered neural networks: Relu vs. sigmoidal activation. *Physica A: Statistical Mechanics and its Applications*, 564:125517, 2021.

Orvieto, A., Kohler, J., Pavllo, D., Hofmann, T., and Lucchi, A. Vanishing curvature and the power of adaptive methods in randomly initialized deep networks. *arXiv preprint arXiv:2106.03763*, 2021.

Panigrahi, I. and Zhu, R. Comparing importance sampling based methods for mitigating the effect of class imbalance. *arXiv preprint arXiv:2402.18742*, 2024.

Parraga, O., More, M. D., Oliveira, C. M., Gavenski, N. S., Kupssinskü, L. S., Medronha, A., Moura, L. V., Simões, G. S., and Barros, R. C. Debiasing methods for fairer neural models in vision and language research: A survey. *arXiv preprint arXiv:2211.05617*, 2022.

Paul, W. and Baschnagel, J. Stochastic processes. *From Physics to Finance, Springer, Berlin*, 1999.

Pennington, J. and Worah, P. The spectrum of the fisher information matrix of a single-hidden-layer neural network. *Advances in neural information processing systems*, 31, 2018.

Pennington, J., Schoenholz, S., and Ganguli, S. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

Pessach, D. and Shmueli, E. Algorithmic fairness. In *Machine Learning for Data Science Handbook: Data*

Mining and Knowledge Discovery Handbook, pp. 867–886. Springer, 2023.

Petrov, V. V. *Sums of independent random variables*, volume 82. Springer Science & Business Media, 2012.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016.

Pot, M., Kieusseyan, N., and Prainsack, B. Not all biases are bad: equitable and inequitable biases in machine learning and radiology. *Insights into imaging*, 12(1):1–10, 2021.

Qadeer, A. S. and Millar, E. Keeping you on track. *Science for the People*, 24(2):19–23, 2021. URL https://magazine.scienceforthepeople.org/vol24-2-dont-be-evil/keeping-you-on-track/.

Ramasinghe, S., Macdonald, L. E., Farazi, M., Saratchandran, H., and Lucey, S. How much does initialization affect generalization? In *International Conference on Machine Learning*, pp. 28637–28655. PMLR, 2023.

Ren, Y., Guo, S., Bae, W., and Sutherland, D. J. How to prepare your task head for finetuning. *arXiv preprint arXiv:2302.05779*, 2023.

Sarao Mannelli, S., Biroli, G., Cammarota, C., Krzakala, F., Urbani, P., and Zdeborová, L. Complex dynamics in simple neural networks: Understanding gradient flow in phase retrieval. *Advances in Neural Information Processing Systems*, 33:3265–3274, 2020.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Schehr, G. and Majumdar, S. N. Exact record and order statistics of random walks via first-passage ideas. In *First-passage phenomena and their applications*, pp. 226–251. World Scientific, 2014.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

Siddique, S., Haque, M. A., George, R., Gupta, K. D., Gupta, D., and Faruk, M. J. H. Survey on machine learning biases and mitigation techniques. *Digital*, 4(1):1–68, 2023.

Sompolinsky, H., Crisanti, A., and Sommers, H.-J. Chaos in random neural networks. *Physical review letters*, 61 (3):259, 1988.

Tarmoun, S., Franca, G., Haeffele, B. D., and Vidal, R. Understanding the dynamics of gradient flow in overparameterized linear models. In *International Conference on Machine Learning*, pp. 10153–10161. PMLR, 2021.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Uchaikin, V. V. and Zolotarev, V. M. *Chance and stability: stable distributions and their applications*. Walter de Gruyter, 2011.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Xiao, G., Lin, J., and Han, S. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 2023.

Xiao, L., Bahri, Y., Sohl-Dickstein, J., Schoenholz, S., and Pennington, J. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pp. 5393–5402. PMLR, 2018.

Yang, G. and Schoenholz, S. Mean field residual networks: On the edge of chaos. *Advances in neural information processing systems*, 30, 2017.

Yang, G., Pennington, J., Rao, V., Sohl-Dickstein, J., and Schoenholz, S. S. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

Ye, H.-J., Zhan, D.-C., and Chao, W.-L. Procrustean training for imbalanced deep learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 92–102, 2021.

Appendix

Contents

A. Notation

In this section, we meticulously present the setting and notation used in our analysis. Our theory examines Multi-Layer Perceptrons (MLPs), where the propagation of an input signal, denoted as $\boldsymbol{\xi}$, through the network is governed by the following pair of equations:

$$h_i^{(l+1)} = \sum_j w_{ij}^{(l+1)} \rho_j^{(l;m)}, \tag{21}$$

$$g_i^{(l)} = g\left(h_i^{(l)}\right), \tag{22}$$

$$\rho_i^{(l;m)} = \rho^{(m)}\left(\left\{g\left(h_j^{(l)}\right)\right\}_{j\in S_l^m}\right), \tag{23}$$

where $l \in \{0,\dots,L+1\}$, $g(\cdot),: \mathbb{R}\to\mathbb{R}$ represents the activation function, $\rho^{(m)}(\cdot),: \mathbb{R}^m \to \mathbb{R}$ denotes the pooling function with kernel size equals to m, S_l^m indicate the l^{th} subgroup of m nodes; $\rho_i^{(0;m)} \equiv \boldsymbol{\xi}$ and $h_c^{(L+1)} \equiv O^{(c)}$. The principal notation is summarized in Fig. A.1.

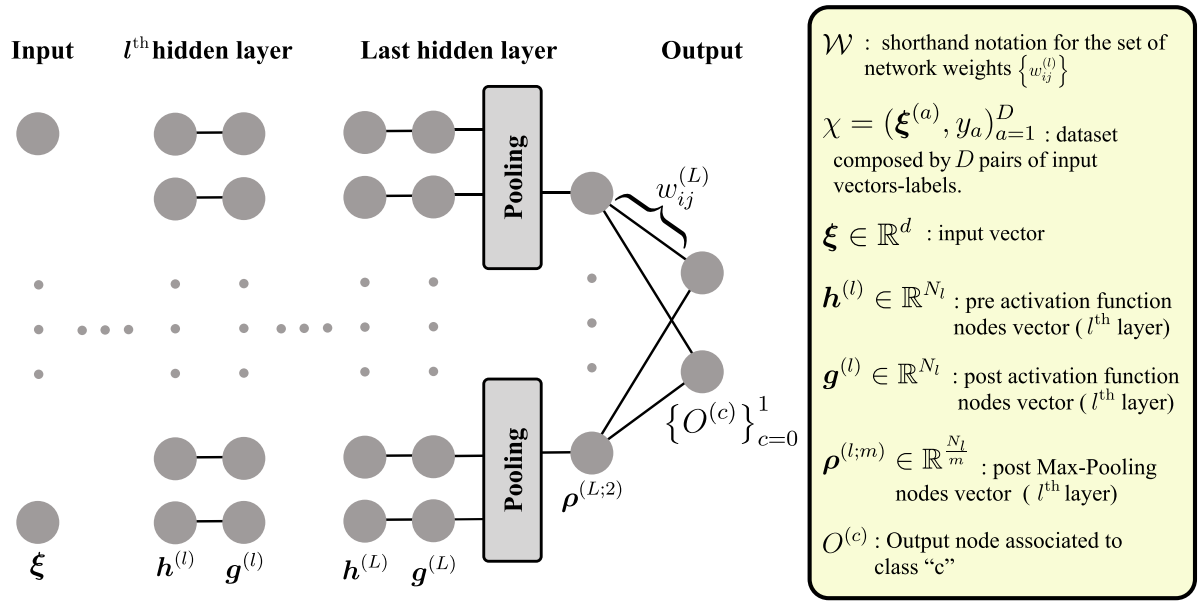

Figure A.1. Illustration of a generic neural network for a binary classification problem (left) and main symbols (right).

We also introduce the following notation:

- $\{\cdot\}_{i=0}^{M-1}$: set of M elements. If some of the indices of the variables are fixed (*i.e.* equal for every element of the set), the set indices (indices that vary across different elements of the set) are reported explicitly on the right. If the index of the set elements is not explicitly reported, it means the absence of fixed indices for the set variables (*i.e.* all indices are set indices).
- $\mathbb{E}(x)$: Indicate the expectation value of the argument, x. If the average involves only one source of randomness this is explicitly indicated, *e.g.* $\mathbb{E}_\chi(x)$ indicates an average over the dataset distribution, while $\mathbb{E}_{\mathcal{W}}(x)$ an average over the distribution of network weights. For the sake of compactness, we will employ, where necessary the shorthand notation $\langle x\rangle \equiv \mathbb{E}_\chi(x)$ and $\bar{x} \equiv \mathbb{E}_{\mathcal{W}}(x)$.

- $\mathrm{erf}\,(\cdot)$: Error function. $\mathrm{erf}\,(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$
- $F_X\,(x)$:Given a *r.v.* X, we denote its cumulative distribution function (*c.d.f.*) as $F_X\,(x)$, i.e., $F_X\,(x) = \mathbb{P}\,(X < x)$. Considering the *r.v.* $X(\chi, \mathcal{W})$, which is a function of two independent sets of random variables χ and $\mathcal{W}$, when one of these sources of randomness is fixed, we explicitly indicate the active source in the notation. For example, $F_X^{(\chi)}\,(x) = \mathbb{P}\,(X < x \mid \mathcal{W})$.
- f_c: fraction of dataset elements classified as belonging to class c. The argument M indicates the total number of output nodes for the variable definition, *i.e.* the number of classes considered. For binary problems we omit this argument (f_0) as there is only one non-trivial possibility, *i.e.* $M = 2$.
- $f_{\tilde{c}}$: the set $\{f_c\}_{c=0}^{M-1}$ contain the same elements of $\{f_{\tilde{c}}\}_{c=0}^{M-1}$, but these are ranked by magnitude, such that $f_{\tilde{0}}$ is the greatest output value between the M elements of the set, $f_{\tilde{1}}$ the second one and so on.
- $\boldsymbol{g}^{(l)}$: vector of the l^{th} hidden layer nodes, $\left(g_0^{(l)}, \ldots, g_{N_l-1}^{(l)}\right)$, after passing through the activation function; $g_i^{(l)}$ indicate the component corresponding to node i.
- $\boldsymbol{h}^{(l)}$: vector of the l^{th} hidden layer nodes, $\left(h_0^{(l)}, \ldots, h_{N_l-1}^{(l)}\right)$, before passing through the activation function; $h_i^{(l)}$ indicate the component corresponding to node i.
- $N_C \equiv N_{L+1}$: number of output nodes, *i.e.* the number of classes.
- N_l: number of nodes in the l^{th} layer; $N_0 \equiv d$ indicates the dimension of the input data (number of input layer nodes) while N_{L+1} the number of classes (number of output layer nodes).
- $\mathcal{N}\left(x; \mu, \sigma^2\right)$: Given a Gaussian *r.v.* X, we indicate with $\mathcal{N}\left(x; \mu, \sigma^2\right)$ the *p.d.f.* computed at $X = x$, *i.e.* $\mathcal{N}\left(x; \mu, \sigma^2\right) \equiv p_X\,(x) = \frac{e^{-\frac{1}{2\sigma^2}(x-\mu)^2}}{\sqrt{2\pi\sigma^2}}$
- $O_M^{(c)}$: output layer node; c is the node index; the index M, instead, indicate the total number of nodes considered. For binary problems we omit the subscript index to keep the notation lighter, *i.e.* $O^{(c)}$.
- $O_M^{(\tilde{0})}$: the set $\left\{O_M^{(\tilde{c})}\right\}_{c=0}^{M-1}$ contain the same elements of $\left\{O_M^{(\tilde{c})}\right\}_{\tilde{c}=0}^{M-1}$, but these are ranked by magnitude, such that $O_M^{(\tilde{0})}$ is the greatest output value between the M, $O_M^{(\tilde{1})}$ the second one, and so on.
- $\mathbb{P}\,(A)$: Denotes the probability associated with event A. $\mathbb{P}\,(A \mid B)$ indicates the probability of event A given event B.
- $p_X\,(x')$: Given a random variable X, $p_X\,(x')$ denotes the probability density function (*p.d.f.*) evaluated at $X = x'$. Formally, $p_X\,(x') = \frac{d}{dx} F_X\,(x)\Big|_{x=x'}$. For variables with multiple sources of randomness, if some of these sources are either fixed or marginalized, we will specify the active sources of randomness as subscripts. For example, given $X(\chi, \mathcal{W})$, a function of two independent sets of random variables χ and $\mathcal{W}$, $p_X^{(\chi)}\,(x')$ equals $F_X^{(\chi)}\,(x)\Big|_{x=x'}$.
- $\mathrm{Var}\,(\cdot)$: Indicate the variance of the argument. Since we have *r.v.*s with multiple sources of randomness where necessary we will specify in the subscript the source of randomness used to compute the expectation. For example $\mathrm{Var}_\chi\,(\cdot) \equiv \langle \cdot - \langle \cdot \rangle \rangle^2$. For the sake of compactness, we will employ sometimes the shorthand notation $\mathrm{Var}_\chi\,(\cdot) = \sigma_\cdot^2$.
- $\mathcal{W}_t$: shorthand notation for the set of network weights, $\{w_{ij}^{(l)}\}$ at time t; $\mathcal{W} \equiv \mathcal{W}_0$. We use, instead the notation $\mathcal{W}^l \subseteq \mathcal{W}$ to indicate the subset of weights relative to a specific layer, *i.e.* $\mathcal{W}^l \equiv \{w_{ij}^{(l)}\}_{\substack{j\in[0,\ldots,N_l] \\ i\in[0,\ldots,N_{l+1}]}}$. $\mathcal{W}^{<l}$, $\mathcal{W}^{>l}$, ... are defined analogously.
- $w_{ij}^{(l)}$: element ij of the matrix $\boldsymbol{W}^{(l)}$, connecting two consecutive layers ($l \in [0, \ldots, L]$). Given the matrix $\boldsymbol{W}^{(l)}$ we use a 'placeholder' index, $\cdot$, to return column and row vectors from the weight matrices. In particular $w_{j\cdot}^{(l)}$ denotes row j of the weight matrix $\boldsymbol{W}^{(l)}$; similarly, $w_{\cdot j}^{(l)}$ denotes column j.

- $\mu_c \equiv \left\langle O^{(c)} \right\rangle$: This notation is a shorthand to represent the mean value of the output node c, computed with respect to the dataset, for a fixed set of initialized weights $\mathcal{W}$. In mathematical terms, it denotes the conditional expectation of the output node c, given a specific set of weights.
- $\boldsymbol{\rho}^{(l;m)}$: vector of the l^{th} hidden layer, $\left(\rho_0^{(l;m)}, \ldots, \rho_{\left\lceil \frac{N_l - 1}{m} \right\rceil}^{(l;m)}\right)$, after passing through the max pooling layer with kernel size m; $\rho_i^{(l;m)}$ indicate the component corresponding to node i.
- $\chi = (\boldsymbol{\xi}^{(a)}, y_a)_{a=1}^{D}$: dataset composed by D pairs of input vectors-labels.
- $\boldsymbol{\xi}^{(a)} \in \mathbb{R}^d$: a^{th} input vector; when the index a is omitted we mean a generic vector, $\boldsymbol{\xi}$, drawn from the population distribution.
- $\Theta(x)$: Heaviside step function.
- $\delta(x)$: Dirac delta function.

Abbreviations

- *c.d.f.*: cumulative distribution function
- CNN: Convolutional neural network
- DNN: Deep neural network
- IGB : Initial guessing bias
- MLP: Multi-layer perceptron
- NSB: Node symmetry breaking
- *p.d.f.*: probability density function
- *r.v.*: random variable
- *w.h.p.*: with high probability

B. Additional related work

In the following, we discuss in detail some crucial aspects that distinguish our analysis from other investigations present in the literature.

Sources of randomness A DNN at initialization can be interpreted as a random function parameterized by random weights, and whose inputs are sampled from an unknown data distribution. There are thus two distinct and independent sources of randomness at initialization: weights and data. Previous theoretical studies of DNNs typically consider, for a given input, the whole ensemble of random initializations of the weights (Poole et al., 2016; Matthews et al., 2018; Novak et al., 2018). In contrast, we fix the weight initialization and study the network's behavior by taking expectations over the data. Technically speaking, while previous work first averages over the weights and then over the data, our averages are first over the data and then over the weights. Note that this approach is closer to the natural order followed in practice, where data is classified by a single neural network (and thus for a single weight realization).

Breaking of self-averageness The mentioned inversion might seem a technicality but it actually constitutes a fundamental point in the study of IGB. One observable that characterizes the phenomenon is the fraction of points classified as class c, and denoted by $f_c(\mathcal{W})$, (this quantity will be formally introduced shortly) in fact does not respect the self-averaging property (Mézard et al., 1987; Dotsenko, 1994).[7] In other words, even in the infinite dataset/network size limit, the value of

[7] In a system defined over an ensemble of realizations (in our case, each realization is a different weight initialization), a self-averaging quantity is one that can be equivalently calculated either by averaging over the whole ensemble or on a single, sufficiently large, realization; in such a situation, a single huge system is adequate to represent the entire ensemble.

$f_0(\mathcal{W})$ obtained on a given realization of the network initialization, $\mathcal{W}$, differs from that computed from the average over the ensemble of initializations. Self-averaging is often exploited (*e.g.* in *dynamical mean field theory* (Sompolinsky et al., 1988), employed also in the context of mean-field theories of deep learning (Schoenholz et al., 2016; Poole et al., 2016; Xiao et al., 2018; Yang & Schoenholz, 2017; Yang et al., 2019)), as it can lead to a simplification of the analysis. For the phenomena related to IGB we cannot exploit self-averaging.

Nodes symmetry breaking When we think of an untrained multi-layer perceptron (MLP), we are naturally inclined to assume a permutation symmetry among the various nodes of a given layer. Instead, we will show that this symmetry can be broken, and indeed, this symmetry breaking (*i.e.* difference in the distribution of nodes in a given layer) constitutes the foundation of the IGB. Nodes Permutation Symmetry Breaking (NSB) was already reported in previous work, *e.g.* in specific shallow networks with a large number of examples (Kang et al., 1993).
We will see that the choice of architecture significantly influences the presence of IGB. Architecture design, particularly the selection of activation functions, has been extensively studied (Dubey et al., 2022), with a focus on ReLU versus differentiable activations such as sigmoid. For instance, sigmoid activations can achieve dynamical isometry, maximizing signal propagation depth (Schoenholz et al., 2016), unlike ReLUs (Pennington et al., 2017). These activations are also compared in terms of generalization performance, revealing distinct behaviors for ReLUs and sigmoids (Oostwal et al., 2021). Notably, these studies often consider averaging over weight initializations.

C. Limit distributions

This section reviews the convergence in distribution of a sequence of *r.v.* to its asymptotic distribution. We focus on the conditions that guarantee this convergence and the characterization of the resulting asymptotic distribution.

C.1. Convergence in distribution

Our discussion begins with the convergence of *r.v.* combinations to their asymptotic distributions. We will examine the conditions under which this convergence occurs and how to define the resulting asymptotic distribution. The starting point is the *Central Limit Theorem (CLT)* and its subsequent extensions.

C.1.1. Central limit theorem

The *Central Limit Theorem* (CLT) plays a significant role in our analysis. Here, we provide a foundational discussion of the theorem, excluding proofs. For a more comprehensive exploration, refer to (Gnedenko et al., 1968; Uchaikin & Zolotarev, 2011; Paul & Baschnagel, 1999). Consider a sequence of *i.i.d. r.v.*s $x_1, x_2, x_3, \ldots$, each drawn from a population with a finite overall mean μ and variance σ^2. Denoting $\bar{x}_n$ as the sample mean of the first n samples, the distribution's limiting form, $Z = \lim_{n\to\infty}\left(\frac{\bar{x}_n-\mu}{\sigma_{\bar{x}_n}}\right)$ where $\sigma_{\bar{x}_n} \equiv \frac{\sigma}{\sqrt{n}}$, converges to a standard normal distribution, *i.e.*, $\mathcal{N}(z;0,1)$. For large but finite n, this represents the leading term of an expansion; corrections to the asymptotic Gaussian profile are $o\left(\frac{1}{\sqrt{n}}\right)$ as detailed in Keller & Kuske (2001).

C.1.2. CLT extension

The CLT can be extended beyond the assumption of identically distributed variables. This extension is possible by applying conditions like the Lyapunov or Lindeberg conditions, which ensure the validity of the CLT under broader circumstances. For a sequence of independent *r.v.*s $x_1, x_2, x_3, \ldots$, each with mean $\mu_i = \mathbb{E}(x_i)$, the Lindeberg condition is formulated as:

$$\lim_{n\to\infty}\sum_{i=1}^{n}\frac{1}{s_n^2}\int_{|x|\geq\epsilon s_n}(x-\mu_i)^2 dF_{X_i}(x) = 0\,, \tag{24}$$

where $s_n^2 = \sum_{i=1}^{n}\mathbb{E}\left((x_i-\mu_i)^2\right)$ and $F_{X_i}(x)$ is the cumulative distribution function (c.d.f.) of x_i.

Theorem C.1 (Lindeberg theorem)**.** *For a set of independent r.v.s $\{X_i\}_{i=1}^n$, if Lindeberg's condition holds for all positive ϵ, then, defining $S_n = X_1 + \cdots + X_n$ and $Z_n = \frac{S_n - \sum_i \mu_i}{s_n}$, we have*

$$p_{Z_n}(z) \xrightarrow{n\to\infty} \mathcal{N}(z;0,1)\,. \tag{25}$$

Our analysis will utilize an alternative set of necessary and sufficient conditions that guarantee the asymptotic convergence to a normal distribution. For further details, see Petrov (2012).

Firstly, Thm. C.2 introduces necessary and sufficient conditions for the convergence of *r.v.* sums to a Gaussian distribution. Subsequently, Thm. C.3 outlines a set of sufficient conditions, which are particularly applicable to our cases of interest. We will demonstrate that *r.v.*s satisfying these sufficient conditions also fulfill the requirements of Thm. C.2.

Theorem C.2 (Distribution convergence)**.** *Let us consider a set of independent zero-mean r.v.s* $\{X_i\}_{i=1}^n$.[a] *If and only if, for every fixed* $\epsilon > 0$*, the following conditions are satisfied:*

Concentration:

$$\sum_{i=1}^{n} \mathbb{P}\left(|X_i| \geq \epsilon\right) \xrightarrow{n\to\infty} 0 \quad \forall \epsilon \in \mathbb{R}^+ , \tag{26}$$

Mean Normalization:

$$\sum_i \left(\int_{|x|<\epsilon} x\, p_{X_i}(x)\, dx \right) \xrightarrow{n\to\infty} 0 , \tag{27}$$

Variance Normalization:

$$s_n^2 = \sum_i \left(\int_{|x|<\epsilon} x^2\, p_{X_i}(x)\, dx - \left(\int_{|x|<\epsilon} x\, p_{X_i}(x)\, dx \right)^2 \right) \xrightarrow{n\to\infty} \sigma^2 , \tag{28}$$

the distributions of the sum $\sum_i X_i$ *will converge to* $\mathcal{N}\left(0, \sigma^2\right)$.

[a] In general if $\mathbb{E}\left(X_i\right) \neq 0$ we can define a new set of variables $\{(X_i - \mathbb{E}\left(X_i\right))\}$.

For the proof of Thm. C.2, refer to Chapter 4 of Petrov (2012).

Theorem C.3 (Sufficient condition for Thm. C.2)**.** *Let us consider a set* $\{X_i\}_{i=1}^n$ *of independent, zero-mean,*[a] *r.v.s satisfying, for some constants* $\tilde{\sigma}^2 \in \mathbb{R}^+$*, the following conditions*

Variance convergence:

$$\sum_{i=1}^{n} \mathbb{E}\left(X_i^2 - \mathbb{E}\left(X_i\right)^2 \right) \xrightarrow{n\to\infty} n\tilde{\sigma}^2 , \tag{29}$$

Fast decreasing tails:

$$\lim_{x\to\pm\infty} p_{X_i}(x) = \mathcal{O}\left(\frac{1}{x^4}\right), \quad \forall i . \tag{30}$$

Let us define the new set $\{\tilde{X}_i\}_{i=1}^n$ *such that*

$$\tilde{X}_i \equiv \frac{X_i}{c\sqrt{n}} , \tag{31}$$

where $c > 0$ *is a constant. We now proof that the set* $\{\tilde{X}_i\}_{i=1}^n$ *satisfy* (26)*,* (27)*,* (28) *leading to the convergence in distribution of* $\tilde{S}_n = \sum_i \tilde{X}_i$ *to* $\mathcal{N}\left(0, \sigma^2\right)$.

[a] In general if $\mathbb{E}\left(X_i\right) \neq 0$ we can define a new set of variables $\{(X_i - \mathbb{E}\left(X_i\right))\}$.

Proof. We will now prove that, starting from Eqs. (29) and(30), the conditions of Thm. C.2 holds.

We start showing the Concentration condition [(26)]

$$\begin{aligned}
&\sum_{i=1}^{n}\mathbb{P}\left(|\tilde{X}_i|\geq\epsilon\right)=\sum_{i=1}^{n}\left(\int_{-\infty}^{-\epsilon}p_{\tilde{X}_i}\left(\tilde{x}\right)d\tilde{x}+\int_{\epsilon}^{\infty}p_{\tilde{X}_i}\left(\tilde{x}\right)d\tilde{x}\right)= \\
&\sum_{i=1}^{n}\left(\int_{-\infty}^{-c\sqrt{n}\epsilon}p_{X_i}\left(x\right)dx+\int_{c\sqrt{n}\epsilon}^{\infty}p_{X_i}\left(x\right)dx\right)\overset{n\gg1}{=} \\
&\sum_{i=1}^{n}\left(\int_{-\infty}^{-c\sqrt{n}\epsilon}\mathcal{O}\left(\frac{1}{x^4}\right)dx+\int_{c\sqrt{n}\epsilon}^{\infty}\mathcal{O}\left(\frac{1}{x^4}\right)dx\right)=\sum_{i=1}^{n}\left(\mathcal{O}\left(\int_{-\infty}^{-c\sqrt{n}\epsilon}\left|\frac{1}{x^4}\right|dx\right)+\mathcal{O}\left(\int_{c\sqrt{n}\epsilon}^{\infty}\left|\frac{1}{x^4}\right|dx\right)\right)= \\
&\sum_{i=1}^{n}\mathcal{O}\left(\frac{1}{n^{\frac{3}{2}}}\right)=\mathcal{O}\left(\frac{1}{n^{\frac{1}{2}}}\right)\xrightarrow{n\to\infty}0.
\end{aligned}\tag{32}$$

In the third line we used the *Fast decreasing tails* condition [Eq. (30)].
Now we show the validity of mean normalization condition [Eq. (27)]:

$$\begin{aligned}
&\sum_{i}^{n}\left(\int_{|\tilde{x}|<\epsilon}\tilde{x}\,p_{\tilde{X}_i}\left(\tilde{x}\right)d\tilde{x}\right)\overset{n\gg1}{=}\sum_{i}^{n}\underbrace{\mathbb{E}\left(\tilde{X}_i\right)}_{=0}-\sum_{i}^{n}\left(\mathcal{O}\left(\int_{-\infty}^{-c\sqrt{n}\epsilon}\frac{1}{c\sqrt{n}}\left|\frac{1}{x^3}\right|dx\right)+\mathcal{O}\left(\int_{c\sqrt{n}\epsilon}^{\infty}\frac{1}{c\sqrt{n}}\left|\frac{1}{x^3}\right|dx\right)\right)= \\
&=\mathcal{O}\left(\frac{1}{n^{\frac{1}{2}}}\right)\xrightarrow{n\to\infty}0\,.
\end{aligned}\tag{33}$$

Finally, for the Variance normalization condition [Eq. (28)], we can now replace Eq. (33) into the definition of the variance. This gives

$$\sum_{i}\left(\int_{|\tilde{x}|<\epsilon}\tilde{x}^2\,p_{\tilde{X}_i}\left(\tilde{x}\right)d\tilde{x}-\left(\int_{|\tilde{x}|<\epsilon}\tilde{x}\,p_{\tilde{X}_i}\left(\tilde{x}\right)d\tilde{x}\right)^2\right)\overset{n\gg1}{=}\tag{34}$$

$$\frac{\tilde{\sigma}^2}{c^2}-\sum_{i}^{n}\left(\mathcal{O}\left(\int_{-\infty}^{-c\sqrt{n}\epsilon}\frac{1}{c^2n}\left|\frac{1}{x^2}\right|dx\right)+\mathcal{O}\left(\int_{c\sqrt{n}\epsilon}^{\infty}\frac{1}{c^2n}\left|\frac{1}{x^2}\right|dx\right)\right)+\mathcal{O}\left(\frac{1}{n}\right)=\tag{35}$$

$$\frac{\tilde{\sigma}^2}{c^2}+\mathcal{O}\left(\frac{1}{n^{\frac{1}{2}}}\right)\xrightarrow{n\to\infty}\frac{\tilde{\sigma}^2}{c^2}.\tag{36}$$

□

C.2. Concentration of *r.v.*s distribution

When analyzing a set of n *r.v.*s whose variances scale with n, we encounter phenomena related to the concentration of their sum's distribution. In App. C.2.1, we examine examples pertinent to our study, illustrating various scenarios based on the scaling differences. We will explore cases where the distribution remains asymptotically stable and others where it concentrates, *i.e.*, narrows around a single value.

Then we apply these results to analyze the distribution $p^{(\chi)}_{h_i^{(l+1)}}\left(x\right)$; fixed $\mathcal{W}$, in fact, $h_i^{(l+1)}$ can be expressed as a combination of *r.v.*s. We will demonstrate how the nodes $\left\{h_i^{(l+1)}\right\}_i$ may not be identically distributed for $l\geq1$.

C.2.1. Combination of random variables

We approach the problem step by step, starting with a simplified case to demonstrate how the scaling of the variance of independent random variables is crucial in the distributional convergence of their sum. As we will see, this preliminary example contains key elements that we will extend to cases of interest for our analysis.
Consider a set of independent *r.v.*s $\{X_i\}_{i=1}^n$ where $X_i\sim\mathcal{N}\left(0,\frac{\sigma^2}{n}\right),\ \forall i$, and σ^2 does not scale with n, *i.e.*, $\sigma^2=\mathcal{O}(1)$. Our

interest lies in analyzing the sum's distribution, particularly to ascertain if it exhibits asymptotic concentration phenomena. Define $S_x^{(n)} = \sum_{i=1}^n X_i$. To determine if the distribution of $S_x^{(n)}$ narrows as $n \to \infty$, we examine the standard deviation as an estimate for the fluctuation magnitude from the mean. From X_i's definition and variance additivity, we derive that

$$\sigma_{S_x^{(n)}} = \sqrt{\mathbb{E}\left(S_x^{(n)} - \mathbb{E}\left(S_x^{(n)}\right)\right)^2} = \mathcal{O}(1). \tag{37}$$

This implies that the sum's distribution remains asymptotically stable, meaning it does not narrow around the mean value.

Now, consider the *r.v.* $S_{x^2}^{(n)} = \sum_{i=1}^n X_i^2$. Given that for a Gaussian variable $\sigma_{x^2}^2 = \mathcal{O}\left(\sigma_x^4\right)$, and employing the variance's additivity, it follows that

$$\sigma_{S_{x^2}^{(n)}} = \sqrt{\mathbb{E}\left(S_{x^2}^{(n)} - \mathbb{E}\left(S_{x^2}^{(n)}\right)\right)^2} = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right). \tag{38}$$

In this case, there is a concentration phenomenon; the fluctuations of $S_{x^2}^{(n)}$ asymptotically approach zero, *i.e.*, the distribution's measure concentrates around the mean value.

We can proceed similarly in the presence of linear combinations of *r.v.*s, where we again consider the sum of *r.v.*s, this time each coupled with a fixed coefficient.
To examine the changes in node distribution from one layer to the next, we analyze the distribution of $h_i^{(l+1)} = \sum_j w_{ij}^{(l)} g_j^{(l)}$, with a fixed set $\mathcal{W}$. This leads to a linear combination of *r.v.*s. It is straightforward to show that the hypotheses of Thm. C.3 are satisfied in this context. We then have

$$p^{(\chi)}_{h_i^{(l+1)}}(x) = \mathcal{N}\left(x; \sum_j w_{ij}^{(l)} \left\langle g_j^{(l)} \right\rangle, \sum_j \left(w_{ij}^{(l)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(l)}\right)\right). \tag{39}$$

Our goal is to understand how $p^{(\chi)}_{h_i^{(l+1)}}(x)$ changes across the set $\left\{h_i^{(l+1)}\right\}_{i=1}^{N_{l+1}}$. Since we are dealing with a Gaussian distribution, it suffices to examine how its mean and variance vary over different nodes $\left\{h_i^{(l+1)}\right\}_{i=1}^{N_{l+1}}$, *i.e.*, using different sets of weights $\left\{w_{i\cdot}^{(l)}\right\}_{i=1}^{N_{l+1}}$, with $w_{i\cdot}^{(l)} \equiv \left\{w_{ij}^{(l)}\right\}_{j=1}^{N_l}$.

Consider the mean $\left\langle h_i^{(l+1)} \right\rangle = \sum_j w_{ij}^{(l)} \left\langle g_j^{(l)} \right\rangle$. Now, the set $\left\{\left\langle g_j^{(l)} \right\rangle\right\}_{j=1}^{N_l}$ acts as a fixed constant set,[8] while the set $\left\{w_{i\cdot}^{(l)}\right\}_{j=1}^{N_l}$ changes with the node index i. Again, Thm. C.3 is easily seen to be applicable. We get

$$p^{(w_{i\cdot}^{(l)})}_{\left\langle h_i^{(l+1)} \right\rangle}(x) = \mathcal{N}\left(x; \sum_j \overline{w_{ij}^{(l)}} \left\langle g_j^{(l)} \right\rangle, \sum_j \mathrm{Var}_{\mathcal{W}}\left(w_{ij}^{(l)}\right) \left\langle g_j^{(l)} \right\rangle^2\right) = \mathcal{N}\left(x; 0, \sigma_w^2 \frac{1}{N_l} \sum_{j=1}^{N_l} \left\langle g_j^{(l)} \right\rangle^2\right), \tag{40}$$

where the last step follows from the fact that $\overline{w_{ij}^{(l)}} = 0$ and $\mathrm{Var}_{\mathcal{W}}\left(w_{ij}^{(l)}\right) = \frac{\sigma_w^2}{N_l}$.
Assuming $\left\langle g_j^{(l)} \right\rangle^2$ follows the CLT,[9] we have

$$\frac{1}{N_l} \sum_{j=1}^{N_l} \left\langle g_j^{(l)} \right\rangle^2 \xrightarrow{N_l \to \infty} \mathbb{E}_{w_{j\cdot}^{(l-1)}}\left(\left\langle g_j^{(l)} \right\rangle^2\right), \tag{41}$$

leading to

$$p^{(w_{i\cdot}^{(l)})}_{\left\langle h_i^{(l+1)} \right\rangle}(x) \xrightarrow{N_l \to \infty} \mathcal{N}\left(x; 0, \sigma_w^2 \mathbb{E}_{w_{j\cdot}^{(l-1)}}\left(\left\langle g_j^{(l)} \right\rangle^2\right)\right). \tag{42}$$

[8] This is the same set of elements $\forall i \in [1, \ldots, N_{l+1}]$.
[9] The *law of large numbers* would suffice here as we are not studying convergence to the mean value.

Regarding the variance in Eq. (39), it will also be a *r.v.* dependent on the set $\{w_{ij}^{(l)}\}_{j=1}^{N_l}$. Following a similar approach to the one used for the sum of Gaussian *r.v.*s discussed before, we can calculate its mean and variance to show that asymptotically $\sum_j \left(w_{ij}^{(l)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(l)}\right)$ converges to a deterministic value equal for all N_{l+1} nodes.

To make the discussion more concrete we will discuss now, more in detail the specific case of $l = 3$ This will be used as an integral part of deriving $p_{f_0}(x)$ for deep architectures (see App. G.3.1). In App. G.3.1, we demonstrate that

$$p^{(\chi)}_{h_i^{(3)}}(x) \xrightarrow{N_2\to\infty} \mathcal{N}\left(x; \left\langle h_i^{(3)}\right\rangle, \mathrm{Var}_\chi\left(h_i^{(3)}\right)\right) \equiv \mathcal{N}\left(x; \sum_{j=1}^{N_2} w_{ij}^{(2)}\left\langle g_j^{(2)}\right\rangle, \sum_{j=1}^{N_2}\left(w_{ij}^{(2)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(2)}\right)\right). \tag{43}$$

Our aim is to apply the analysis described above to discern the differences between the set of distributions $\left\{p^{(\chi)}_{h_i^{(3)}}(x)\right\}$. Variations in these distributions are induced by differences in the corresponding sets of weight vectors $\{w_{i\cdot}^{(2)}\}$. The sets $\left\{\left\langle g_j^{(2)}\right\rangle\right\}$ and $\left\{\mathrm{Var}_\chi\left(g_j^{(2)}\right)\right\}$, however, remain consistent across each node $h_i^{(3)}$. Therefore, we treat the latter sets as fixed random coefficients, while the variations are attributed to $\{w_{i\cdot}^{(2)}\}$. Let us focus on the variance:

$$\mathrm{Var}_\chi\left(h_i^{(3)}\right) = \sum_{j=1}^{N_2}\left(w_{ij}^{(2)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(2)}\right). \tag{44}$$

To understand how much the random variable $\mathrm{Var}_\chi\left(h_i^{(3)}\right)$ varies, we calculate its mean and variance:

$$\mathbb{E}_{w_{i\cdot}^{(2)}}\left(\sum_{j=1}^{N_2}\left(w_{ij}^{(2)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(2)}\right)\right) = \sigma_w^2 \frac{1}{N_2}\sum_{j=1}^{N_2}\mathrm{Var}_\chi\left(g_j^{(2)}\right) \xrightarrow{N_2\to\infty} \sigma_w^2 \mathbb{E}_{w_{j\cdot}^{(1)}}\left(\mathrm{Var}_\chi\left(g_j^{(2)}\right)\right) \tag{45}$$

with

$$\mathbb{E}_{w_{j\cdot}^{(1)}}\left(\mathrm{Var}_\chi\left(g_j^{(2)}\right)\right) = \int_{\mathbb{R}} \underline{\mathrm{Var}_\chi\left(g_j^{(2)}\right)}(x)\ \mathcal{N}\left(x; \left\langle h_j^{(2)}\right\rangle, \sigma_w^2\left\langle g^{(1)}\right\rangle^2\right) dx\,, \tag{46}$$

where with $\underline{\mathrm{Var}_\chi\left(g_j^{(2)}\right)}(x)$ we underlined the dependence of $\mathrm{Var}_\chi\left(g_j^{(2)}\right)$ from the *r.v.* $\left\langle h_j^{(2)}\right\rangle$. Note that in the integrand of Eq. (46) there is no scaling dependence with respect to N_2; this means that

$$\mathbb{E}_{w_{j\cdot}^{(1)}}\left(\mathrm{Var}_\chi\left(g_j^{(2)}\right)\right) = \mathcal{O}(1) \Longrightarrow \mathbb{E}_{w_{i\cdot}^{(2)}}\left(\mathrm{Var}_\chi\left(h_i^{(3)}\right)\right) = \mathcal{O}(1). \tag{47}$$

To evaluate the order of magnitude of the fluctuations, we recall that, given a fixed coefficient $\mathrm{Var}_\chi\left(g_j^{(2)}\right)$,

$$\mathrm{Var}_{w_{i\cdot}^{(2)}}\left(\left(w_{ij}^{(2)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(2)}\right)\right) = \mathrm{Var}_\chi\left(g_j^{(2)}\right)^2 \mathrm{Var}_{w_{i\cdot}^{(2)}}\left(\left(w_{ij}^{(2)}\right)^2\right). \tag{48}$$

Also, for gaussian variables $\sigma^2_{x^2} = \mathcal{O}\left(\sigma_x^4\right)$ so $\mathrm{Var}_{w_{i\cdot}^{(2)}}\left(\left(w_{ij}^{(2)}\right)^2\right) = \mathcal{O}\left(\frac{1}{N_2^2}\right)$. Then from the extensitivity of the variance follows that

$$\sqrt{\mathbb{E}_{w_{i\cdot}^{(2)}}\left(\mathrm{Var}_\chi\left(h_i^{(3)}\right)^2\right) - \mathbb{E}_{w_{i\cdot}^{(2)}}\left(\mathrm{Var}_\chi\left(h_i^{(3)}\right)\right)^2} = \mathcal{O}\left(\frac{1}{\sqrt{N_2}}\right). \tag{49}$$

By this we conclude that, in the $N_2 \to \infty$ limit, the distribution of the *r.v.* $\sum_{j=1}^{N_2}\left(w_{ij}^{(2)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(2)}\right)$ narrows around the mean value $\sigma_w^2 \mathbb{E}_{w_{j\cdot}^{(1)}}\left(\mathrm{Var}_\chi\left(g_j^{(2)}\right)\right)$.

We can proceed analogously to evaluate the mean value fluctuations; therefore we have to repeat the same analysis on the mean of $p^{(\chi)}_{h_i^{(3)}}(x)$, *i.e.* (from Eq. (43)) $\sum_{j=1}^{N_2} w_{ij}^{(2)} \left\langle g_j^{(2)} \right\rangle$. In this case, we have

$$\mathbb{E}_{w_{i\cdot}^{(2)}} \left(\sum_{j=1}^{N_2} w_{ij}^{(2)} \left\langle g_j^{(2)} \right\rangle \right) = \sum_{j=1}^{N_2} \underbrace{\mathbb{E}_{w_{i\cdot}^{(2)}} \left(w_{ij}^{(2)} \right)}_{=0} \left\langle g_j^{(2)} \right\rangle = 0 \,. \tag{50}$$

Proceeding as in (48) we get

$$\sqrt{\mathbb{E}_{w_{i\cdot}^{(2)}} \left(\left\langle h_i^{(3)} \right\rangle^2 \right) - \mathbb{E}_{w_{i\cdot}^{(2)}} \left(\left\langle h_i^{(3)} \right\rangle \right)^2} = \mathcal{O}(1) \,. \tag{51}$$

This means that the center of the distribution $p^{\chi}_{h_i^{(3)}}(x)$ keeps fluctuating from node to node even in the limit $N_2 \to \infty$.

More specifically

$$\left(\mathbb{E}_{w_{i\cdot}^{(2)}} \left(\left\langle h_i^{(3)} \right\rangle^2 \right) - \mathbb{E}_{w_{i\cdot}^{(2)}} \left(\left\langle h_i^{(3)} \right\rangle \right)^2 \right) = \sigma_w^2 \frac{1}{N_2} \sum_{j=1}^{N_2} \left\langle g_j^{(2)} \right\rangle^2 \xrightarrow{N_2 \to \infty} \sigma_w^2 \mathbb{E}_{w_{j\cdot}^{(1)}} \left(\left\langle g_j^{(2)} \right\rangle^2 \right) . \tag{52}$$

Also, it is easy to show that the set $\left\{ w_{ij}^{(2)} \left\langle g_j^{(2)} \right\rangle \right\}_{j=1}^{N_2}$ respect the conditions of Thm. C.3. Therefore, to summarize

$$p_{\left\langle h_i^{(3)} \right\rangle}(x) \xrightarrow{N_2 \to \infty} \mathcal{N} \left(x; 0, \sigma_w^2 \mathbb{E}_{w_{j\cdot}^{(1)}} \left(\left\langle g_j^{(2)} \right\rangle^2 \right) \right) , \tag{53}$$

while

$$p_{\mathrm{Var}_\chi \left(h_i^{(3)} \right)}(x) \xrightarrow{N_2 \to \infty} \delta \left(x - \sigma_w^2 \mathbb{E}_{w_{j\cdot}^{(1)}} \left(\mathrm{Var}_\chi \left(g_j^{(2)} \right) \right) \right) . \tag{54}$$

D. Derivation of the distributions

In this section, our focus is on the *r.v.* $O^{(c)}$. The primary goal is to prove Thm. 4.1 by deriving expressions for $p_{\mu_c}(x)$ and $p^{(\chi)}_{O^{(c)}}(x)$. These expressions are crucial for determining $p_{f_0}(x)$, as explored in App. E.1.

We begin by demonstrating Thm. 4.1 for Multi-Layer Perceptrons (MLPs) with a single hidden layer. Due to the modular structure of MLPs, this demonstration can be extended to deeper network architectures, as shown in App. G.3. The sketch proof of Thm. 4.1 (found in App. D.1) will be followed by a detailed analysis focused on specific settings, serving as exemplary cases to highlight key qualitative differences.

For this detailed analysis on concrete examples, we will proceed in steps:

- In App. D.2, we will derive an expression for $p_{\mu_c}(x)$ for various cases of interest.
- In App. D.3, we will derive $p^{(\chi)}_{O^{(c)}}(x)$ for the same cases.

Remark 1. Consider the *r.v.* $h_i^{(1)}$ defined as:

$$h_i^{(1)} = \sum_j w_{ij}^{(0)} \xi_j . \tag{55}$$

The independence of $h_i^{(1)}$ is a direct consequence of the assumed independence of the input components ξ_j.

For the *r.v.* $O^{(c)}$, defined by

$$O^{(c)}(\boldsymbol{\xi}; \mathcal{W}) = \sum_{m=1}^{N_1} w_{cm}^{(1)} g_m^{(1)}, \tag{56}$$

we require independence in the set $\{g_m^{(1)}\}$ to apply the Central Limit Theorem (CLT). Since the activation function is an element-wise operation on $\{h_m^{(1)}\}$, the independence of $h_m^{(1)}$ suffices.

Jointly normally distributed random variables are independent if and only if their covariance is zero. We calculate the covariance $\mathrm{Cov}(h_i^{(1)}, h_j^{(1)})$ and demonstrate that it converges to 0 *w.h.p.* as the input dimensionality $d \to \infty$. Specifically, we have:

$$\mathrm{Cov}\left(\sum_{m=1}^{d} w_{im}^{(1)}\xi_m, \sum_{n=1}^{d} w_{jn}^{(1)}\xi_n\right) \overset{\boldsymbol{a}}{=} \sum_{m=1}^{d}\sum_{n=1}^{d} w_{im}^{(1)}w_{jn}^{(1)}\mathbb{E}(\xi_m\xi_n) \overset{\boldsymbol{b}}{=} \sum_{m=1}^{d} w_{im}^{(1)}w_{jm}^{(1)}\mathbb{E}(\xi_m^2) \overset{\boldsymbol{c}}{=} \sum_{m=1}^{d} w_{im}^{(1)}w_{jm}^{(1)}. \tag{57}$$

Here, step $\boldsymbol{a}$ utilizes linearity and the zero mean assumption of ξ_m. Steps $\boldsymbol{b}$ and $\boldsymbol{c}$ result from substituting the covariance matrix of $\boldsymbol{\xi}$, assuming $\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbb{I})$. The final summation comprises products of pairs of Gaussian-distributed terms, each with zero mean and variance $\mathcal{O}\left(\frac{1}{d}\right)$. As such, it follows from the properties of Gaussian distributions that this summation converges to 0 *w.h.p.* as $d \to \infty$.

D.1. Proof of Thm. 4.1

We provide here a sketch proof of Thm. 4.1, detailed as follows:

Theorem D.1. *Consider a Gaussian distributed dataset with i.i.d. components, i.e., $\xi_b^{(a)} \sim \mathcal{N}(0,1)$ [a] processed through an MLP (mapping the input to output according to Eq. (21), Eq. (22), Eq. (23)) with a single hidden layer (i.e. $L = 1$) and weights initialized according to the Kaiming normal scheme, i.e., $w_{ij}^{(l)} \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{N_{l-1}}\right)$, with zero bias weights. Assuming that the variables $\{\rho_l^{(1;m)}\}_{l=1}^{\lceil \frac{N_1}{m} \rceil}$ satisfy the conditions of Thm. C.3, in the limit of infinite input size and infinite width hidden layer, the distribution of an output node $O^{(c)}$, at initialization, is given by:*

$$p_{O^{(c)}}^{(\chi)}(x) \xrightarrow{d,N_1\to\infty} \mathcal{N}\left(x; \mu_c, Var_\chi\left(O^{(c)}\right)\right), \tag{58}$$

Furthermore, while the variance of the distribution, $Var_\chi\left(O^{(c)}\right)$, converges w.h.p. to a deterministic value, the center of this distribution, μ_c, is itself a random variable, varying from node to node, following the distribution:

$$p_{\mu_c}(x) = \mathcal{N}\left(x; 0, Var_{\mathcal{W}}(\mu_c)\right). \tag{59}$$

We can distinguish two cases:

- ***Absence of IGB**: In the absence of IGB, in the limit of infinite datasets, $Var_{\mathcal{W}}(\mu_c)$ converges to 0, i.e.,*

$$\lim_{D\to\infty} Var_{\mathcal{W}}(\mu_c) = 0 \Rightarrow \lim_{D\to\infty} p_{\mu_c}(x) = \delta(x) \tag{60}$$

- ***Presence of IGB**: In the presence of IGB, the differences in node centers are not exclusively due to finite size effects, in other words:*

$$\lim_{D\to\infty} Var_{\mathcal{W}}(\mu_c) \neq 0 \tag{61}$$

[a] $\xi_b^{(a)}$ denotes the b-th component of the a-th input vector

Proof. Fixing the set $\mathcal{W}$, $h_i^{(1)}$ is a linear combination of *r.v.*s which meet the conditions of Thm. C.3. Thus, in the limit as $d \to \infty$, we have

$$p_{h_i^{(1)}}^{(\chi)}(x) \xrightarrow{d\to\infty} \mathcal{N}\left(x; \sum_j w_{ij}^{(0)}\langle\xi_j\rangle, \sum_j \left(w_{ij}^{(0)}\right)^2 \sigma_{\xi_j}^2\right) = \mathcal{N}\left(x; 0, \sum_j \left(w_{ij}^{(0)}\right)^2\right), \tag{62}$$

where we have used the definition of $\boldsymbol{\xi}$ for our analysis to substitute $\langle \xi_j \rangle = 0$ and $\sigma^2_{\xi_j} = 1 \; \forall j$. As shown in Sec. C.2.1, we find

$$\sum_{j=1}^{d} \left(w_{ij}^{(0)} \right)^2 = d \, \overline{\left(w_{ij}^{(0)} \right)^2} \xrightarrow{d \to \infty} \sigma_w^2 \Longrightarrow p^{(\chi)}_{h_i^{(1)}}(x) \xrightarrow{d \to \infty} \mathcal{N}\left(x; 0, \sigma_w^2\right). \tag{63}$$

Now, passing through the activation function, and possibly through the pooling layer, we have a set of *i.i.d.* variables, $\{\rho_l^{(1;m)}\}_{l=1}^{\lceil \frac{N_1}{m} \rceil}$. Furthermore, by hypothesis, the set $\{\rho_l^{(1;m)}\}_{l=1}^{\lceil \frac{N_1}{m} \rceil}$ satisfies the conditions of Thm. C.3. Therefore, the distribution of the *r.v.*

$$O^{(c)}(\boldsymbol{\xi}; \mathcal{W}) = \sum_{l=1}^{N_1} w_{cl}^{(1)} \rho_l^{(1;m)}. \tag{64}$$

will converge, in the limit of $N_1 \to \infty$, to a Gaussian distribution:

$$\lim_{N_1 \to \infty} p^{(\chi)}_{O^{(c)}}(x) = \mathcal{N}\left(x; \mu_c, \mathrm{Var}_\chi\left(O^{(c)}\right)\right). \tag{65}$$

In particular, since the variables $\{\rho_l^{(1;m)}\}_{l=1}^{\lceil \frac{N_1}{m} \rceil}$ are *i.i.d.* and the weights $\{w_{cl}^{(1)}\}_{l=1}^{\lceil \frac{N_1}{m} \rceil}$ are fixed constants from initialization, defining

$$\lim_{D \to \infty} \left\langle \rho_l^{(1;m)} \right\rangle = \left\langle \rho^{(1;m)} \right\rangle \quad \forall l, \tag{66}$$

$$\lim_{D \to \infty} \mathrm{Var}_\chi\left(\rho_l^{(1;m)}\right) = \mathrm{Var}_\chi\left(\rho^{(1;m)}\right) \quad \forall l. \tag{67}$$

from Eq. (64) it follows:

$$\mu_c = \left\langle \rho^{(1;m)} \right\rangle \sum_{l=1}^{N_1} w_{cl}^{(1)} \equiv \left\langle \rho^{(1;m)} \right\rangle S_{w_c}^{(N_1)} \tag{68}$$

$$\mathrm{Var}_\chi\left(O^{(c)}\right) = \mathrm{Var}_\chi\left(\rho^{(1;m)}\right) \sum_{l=1}^{N_1} \left(w_{cl}^{(1)}\right)^2 \equiv \mathrm{Var}_\chi\left(\rho^{(1;m)}\right) S_{w_c^2}^{(N_1)}. \tag{69}$$

From the analysis in App. C.2.1, we know that while $S_{w_c}^{(N_1)}$ is a *r.v.* with a non-degenerate distribution for all N_1, the distribution of $S_{w_c^2}^{(N_1)}$ concentrates around its mean value in the limit $N_1 \to \infty$. In other words,

$$\lim_{N_1 \to \infty} \mathrm{Var}_\chi\left(O^{(c)}\right) = \sigma_w^2 \mathrm{Var}_\chi\left(\rho^{(1;m)}\right) \quad \forall c. \tag{70}$$

We can thus distinguish two cases:

- $\left\langle \rho^{(1;m)} \right\rangle = 0$:
  In this case, the output nodes will be identically distributed, i.e., there will be an absence of IGB. Specifically, from Eq.s (65) and (68) and , it follows:

$$\lim_{N_1 \to \infty} p^{(\chi)}_{O^{(c)}}(x) = \mathcal{N}\left(x; 0, \sigma_w^2 \mathrm{Var}_\chi\left(\rho^{(1;m)}\right)\right). \tag{71}$$

- $\left\langle \rho^{(1;m)} \right\rangle \neq 0$:
  In this case, the output nodes will be centered at different points, leading to the emergence of IGB. Specifically, from Eq.s (65) and (68) , it follows:

$$\lim_{N_1 \to \infty} p^{(\chi)}_{O^{(c)}}(x) = \mathcal{N}\left(x; \left\langle \rho^{(1;m)} \right\rangle S_{w_c}^{(N_1)}, \sigma_w^2 \mathrm{Var}_\chi\left(\rho^{(1;m)}\right)\right). \tag{72}$$

Since $S_{w_c}^{(N_1)}$ is a sum of N_1 *i.i.d.* Gaussians, $w_{cl}^{(1)} \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{N_1}\right)$, it follows

$$S_{w_c}^{(N_1)} \sim \mathcal{N}\left(0, \sigma_w^2\right). \tag{73}$$

Consequently,

$$p_{\mu_c}(x) = \mathcal{N}\left(x; 0, \sigma_w^2 \left\langle \rho^{(1;m)} \right\rangle^2\right). \tag{74}$$

□

D.2. Derivation of $p_{\mu_c}(x)$ for some emblematic cases

Thm. D.1 demonstrates that the decisive element for deriving $p_{\mu_c}(x)$ is $\left\langle \rho^{(1;m)} \right\rangle$. To make the discussion more concrete, we will consider some cases to illustrate their distinguishing features. In particular, we will use the same settings discussed at the beginning of Sec. 5.

Linear Consider the case where

$$\rho_i^{(1;1)} = \rho^{(1;1)}\left(g\left(h_i^{(1)}\right)\right) = h_i^{(1)}. \tag{75}$$

From Eq.s (62) and (75), it follows that $\left\langle \rho^{(1;1)} \right\rangle = 0$. Therefore, in this setting, we have no presence of IGB, and according to Eq. 60, we have:

$$\lim_{D\to\infty} p_{\mu_c}(x) = \delta(x). \tag{76}$$

Note that (76) is formally true only in the limit of an infinite size dataset. For a finite size dataset, we must compute the average $\left\langle \rho^{(1;1)} \right\rangle$ using the empirical distribution, $p_{h_k^{(1)}}^{(\chi;E)}(x) \equiv \sum_{a=1}^{D} \frac{1}{D}\delta\left(h_k^{(1)} - \hat{h}_k^{(1)}\left(\boldsymbol{\xi}^{(a)}\right)\right)$, where the set $\left\{\hat{h}_k^{(1)}\left(\boldsymbol{\xi}^{(a)}\right)\right\}_{a=1}^{D}$ contains the values mapped from each element of the dataset to the k^{th} node of the first hidden layer. We then have

$$\left\langle \rho_j^{(1;1)} \right\rangle = \int_{\mathbb{R}} x\, p_{h_k^{(1)}}^{(\chi;E)}(x)\, dx = \frac{1}{D}\sum_{a=1}^{D} \hat{h}_j^{(1)}\left(\boldsymbol{\xi}^{(a)}\right) \underset{\boldsymbol{A}}{\simeq} \frac{1}{\sqrt{D}} Z_j = \tilde{Z}_j, \tag{77}$$

where $Z_j \sim \mathcal{N}(0, \sigma_w^2)$, and $\tilde{Z}_j \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{D}\right)$.

Therefore, considering finite size effects, $\left\langle \rho^{(1;1)} \right\rangle \neq \left\langle \rho_j^{(1;1)} \right\rangle$. In this case, instead of starting from Eq. (68), we can proceed similarly to the derivation of Eq. (40), obtaining

$$p_{\mu_c}(x) = \mathcal{N}\left(x; 0, \sigma_w^2 \overline{\tilde{Z}_j^2}\right) = \mathcal{N}\left(x; 0, \frac{4}{D}\right). \tag{78}$$

Note that in step $\boldsymbol{A}$ of Eq. (77), we again used the CLT for the sum of *r.v.*s. The key difference with the previous derivation is that we do not use the CLT for $p_{h_k^{(1)}}^{(\chi)}(x)$; instead, we substitute the empirical distribution, $p_{h_k^{(1)}}^{(\chi;E)}(x)$. This latter distribution, defined on the dataset elements, accounts for the finite size effects of the dataset itself (see dependence from D). Note that the distribution of the variables involved in the sum in the last part of Eq. (77) ($\tilde{Z}_j$) narrows as $D \to \infty$; in this limit, the finite size effects disappear, and we converge to the result in Eq. (76).

ReLU We now repeat the computation using the ReLU activation function (introduced in Hahnloser et al. (2000)), *i.e.*,

$$\rho_i^{(1;1)} = \rho^{(1;1)}\left(g_j^{(1)}\right) = g_j^{(1)} = g\left(h_j^{(1)}\right) \equiv \max\{0, h_j^{(1)}\}. \tag{79}$$

$g_j^{(1)}$ will follow the same distribution as $h_j^{(1)}$ on the positive support, since $g_j^{(1)} = h_j^{(1)}$ for $h_j^{(1)} > 0$. The probability density on the negative support of $p^{(\chi)}_{h_j^{(1)}}(x)$ will collapse to 0 since $g_j^{(1)} = 0$ for $h_j^{(1)} < 0$. Substituting $p^{(\chi)}_{h_j^{(1)}}(x)$ with the Gaussian distribution from Eq. (63), we get:

$$p^{(\chi)}_{g_j^{(1)}}(x) = \left(\int_{\mathbb{R}^-} p^{(\chi)}_{h_j^{(1)}}(\tilde{x})\, d\tilde{x}\right)\delta(x) + \Theta(x)\, p^{(\chi)}_{h_j^{(1)}}(x) = \frac{1}{2}\delta(x) + \Theta(x)\,\mathcal{N}\left(x; 0, \sigma_w^2\right), \tag{80}$$

where $\delta(x)$ represents the Dirac delta distribution, and $\Theta(x)$ represents the Heaviside step function.

Integrating over this distribution, we proceed as in the previous case:

$$\begin{aligned}\left\langle \rho^{(1;1)} \right\rangle = \left\langle \rho_j^{(1;1)} \right\rangle = \int_{\mathbb{R}} x\, p^{(\chi)}_{g_j^{(1)}}(x)\, dx = \int_{\mathbb{R}} x\left(\frac{1}{2}\delta(x) + \mathcal{N}\left(x; 0, \sigma_w^2\right)\right) dx = \\ \int_0^{+\infty} x\mathcal{N}\left(x; 0, \sigma_w^2\right) dx = \sigma_w^2 \int_0^{+\infty} \mathcal{N}\left(x; 0, \sigma_w^2\right) d\frac{x^2}{2\sigma_w^2} = \frac{\sigma_w}{\sqrt{2\pi}}.\end{aligned} \tag{81}$$

The first notable difference in the case of ReLU activation is that $\left\langle \rho^{(1;1)} \right\rangle \neq 0$, thus indicating the emergence of IGB. Specifically, from Eq. (74), it follows:

$$p_{\mu_c}(x) = \mathcal{N}\left(x; 0, \sigma_w^2 \frac{\sigma_w^2}{2\pi}\right) = \mathcal{N}\left(x; 0, \frac{2}{\pi}\right), \tag{82}$$

where in the last step we have substituted σ_w with the typical gain value used in the presence of ReLU, *i.e.*, $\sigma_w = \sqrt{2}$ (He et al., 2015).

In this scenario, unlike the linear activation function case discussed at the end of App. D.2, we do not need to consider the finite size effects of the dataset. While for a linear activation function $\lim_{D\to\infty} \left\langle \rho^{(1;1)} \right\rangle = 0$, for ReLU $\lim_{D\to\infty} \left\langle \rho^{(1;1)} \right\rangle \neq 0$ (as shown in (81)). Therefore, the finite size corrections of $\mathcal{O}\left(\frac{1}{D}\right)$ are negligible when $D \left\langle \rho^{(1;1)} \right\rangle \gg 1$.

In Fig. D.1, the distributions $p_{\mu_c}(x)$ along with their respective theoretical curves, *i.e.*, Eq. (78) and Eq. (82), are compared for the two types of activation functions. Consistent with Eq. (78) and Eq. (82), we observe that in the linear case, the distribution narrows around 0 as D increases, whereas in the ReLU case, it remains stable.

ReLU + max pool With max pooling, we group the set $\{g_k^{(1)}\}$ into subgroups and select the maximum value from each, thereby reducing the dimensionality of the layer. Incorporating this into the analysis in App. D.2, we have:

$$\rho_l^{(1;m)} = \rho\left(\left\{g\left(h_j^{(1)}\right)\right\}_{j\in S_l^m}\right) = \max_{j\in S_l^m}\{\max\{0, g_j^{(1)}\}\}, \tag{83}$$

where S_l^m indicates the l-th subgroup of m nodes.

To proceed with the computation from the previous section, we need to derive $p^{(\chi)}_{\rho_l^{(1;m)}}(x)$. Generally, if $Y \equiv \max\{X_1, \ldots, X_m\}$ with $\{X_i\}_{i=1}^m$ being i.i.d., we have:

$$F_Y(y) \equiv \mathbb{P}(Y \leq y) = \mathbb{P}(\max\{X_1, \ldots, X_m\} \leq y) = \prod_{i=1}^m \mathbb{P}(X_i \leq y) \equiv \prod_{i=1}^m F_X(y) = F_X(y)^m. \tag{84}$$

Differentiating, we find:

$$p_Y(y) = m\, p_X(y)\, F_X(y)^{m-1}. \tag{85}$$

For our case, we substitute $p_X(x)$ with the distribution used in App. D.2, *i.e.*,

$$p_X(x) \equiv p^{(\chi)}_{g_j^{(1)}}(x) = \frac{1}{2}\delta(x) + \Theta(x)\,\mathcal{N}\left(x; 0, \sigma_w^2\right). \tag{86}$$

From Eq. (86), it follows:

$$F_X(y) \equiv \mathbb{P}\left(g_j^{(1)} \leq y \mid \mathcal{W}\right) = \int_{-\infty}^{y} \frac{1}{2}\delta(x) + \Theta(x)\,\mathcal{N}\left(x; 0, \sigma_w^2\right) dx = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{y}{\sqrt{2\sigma_w^2}}\right)\Theta(y). \tag{87}$$

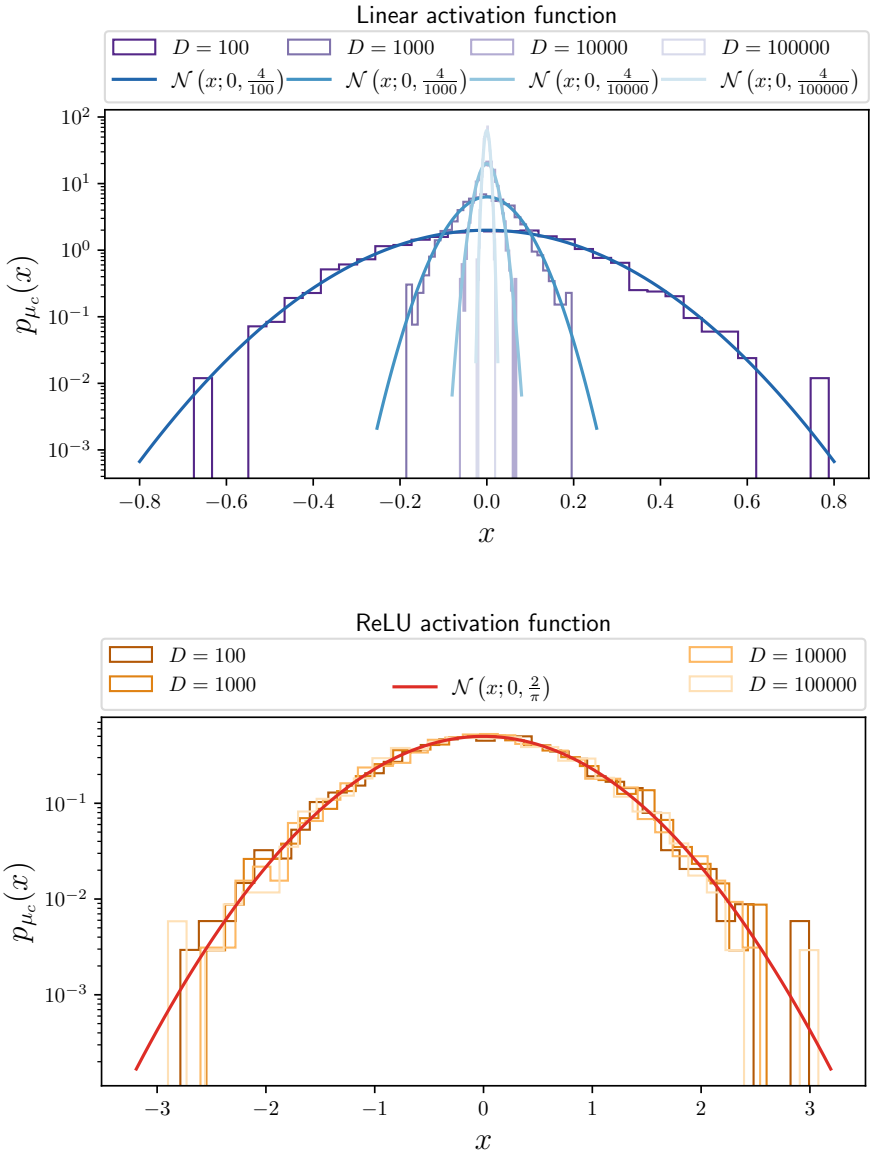

Figure D.1. Comparison of the distribution $p_{\mu_c}(x)$ between the linear activation function (upper plot) and ReLU (bottom plot) across different dataset sizes (D). Both plots also include the theoretical distributions derived in App. D.2. Observe the distinct behaviors in the two cases: for the linear activation function, the distribution narrows as D increases, while for the ReLU case, it remains stable. This stability in the ReLU case aligns with expectations from Eq. (81) and Eq. (77). These simulations employed the `GB` dataset and the `SHLP` model. For more details, refer to App. I.4.

Applying the Fisher–Tippett–Gnedenko theorem, we can find an asymptotic result for $p^{(\chi)}_{\rho_l^{(1;m)}}(x)$ as $m \to \infty$. For general m values, combining all components yields:

$$\left\langle \rho^{(1;m)} \right\rangle = \left\langle \rho_l^{(1;m)} \right\rangle = \int_{\mathbb{R}} x\, p^{(\chi)}_{\rho_l^{(1;m)}}(x)\, dx = \int_0^\infty m\, x\, \mathcal{N}\left(x; 0, \sigma_w^2\right) \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2\sigma_w^2}}\right)\right)^{m-1} dx. \tag{88}$$

This can be approximated by an asymptotic expansion or evaluated numerically. In the case of linear activation with max pooling, a simple Gaussian distribution is substituted for $p_X(x)$.

Defining a notation to simplify expressions:

$$c_m(\sigma_w) \equiv \int_0^\infty m\, x\, \mathcal{N}\left(x; 0, \sigma_w^2\right) \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2\sigma_w^2}}\right)\right)^{m-1} dx. \tag{89}$$

In this case too, IGB emerges; specifically, from Eq. (74), it follows:

$$p_{\mu_c}(x) = \mathcal{N}\left(x; 0, \sigma_w^2 c_m(\sigma_w)^2\right). \tag{90}$$

D.3. Derivation of $p^{(\chi)}_{O^{(c)}}(x)$ for some emblematic cases

As discussed, μ_c is a *r.v.* that varies with the set of weights $\mathcal{W}$. In App. D.2, we derived its distribution. In this section, we focus on the distribution of the output, given a specific configuration of the weight set, $p^{(\chi)}_{O^{(c)}}(x)$, for the same cases analyzed in App. D.2.

- **Linear**: Starting from Eq. (75) and combining Eq.(63) with Eq. 71, we get

$$\lim_{N_1 \to \infty} p^{(\chi)}_{O^{(c)}}(x) = \mathcal{N}\left(x; 0, \sigma_w^4\right). \tag{91}$$

- **ReLU**: From Eq. (80), we have

$$\left\langle g_m^{(1)} \right\rangle = \frac{\sigma_w}{\sqrt{2\pi}}, \tag{92}$$

$$\begin{aligned} \mathrm{Var}_\chi\left(g_m^{(1)}\right) &= \int_{\mathbb{R}^+} x^2 p^{(\chi)}_{g_m^{(1)}}(x)\, dx - \left\langle g_m^{(1)} \right\rangle^2 = \\ &= \frac{\sigma_w^2(\pi - 1)}{2\pi}, \end{aligned} \tag{93}$$

  where the symmetry of $p^{(\chi)}_{h_m^{(1)}}(x)$ was used. Substituting into Eq. (72) and using Eq. (93), we obtain

$$\lim_{N_1 \to \infty} p^{(\chi)}_{O^{(c)}}(x) = \mathcal{N}\left(x; \mu_c, \sigma_w^2 \frac{\sigma_w^2(\pi - 1)}{2\pi}\right), \tag{94}$$

  where μ_c is a *r.v.* distributed according to Eq. (82).

- **ReLU + max pool**: This case is conceptually similar to the ReLU case; the main difference is that we do not have an analytical expression for $\left\langle \rho_i^{(1;m)} \right\rangle$ and $\mathrm{Var}_\chi\left(\rho_i^{(1;m)}\right)$. We can numerically compute these cumulants for a fixed m, with their expressions given by

$$\left\langle \rho^{(1;m)} \right\rangle = \int_0^\infty mx\mathcal{N}\left(x; 0, \sigma_w^2\right) \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2\sigma_w^2}}\right)\right)^{m-1} dx, \tag{95}$$

$$\mathrm{Var}_\chi\left(\rho^{(1;m)}\right) = \left(\int_0^\infty mx^2\mathcal{N}\left(x; 0, \sigma_w^2\right) \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{x}{\sqrt{2\sigma_w^2}}\right)\right)^{m-1} dx\right) - \left\langle \rho^{(1;m)} \right\rangle^2. \tag{96}$$

Thus, analogous to Eq. (94), we obtain

$$\lim_{N_1\to\infty} p^{(\chi)}_{O^{(c)}}(x) = \mathcal{N}\left(x; \mu_c, \sigma_w^2 \mathrm{Var}_\chi\left(\rho^{(1;m)}\right)\right), \tag{97}$$

where μ_c is a *r.v.* distributed according to Eq. (90).

D.4. Non identically distributed classes

For our analysis we consider unstructured data identically distributed among classes. This choice places us in a symmetric setting where the presence of a predictive bias cannot be linked to differences between classes. This setting allows us to isolate the effect of the architecture sidestepping potential additional effects coming from dataset attributes. However, extending the analysis to other settings is possible; to illustrate this we now show that the analysis can be extended to include classes that are not identically distributed. Specifically, we will extend the current setting by introducing class-specific differences in the variance of the Gaussian blobs. We begin by demonstrating that, given an initialization, both classes exhibit a predictive bias towards the same class. Let's define a generic value for the class c input variance, $\sigma^2_{\xi^{(c)}}$. We can then repeat the computation for each class subgroup as done in App. D.1, specifically for an MLP with ReLU activation. We divide the dataset into subgroups based on class membership (each now following different statistics) and repeat the computation for each. In particular, consider Eq. (68) and assume (without loss of generality) that for a given class c, we have a prediction bias towards class 0, i.e.

$$\mu_0^{(c)} - \mu_1^{(c)} = \frac{\sigma_w \sigma_{\xi^{(c)}}}{\sqrt{2\pi}}(S_{w_0} - S_{w_1}) > 0. \tag{98}$$

This expression comes from substituting Eq. (81) with the class-dependent variance. Then, for a generic class c', we will have:

$$\mu_0^{(c')} - \mu_1^{(c')} = \frac{\sigma_w \sigma_{\xi^{(c')}}}{\sqrt{2\pi}}(S_{w_0} - S_{w_1}) = \frac{\sigma_{\xi^{(c')}}}{\sigma_{\xi^{(c)}}}\mu_0^{(c)} - \mu_1^{(c)} > 0$$

indicating that the predictions for elements of class c' will also be biased towards class 0.

Thus, for any given initialization, the bias is directed towards the same class for each subgroup. Furthermore, we can show that each subgroup is characterized by the same level of bias, irrespective of $\sigma^2_{\xi^{(c)}}$.The ratio between the two variances, $\mathrm{Var}_{\mathcal{W}}(\mu_{c',c})$ and $\mathrm{Var}_\chi(O^{(c',c)})$, quantifies the level of IGB for the subgroup of datapoints belonging to class c. Repeating the computation from App. D.1 (particularly from Eq. (62), Eq. (74), and Eq. (81)), we get:

$$\mathrm{Var}_{\mathcal{W}}(\mu_{c',c}) = \frac{\sigma_w^4 \sigma^2_{\xi^{(c)}}}{2\pi}$$

Similarly, for the variance of the output nodes (from Eq. (72), Eq. (93), and Eq. (94)), we get:

$$\mathrm{Var}_\chi(O^{(c'.c)}) = \frac{\sigma_w^4 \sigma^2_{\xi^{(c)}}(\pi - 1)}{2\pi}$$

The expressions above confirm that the ratio between the two variances is independent of $\sigma^2_{\xi^{(c)}}$, for each subgroup c.

E. Single hidden layer perceptron

In this section, we utilize the results derived in App. D to deduce our primary goal, $p_{f_0}(x)$. We specifically focus on a single hidden layer perceptron; in App. G.3 we will extend the results to deep architectures.

E.1. Derivation of $p_{f_0}(x)$

Consider a binary classification problem using a single hidden layer perceptron. We will use the notation outlined in Fig. A.1. The nodes of the hidden layer, after processing through the activation function and the pooling layer, are connected

to two output nodes $\left\{O^{(c)}\right\}_{c=0}^{1}$ via two random vectors $\left\{w_{c\cdot}^{(1)}\right\}_{c=0}^{1}$. Each output node $O^{(c)}\left(\boldsymbol{\xi};\mathcal{W}^{0},w_{c\cdot}^{(1)}\right)$ follows the distributions derived in App. D.3.

Note that, given a set $\mathcal{W}$, different output nodes depend on distinct subsets of $\mathcal{W}$, as the final matrix of weights is not shared among all nodes. The notation $O^{(c)}\left(\boldsymbol{\xi};\mathcal{W}^{0},w_{c\cdot}^{(1)}\right)$ highlights this aspect, underscoring why the output nodes have different distributions. However, for brevity, we will denote a general dependency on the entire set $\mathcal{W}$, *i.e.*, we will use the notation $O^{(c)}\left(\boldsymbol{\xi};\mathcal{W}\right)$.

To derive the distribution $p_{f_0}\left(x\right)$, we proceed as follows:

In a single experiment, the set $\left\{w_{c\cdot}^{(1)}\right\}_{c=0}^{1}$ is fixed, allowing us to compute $p_{O^{(c)}}^{(\chi)}\left(x\right)$. Recalling that f_0 represents the fraction of the dataset classified as belonging to class 0, according to the *Law of Large Numbers*, this fraction will converge to the probability

$$\lim_{D\to\infty} f_0\left(\mathcal{W}\right)=\mathbb{P}\left(O^{(0)}>O^{(1)}\mid\mathcal{W}\right)=\mathbb{P}\left(O^{(0)}-O^{(1)}>0\mid\mathcal{W}\right)=\int_0^\infty p_{\Delta_O}^{(\chi)}\left(z\right)\,dz, \tag{99}$$

where $f_0\left(\mathcal{W}\right)$ indicates the fraction f_0 obtained for the specific fixed configuration $\mathcal{W}$.

f_0 is a quantity clearly dependent on the set of random variables $\mathcal{W}$. Consequently, it is a random variable itself, varying with the network's set of weights. Defining $\Delta_O=O^{(0)}-O^{(1)}$, we can express:

$$p_{f_0}\left(x\right)=\int_\Omega p_{\mathcal{W}}\left(\tilde{\mathcal{W}}\right)\delta\left(\mathbb{P}\left(\Delta_O>0\mid\tilde{\mathcal{W}}\right)-f_0\right)d\tilde{\mathcal{W}} \tag{100}$$

where Ω represents the space of all possible initial configurations.

Eq. (100) provides an expression for $p_{f_0}\left(x\right)$. However, computing the integral on the right-hand side (r.h.s.) is not straightforward. To circumvent this, we introduce a second approach that is asymptotically exact in the limit of $N_1\to\infty$.

In App. D.3, we derived expressions for $p_{O^{(c)}}^{(\chi)}\left(x\right)$ for different cases. These distributions are generally Gaussian, with cumulants dependent on $S_{\boldsymbol{w}_c}\equiv\sum_j^{N_1}w_{cj}^{(1)}$ and $S_{\boldsymbol{w}_c^2}\equiv\sum_j^{N_1}\left(w_{cj}^{(1)}\right)^2$. The standard deviations can be viewed as estimates for the magnitude of fluctuations from the mean value.

In App. C.2.1, we show that:

$$\sqrt{\overline{S_{\boldsymbol{w}_c}-\overline{S_{\boldsymbol{w}_c}}^2}}=\mathcal{O}(1), \tag{101}$$

$$\sqrt{\overline{S_{\boldsymbol{w}_c^2}-\overline{S_{\boldsymbol{w}_c^2}}^2}}=\mathcal{O}\left(\frac{1}{\sqrt{N_1}}\right). \tag{102}$$

In other words, the distribution of $S_{\boldsymbol{w}_c}$ remains asymptotically stable, whereas the distribution of $S_{\boldsymbol{w}_c^2}$ concentrates around its mean value as we approach the asymptotic limit. This distinction in the scaling of fluctuations between the two cases will be significant in our subsequent discussion.

To establish a unified notation, we can generalize from App. D.3 that:

$$p_{O^{(c)}}^{(\chi)}\left(x\right)=\mathcal{N}\left(x;\mu\left(S_{\boldsymbol{w}_c}\right),\sigma^2\left(S_{\boldsymbol{w}_c^2}\right)\right), \tag{103}$$

where $\mu\left(S_{\boldsymbol{w}_c}\right)$ and $\sigma\left(S_{\boldsymbol{w}_c^2}\right)$ are functions of the set of *r.v.*s $\{w_{cj}^{(1)}\}_{j=1}^{N_1}$, contingent on the specific setting (e.g., the choice of activation function). App. D.2 provides the distribution for $\mu\left(S_{\boldsymbol{w}_c}\right)$ in various scenarios. Conversely, from Eq. (102), we know that the distribution of $\sigma\left(S_{\boldsymbol{w}_c^2}\right)$ becomes increasingly narrow around its mean values as $N_1\to\infty$, thus converging to a deterministic quantity.

Since the distribution of $\mu\left(S_{\boldsymbol{w}_c}\right)$ does not narrow asymptotically around a deterministic value, $p_{O^{(c)}}^{(\chi)}\left(x\right)$ will differ for various nodes, $\{O^{(c)}\}$, even in the limit of infinite size. We refer to the emergence of this discrepancy in distributions

among nodes of the same layer as *Node Symmetry Breaking* (NSB). Section 4 elucidates how NSB and IGB are interrelated phenomena, with IGB being a direct consequence of this symmetry breaking. In Sec. 5.2, we will delve deeper into NSB, illustrating how this phenomenon manifests not only in output layers but also in intermediate hidden layers of deep architectures.

Returning to the initial problem, let's consider a given initialization fixing the configuration $\mathcal{W}$, and in particular, the two random vectors $w_{0\cdot}^{(1)}$ and $w_{1\cdot}^{(1)}$. This leads to two corresponding distributions for $p_{O^{(0)}}^{(\chi)}(x)$ and $p_{O^{(1)}}^{(\chi)}(x)$. To calculate the probability that $O^{(0)} > O^{(1)}$, we first transition to the distribution $p_{\Delta_O}^{(\chi)}(x)$. Since Δ_O is the difference between two Gaussian random variables, it follows that:

$$p_{\Delta_O}^{(\chi)}(x) = \mathcal{N}\left(x; \mu\left(S_{\boldsymbol{w}_0}\right) - \mu\left(S_{\boldsymbol{w}_1}\right), \sigma^2\left(S_{\boldsymbol{w}_0^2}\right) + \sigma^2\left(S_{\boldsymbol{w}_1^2}\right)\right). \tag{104}$$

From Eq. (102), we understand that the distribution of $S_{\boldsymbol{w}_c^2}$ narrows around its mean in the limit $N_1 \to \infty$, allowing us to disregard the fluctuations of the random variable $\sigma^2\left(S_{\boldsymbol{w}_c^2}\right)$. Substituting $\sigma^2\left(S_{\boldsymbol{w}_c^2}\right) = \sigma^2\left(\overline{S_{\boldsymbol{w}_c^2}}\right) \equiv \sigma_\infty^2$, we obtain:

$$p_{\Delta_O}^{(\chi)}(x) \xrightarrow{N_1 \to \infty} \mathcal{N}\left(x; \mu\left(S_{\boldsymbol{w}_0}\right) - \mu\left(S_{\boldsymbol{w}_1}\right), 2\sigma_\infty^2\right). \tag{105}$$

This convergence is pivotal to eliminate the explicit dependence on the random vectors $\{w_{c\cdot}^{(1)}\}$ and to avoid integration over all possible configurations. Once we have an explicit form for the distribution of $\mu\left(S_{\boldsymbol{w}_c}\right)$ (see App. D.2), and defining

$$\Delta_\mu = \mu\left(S_{\boldsymbol{w}_0}\right) - \mu\left(S_{\boldsymbol{w}_1}\right), \tag{106}$$

we can find an implicit expression for Δ_μ, *i.e.*,

$$f_0(\mathcal{W}) = \int_0^\infty \mathcal{N}\left(y; \Delta_\mu(f_0), 2\sigma_\infty^2\right) dy. \tag{107}$$

Given the centers of the two Gaussian distributions $p_{O^{(c)}}^{(\chi)}(x)$, specifically their difference Δ_μ, Eq. (107) provides the corresponding value of f_0. Inverting Eq. (107) numerically yields $\Delta_\mu(f_0)$ associated with a given value of f_0.

Notably, $\mu\left(S_{\boldsymbol{w}_c}\right) \sim p_{\mu_c}(x)$ is a *r.v.*; consequently, $\Delta_\mu(f_0)$ is a *r.v.* as well. From the monotonicity of the relationship $\Delta_\mu(f_0)$, it follows that

$$p_{f_0}(x)\, dx = p_{\Delta_\mu(f_0)}\left(\Delta_\mu(x)\right)\, d\Delta_\mu \tag{108}$$

Since $\mu\left(S_{\boldsymbol{w}_0}\right)$ and $\mu\left(S_{\boldsymbol{w}_1}\right)$ are i.i.d. random variables, we can deduce:

$$p_{\Delta_\mu(f_0)}\left(\Delta_\mu(x)\right) = \int_{-\infty}^{\infty} p_{\mu_1}(\tilde{x})\, p_{\mu_0}\left(\tilde{x} + \Delta_\mu(x)\right) d\tilde{x} = \int_{-\infty}^{\infty} p_{\mu_c}(\tilde{x})\, p_{\mu_c}\left(\tilde{x} + \Delta_\mu(x)\right) d\tilde{x}, \tag{109}$$

where the integral arises from the fact that we only enforce a condition on the difference between the nodes' mean values, which is invariant under a translation of both values. The final equality leverages the identical distribution of μ_0 and μ_1.

Given that $p_{\mu_c}(x) \xrightarrow{N_1 \to \infty} \mathcal{N}\left(x; 0, \hat{\sigma}_\infty^2\right)$,[10] Eq. (109) represents an integral of a product of two independent Gaussian distributions.[11] Solving this integral gives us

$$p_{\Delta_\mu(f_0)}\left(\Delta_\mu(x)\right) = \mathcal{N}\left(\Delta_\mu(x); 0, 2\hat{\sigma}_\infty^2\right). \tag{110}$$

Remark 2. While Eq. (110) illustrates a Gaussian distribution in terms of the variable $\Delta_\mu(f_0)$, it is important to note that $p_{f_0}(x)$ will not generally be Gaussian since $\Delta_\mu(f_0)$ is a non-linear function.

[10] The quantity $\hat{\sigma}_\infty^2$ is model-specific; for more details, see App. D.2.

[11] The product of Gaussian distributions itself is proportional to a Gaussian distribution.

max pool peaks: A practical example for computing $p_{f_0}(x)$ Utilizing the results from App. E.1, we can calculate the theoretical prediction for $p_{f_0}(x)$, particularly focusing on the probability density at extreme peaks empirically observed with max pooling.

Suppose we have a dataset of D elements, and we aim to calculate the probability density $p_{f_0}(0)$. The steps involved are:

- **Computing $\Delta^{(T)}$:** First, we need to invert Eq. (107):

$$f_0 = 1 - \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\Delta_\mu(f_0)}{2\sigma_\infty}\right)\right) \Rightarrow \Delta_\mu(f_0) = 2\sigma_\infty\,\mathrm{erf}^{-1}\left(2f_0 - 1\right). \tag{111}$$

  $\Delta^{(T)}$ is the threshold value; for $\Delta_\mu(f_0) < \Delta^{(T)}$, Eq. (107) yields a value below $\frac{1}{D}$, corresponding to $f_0 = 0$ due to the dataset size D resolution.

- **Integrating Eq. (110):** Next, we integrate over all $\Delta_\mu(f_0) < \Delta^{(T)}$:

$$p_{f_0}(0) = \int_{-\infty}^{\Delta^{(T)}} \mathcal{N}\left(\Delta_\mu(x); 0, 2\hat{\sigma}_\infty^2\right) d\Delta_\mu(x) = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\Delta^{(T)}}{2\hat{\sigma}_\infty}\right). \tag{112}$$

Generalizing for arbitrary probability mass: For the probability mass between $f_0^{(min)}$ and $f_0^{(max)}$, the computation is:

$$\int_{f_0^{(min)}}^{f_0^{(max)}} p_{f_0}(x)\,dx = \int_{\Delta_\mu\left(f_0^{(min)}\right)}^{\Delta_\mu\left(f_0^{(max)}\right)} \mathcal{N}\left(\Delta_\mu(x); 0, 2\hat{\sigma}_\infty^2\right) d\Delta_\mu(x) = \frac{1}{2}\,\mathrm{erf}\left(\frac{\Delta_\mu\left(f_0^{(max)}\right)}{2\hat{\sigma}_\infty}\right) - \frac{1}{2}\,\mathrm{erf}\left(\frac{\Delta_\mu\left(f_0^{(min)}\right)}{2\hat{\sigma}_\infty}\right). \tag{113}$$

These computations facilitate a precise understanding of the impact of max pooling on f_0, especially in terms of evaluating probability densities at the extreme peaks.

E.2. Beyond f_0: relating the confidence in model assignments to IGB

While our discussion is based on an intuitive and clearly interpretable measure, namely f_0 and its distribution, this measure has a significant limitation. f_0 indicates the fraction of the dataset assigned to class 0 by the network at initialization, but it doesn't provide information about the average confidence with which this assignment is made for a given dataset element.

It is important to note that, although f_0 is a central focus of our narrative for its intuitive clarity, it is not the main focus of our analysis. Our analyses are instead centered around the output nodes, from which we derive distributions, and only subsequently, through these and Eq. (3), we transition to the variable f_0.
Therefore, we can leverage our knowledge of the output nodes' statistics to investigate different measures. Typically, to convert the outputs into probabilities or, to better say, into confidence levels, a softmax transformation is applied, defined as:

$$O^{(c)} \to C_{O^{(c)}} \equiv \frac{e^{O^{(c)}}}{\sum_i e^{O^{(i)}}}. \tag{114}$$

We note that this metric provides additional insights, leading to more stringent conditions than just the absence of IGB. Specifically, when every element of the dataset fulfills the condition

$$C_{O^{(c)}} = \frac{1}{N_C}\ \forall c, \tag{115}$$

it implies not only the absence of IGB (wherein the DNN's classifications distribute the dataset evenly across classes), but also that every assignement has an equal probability for each class. This leads us to the concept of Strong Absence of IGB, which we define as follows:

Definition E.1 (Strong Absence of IGB)**.** Given a dataset χ (or its pre-processed version $\psi(\chi)$) and an architecture $\mathcal{A}$, there is a strong absence of Initial Guessing Bias (IGB) if, for every element in the dataset and for every initialization of weights *w.h.p.*, the model predicts each class with the same level of confidence, i.e., Eq. (115) is satisfied.

With binary classification, we can, in general, consider a measure to evaluate the gap between assignment probabilities, defined as:

$$R_C \equiv \frac{C_{O^{(0)}}}{C_{O^{(1)}}} = e^{\Delta_O} \,, \tag{116}$$

where we remind the reader that $\Delta_O \equiv O^{(0)} - O^{(1)}$.

Since we have shown that the distribution of a generic output, $p^{(\chi)}_{O^{(c)}}(x)$, is Gaussian (see Eq. (58)), it follows that

$$p^{(\chi)}_{\Delta_O}(x) = \mathcal{N}\left(x; \Delta_\mu, 2\mathrm{Var}_\chi\left(O^{(c)}\right)\right) \,, \tag{117}$$

where $\Delta_\mu \equiv \mu_0 - \mu_1$. With a given initialization $\mathcal{W}$, the quantity Δ_μ is deterministically fixed. Combining Eqs. (116) and (117), we can explicitly write R_C as a log-normal distribution:

$$p^{(\chi)}_{R_C}(x) = \frac{1}{\Delta_\mu \sqrt{4\pi \mathrm{Var}_\chi\left(O^{(c)}\right)}} e^{-\frac{(\ln(x)-\Delta_\mu)^2}{4\mathrm{Var}_\chi\left(O^{(c)}\right)}} \,. \tag{118}$$

The first two cumulants (over the distribution of the data) of $p^{(\chi)}_{R_C}(x)$ are:

$$\langle R_C \rangle = e^{\Delta_\mu + \mathrm{Var}_\chi\left(O^{(c)}\right)} \,, \tag{119}$$

$$\mathrm{Var}_\chi(R_C) = e^{2\Delta_\mu}\left(e^{2\mathrm{Var}_\chi\left(O^{(c)}\right)} - 1\right) e^{2\mathrm{Var}_\chi\left(O^{(c)}\right)} \,. \tag{120}$$

Equations (119) and (120) demonstrate that:

$$\lim_{\Delta_\mu \to 0} \lim_{\mathrm{Var}_\chi\left(O^{(c)}\right)\to 0} \langle R_C \rangle = 1 \,, \tag{121}$$

$$\lim_{\mathrm{Var}_\chi\left(O^{(c)}\right)\to 0} \mathrm{Var}_\chi(R_C) = 0 \,. \tag{122}$$

In this limit, not only do we have an absence of IGB, but the neural network assigns each input to either of the two classes with the same probability (since the distribution concentrates around a single value), *i.e.* we have strong absence of IGB. This constitutes the most extreme limit case: in the limit $D \to \infty$, half of the dataset will be assigned to each class, and each element will be assigned to one of the two classes with the same probability ($R_C = 1$).

We may encounter a scenario where $\Delta_\mu = 0$ and $\mathrm{Var}_\chi\left(O^{(c)}\right) \neq 0$. Even though this indicates an absence of IGB, the distribution for R_C is non-degenerate. Like the previous case, the dataset will be evenly divided in assignments between the different classes. However, the typical confidence is different for each class (it is described by Eq. (118)).

In the presence of IGB, Δ_μ will have a non-degenerate distribution (see Eq. (59)). As the level of IGB increases (by increasing $\mathrm{Var}_\mathcal{W}(\mu_c)$), there will also be an increase in $\overline{\langle R_C \rangle}$, meaning the typical gap between the two assignment probabilities for a generic element will widen.
It is important to note that, given a specific setting $(\mathcal{A}, \psi(\chi))$, we can explicitly calculate Eq. (119) and Eq. (120) based on the output statistics and evaluate the network's confidence in its assignments.

In conclusion, the approach shown in this section paves the way for more accurate analyses. For example, knowing the $\{C_{O^{(c)}}\}$, we could calculate the entropy $H\left(\{C_{O^{(c)}}\} \mid \mathcal{W}, \chi\right)$ and reformulate the absence of IGB as a principle of maximum entropy. However, it is important to note that the presence of the normalization factor complicates the derivation of the distribution of $C_{O^{(c)}}$, which can be avoided by using R_C. This also provides a natural variable to encapsulate the gap between the assignment variables. Finally, although we focused on the binary problem for simplicity, extensions to the multi-class case are possible using approaches similar to those described in App. H, for example by considering the ratio of the two classes with the highest confidence.

E.2.1. EFFECT OF THE SOFTMAX TEMPERATURE

Eq. (114) is the standard softmaxed expression of the output with the temperature parameter $T = 1$. Acting on the temperature can effectively reduce IGB. By applying a high-temperature parameter, we observe a narrowing distribution of R_C around $R_C = 1$, indicating a reduction in IGB as $T \to \infty$. However, it is crucial to acknowledge the complex effects of temperature on training dynamics; specifically, a small beta value (high temperature) may affect learning stability (Agarwala et al., 2020). While further investigation is needed to fully understand the impact of this mitigation strategy on the training process.

F. Conditions for the emergence/absence of IGB

In our exploration of how activation functions can induce IGB, we have focused on specific, emblematic examples, such as the linear activation function and the ReLU. These were chosen for their analytical tractability. However, the underlying principles extend beyond these cases. In this section, we'll broaden the discussion, classifying activation functions into distinct categories based on their influence on IGB.

The crucial distinction between the activation functions we've discussed can be summarized as follows:

1. With a linear activation function, the output nodes are asymptotically identically distributed. This uniformity leads to the absence of IGB. 2. In contrast, the ReLU activation function introduces a symmetry break in the output nodes. While these nodes remain asymptotically Gaussian, they are centered at different points, leading to the emergence of IGB.

This symmetry breaking is intricately linked to how the ReLU, applied at the first hidden layer, transforms null-averaged inputs into random variables, $g_i^{(1)}$, with a non-zero mean.

To generalize our findings, we'll now examine a typical activation function from each category and demonstrate how the presence or absence of IGB manifests. Our analysis will maintain the same foundational assumptions as before: Gaussian-distributed *i.i.d.* inputs and Kaiming initialization for the weights. As established in App. D.2, regardless of the activation function, we observe that:

$$p^{(\chi)}_{h_i^{(1)}}(x) \xrightarrow{d\to\infty} \mathcal{N}\left(x; 0, \sigma_w^2\right). \tag{123}$$

The forthcoming analysis will elucidate the conditions under which different types of activation functions either foster or mitigate the occurrence of IGB in neural network models.

F.1. Non-null mean activation function

Let us consider a generic activation function $g(\cdot)$, such that

$$\left\langle g_i^{(1)} \right\rangle \equiv \left\langle g\left(h_i^{(1)}\right) \right\rangle = \left\langle g^{(1)} \right\rangle \neq 0. \tag{124}$$

where the second step follow from the fact that $\{h_i^{(1)}\}$ are identically distributed. Starting from this hypothesis on the activation function (Eq. (124)) we can follow exactly the same analysis presented in App. G.3.1. In particular:

$$p^{(\chi)}_{h_i^{(2)}}(x) \xrightarrow{N_1\to\infty} \mathcal{N}\left(x; \left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}, \mathrm{Var}_\chi\left(g^{(1)}\right) S_{\boldsymbol{w}_i^2}\right) = \mathcal{N}\left(x; \left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}, \mathrm{Var}_\chi\left(g^{(1)}\right) \overline{S_{\boldsymbol{w}_i^2}}\right), \tag{125}$$

From (125) we can already see the nodes symmetry breaking; the centers of the distributions is a *r.v.* varying from node to node. Note that for a single hidden layer architecture

$$p^{(\chi)}_{O^{(c)}}(x) = p^{(\chi)}_{h_c^{(2)}}(x). \tag{126}$$

For a generic deep architecture we can keep following the analysis of App. G.3.1, substituiting the right expression to $p^{(\chi)}_{g_i^{(2)}}(x)$ dependent to the specific used activation function. Proceeding in this way we will find, for each layer, a symmetry breaking in the nodes distribution, similarly to what we observed in the second layer (Eq. (125)).

F.2. Null mean activation function

We now explore the converse scenario, namely an activation function $g(\cdot)$ satisfying:

$$\left\langle g_i^{(1)} \right\rangle \equiv \left\langle g\left(h_i^{(1)}\right) \right\rangle = \left\langle g^{(1)} \right\rangle = 0. \tag{127}$$

This equality arises because the set $\{h_i^{(1)}\}$ is identically distributed. Unlike the non-null mean case ((125)), this scenario leads to:

$$p^{(\chi)}_{h_i^{(2)}}(x) \xrightarrow{N_1 \to \infty} \mathcal{N}\left(x; 0, \mathrm{Var}_\chi\left(g^{(1)}\right) S_{\boldsymbol{w}_i^2}\right) = \mathcal{N}\left(x; 0, \mathrm{Var}_\chi\left(g^{(1)}\right) \overline{S_{\boldsymbol{w}_i^2}}\right). \tag{128}$$

Here, the absence of node symmetry breaking is evident. Proceeding to subsequent layers, we find a similar pattern of asymptotically identically distributed nodes. This iterative process ultimately extends to the output nodes, which will also be equally distributed.
Determining whether a generic activation function falls into one of the two categories discussed may not be straightforward. To illustrate, we present an example set of functions, all of which fulfill Condition (127).

F.2.1. Anti-Symmetric activation functions

The previous discussions have underscored how the choice of activation function significantly influences a network's behavior, notably in the context of IGB (refer to Fig. 5). While the trivial identity function does not exhibit IGB, this section introduces a broader class of activation functions, the anti-symmetric activation functions, which similarly do not exhibit IGB. Interestingly, the identity function is a member of this class.

We begin by reinterpreting the concept of IGB within the context of our analysis:

Definition F.1 (IGB)**.** Assume $p^{(\chi)}_{O^{(c)}}(x)$ asymptotically converges to a Gaussian distribution with non-zero variance (see App. D.3). If

$$p_{\mu_c}(x) = \delta(x - a), \ \forall c, \tag{129}$$

for some $a \in \mathbb{R}$, we have an absence of IGB. Conversely, if condition (129) is not satisfied, IGB emerges, leading to disproportionate values of $\{f_i\}$ even in the infinite dataset size limit, thus excluding finite size effects.

To illustrate the concept with a practical example, consider an activation function defined as

$$g_i^{(l)} = g\left(h_i^{(l)}\right) = \tanh\left(h_i^{(l)}\right), \ l \in \{0, \ldots, L\}, \ i \in \{0, \ldots, N_l\}. \tag{130}$$

Recalling Eq. (62), the asymptotic distribution of $p^{(\chi)}_{h_i^{(1)}}(x)$ is known and remains independent of the activation function choice. The question arises: "How is this distribution transformed after passing through the activation function (Eq. (130))?"

By employing the relation

$$p_X(x)\,dx = p_Y(y(x))\,dy, \tag{131}$$

we derive

$$p^{(\chi)}_{g_i^{(1)}}(y) = \frac{e^{-\frac{(\mathrm{atanh}(y))^2}{2\sigma^2_{h^{(1)}}}}}{\sqrt{2\pi\sigma^2_{h^{(1)}}}} \frac{1}{(1-y^2)}. \tag{132}$$

We observe from Eq. (132) that the distribution is symmetric, i.e., $p^{(\chi)}_{g_i^{(1)}}(y) = p^{(\chi)}_{g_i^{(1)}}(-y)$. This symmetry arises directly from the anti-symmetric nature of the activation function combined with the symmetry of $p^{(\chi)}_{h_i^{(1)}}(x)$. This would also hold true for other anti-symmetric activation functions.

As a result of this symmetry, we have:

$$\left\langle g_i^{(1)} \right\rangle = 0 \Longrightarrow \left\langle h_j^{(2)} \right\rangle = 0, \quad \forall i, j. \tag{133}$$

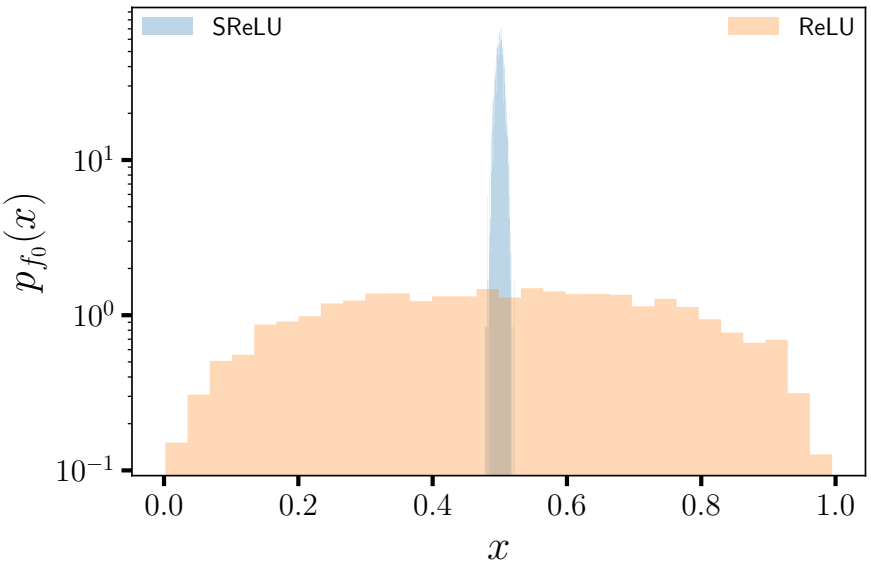

Figure F.1. Contrast between SReLU and ReLU. Shifting ReLU to meet Eq. (128) conditions eliminates IGB. The figure shows the differing regimes through $p_{f_0}(x)$. Dataset `GB` and model `SHLP` were used for these simulations (see App. I.4 for details).

This is a stark contrast to the scenarios discussed in the previous section, as it means all nodes $\{h_j^{(2)}\}$ are identically distributed. Specifically,

$$p_{\left\langle h_i^{(2)} \right\rangle}(x) = \delta(x) \ \forall i. \tag{134}$$

As we continue to propagate through the network's remaining hidden layers, this pattern persists. Given the similarity between $p^{(\chi)}_{h_i^{(2)}}(x)$ and $p^{(\chi)}_{h_i^{(1)}}(x)$, the same reasoning can be applied layer by layer, leading to

$$p_{\left\langle h_i^{(3)} \right\rangle}(x) = \delta(x) \ \forall i, \tag{135}$$

and this holds for all $l \in \{3, \ldots, L+1\}$.

Specifically, for $l = L+1$, it implies that

$$p_{\mu_c}(x) = \delta(x) \ \forall c. \tag{136}$$

Therefore, by employing an anti-symmetric activation function such as $\tanh$, IGB is not observed, under the assumption that the chosen initialization maintains a non-zero asymptotic variance for $p^{(\chi)}_{O^{(c)}}(x)$.

F.3. Eliminate/Trigger IGB with a generic activation function

The preceding discussion reveals that activation functions can be divided into two categories based on their influence on IGB. These categories are defined by simple attributes (*e.g.*, Eq. (124) and Eq. (127)). This insight suggests that modifying an existing activation function, or applying certain regularizations, can switch a network from exhibiting IGB to one that doesn't, and vice versa.

This concept aligns with strategies used in machine learning, such as Batch Normalization introduced in Ioffe & Szegedy (2015), which is employed to mitigate the *Internal Covariate Shift*. While Batch Normalization addresses shifts during training, our focus is on initialization-induced node differences. Nevertheless, our analysis suggests that a similar scaling approach could effectively manage IGB as well.

To illustrate, consider the ReLU activation function, which induces IGB. We introduce a shifted version of ReLU, named

SReLU, defined as follows:

$$g\left(h_i^{(1)}\right) = \begin{cases} h_i^{(1)} - \frac{\sigma_w}{\sqrt{2\pi}}, & \text{if } h_i^{(1)} > 0\,, \\ -\frac{\sigma_w}{\sqrt{2\pi}} & \text{if } h_i^{(1)} < 0\,, \end{cases} \tag{137}$$

where $\frac{\sigma_w}{\sqrt{2\pi}}$ is the mean value of ReLU, calculated from Eq. (80). SReLU, satisfying Eq. (127), falls into the subset of functions that do not exhibit IGB (see Fig. F.1).

While the expectation value is easily computed and subtracted in this example, for a generic activation function, a similar shift could be based on empirical averages computed from the forwarding batch, akin to the approach in Ioffe & Szegedy (2015).

G. Amplification of the IGB level

The previous sections focused on clarifying the conditions that ensure the emergence of IGB. However, as highlighted by the comparison in Fig. 5, the presence of IGB is not uniform across all settings. This section aims to clarify the conditions that can lead to an amplification of IGB. To quantify the level of IGB, we will refer to the measure introduced in Eq. (8), analyzing settings in which $\gamma(\mathcal{A}, \psi(\chi))$ diverges. In the following sections, we will analyze different settings and limits leading to this divergence.

G.1. Effect of data standardization

In this section, we discuss the relationship between dataset standardization and IGB. We begin by showing that this element alone is sufficient to induce the emergence of the phenomenon, irrespective of the architectural design choice (discussed in previous and subsequent sections). Specifically, we focus here on a particular standardization procedure, which involves simply recentering the input data by adding a constant, $K \in \mathbb{R}$, to each component. To be more precise, the modification from the original setting consists of considering the pre-processed data

$$\psi\left(\xi_b^{(a)}\right) = \xi_b^{(a)} + K\,, \tag{138}$$

with $\xi_b^{(a)} \sim \mathcal{N}(0,1)$. We then consider an architectural setting without max pooling and with a linear activation function. As seen in App. D.2, this architectural setting does not lead to the emergence of IGB using independent inputs centered at 0. Repeating the same calculation using the preprocessed dataset, $\psi(\chi)$, according to Eq. (138) we can instead show the appearance of IGB. Indeed, starting from Eq. (75) we have

$$\left\langle \rho_i^{(1;1)} \right\rangle = \left\langle h_i^{(1)} \right\rangle = \left\langle \sum_b (\xi_b^{(a)} + K) w_{ib}^{(0)} \right\rangle = K \sum_b w_{ib}^{(0)}, \tag{139}$$

or

$$\left\langle \rho_i^{(1;1)} \right\rangle = \left\langle h_i^{(1)} \right\rangle \sim \mathcal{N}\left(0, K^2 \sigma_w^2\right)\,. \tag{140}$$

Proceeding in a manner similar to the derivation of Eq. (40)

$$p_{\mu_c}(x) = \mathcal{N}\left(x; \sum_j \overline{w_{ci}^{(1)}} \left\langle h_i^{(1)} \right\rangle, \sum_j \mathrm{Var}_{\mathcal{W}}\left(w_{ij}^{(l)}\right) \left\langle h_i^{(1)} \right\rangle^2\right) = \mathcal{N}\left(x; 0, \sigma_w^2 \frac{1}{N_1} \sum_{j=1}^{N_1} \left\langle h_i^{(1)} \right\rangle^2\right)\,, \tag{141}$$

which, combined with Eq. (140), leads us to

$$\lim_{N_1 \to \infty} p_{\mu_c}(x) = \mathcal{N}\left(x; 0, \sigma_w^4 K^2\right)\,. \tag{142}$$

Note that a similar result would have been obtained by using a generic non-zero vector for standardization, i.e.,

$$\psi\left(\xi_b^{(a)}\right) = \xi_b^{(a)} + K_b.$$

The only difference would have been in Eq. (139), where the index dependence would have prevented the factorization of the generic term K_b outside the summation. This would have resulted in a linear combination of Gaussian variables (instead of a simple scaled sum of Gaussian variables), which similarly still follows a Gaussian distribution. Finally, we would have obtained a result analogous to Eq. (140), with $K^2 \equiv |\boldsymbol{K}|^2 = \sum_b k_b^2$.

Therefore, the distribution of the output node centers is non-degenerate, thus indicating the presence of IGB.

After demonstrating the emergence of IGB, we show how an increase in $|K|$ leads to the amplification of IGB. To measure the level of IGB, we consider the measure $\gamma(\mathcal{A}, \psi(\chi))$ defined in Eq. (8). In particular, proceeding in a manner similar to Eq. (39), it is straightforward to show that:

$$p_{O^{(c)}}^{(\chi)}(x) = \mathcal{N}\left(x; \mu_c, \sum_j \left(w_{ij}^{(1)}\right)^2 \mathrm{Var}_\chi\left(h_j^{(1)}\right)\right). \tag{143}$$

From this, it follows that

$$\lim_{d, N_1 \to \infty} \mathrm{Var}_\chi\left(O^{(c)}\right) = \sigma_w^4 \,. \tag{144}$$

Finally, combining Eq. (141) with Eq. (144), we have

$$\lim_{|K| \to \infty} \gamma(\mathcal{A}, \psi(\chi)) = \lim_{|K| \to \infty} \frac{K^2 \sigma_w^4}{\sigma_w^4} = \infty \,. \tag{145}$$

Thus, the ratio $\frac{\mathrm{Var}_{\mathcal{W}}(\mu_c)}{\mathrm{Var}_\chi\left(O^{(c)}\right)}$ diverges as $|K|$ increases, indicating a significant amplification of IGB.

Building on this result, we can prove one of the claims of Thm. 5.2 (Eq. (18)), which is reformulated more precisely as follows:

Theorem G.1. *Consider a dataset with Gaussian-distributed, i.i.d. components, where $\xi_b^{(a)} \sim \mathcal{N}(0,1)$, and $\xi_b^{(a)}$ represents the b-th component of the a-th input vector. This dataset is processed through an MLP with a single hidden layer ($L = 1$). The MLP maps the preprocessed input, shifted by a constant vector $\boldsymbol{K}$, i.e., $\psi\left(\xi_b^{(a)}\right) = \xi_b^{(a)} + K_b$, to the output as per Eq. (21), Eq. (22), and Eq. (23). The weights are initialized following the Kaiming normal scheme, specifically $w_{ij}^{(l)} \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{N_{l-1}}\right)$, with zero biases. Focusing on a setting without a pooling layer and employing ReLU as the activation function, an increase in the norm of $\boldsymbol{K}$ leads to an asymptotic amplification of the Initial Guessing Bias (IGB). Formally:*

$$\lim_{|\boldsymbol{K}| \to \infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty. \tag{146}$$

Proof. Following a similar approach as outlined in App. C.2.1, particularly considering Eq. (40) and Eq. (41), in our specific case (*i.e.*, with $h_c^{(l+1)} = O^{(c)}$), we obtain:

$$\lim_{N_1 \to \infty} \mathrm{Var}_{\mathcal{W}}(\mu_c) = \sigma_w^2 \int_{-\infty}^{\infty} \left\langle g_j^{(l)} \right\rangle^2 p_{\left\langle h_j^{(1)} \right\rangle}(x)\, dx \,, \tag{147}$$

where the integral on the r.h.s. represents the expectation of $\left\langle g_j^{(l)} \right\rangle^2$ over the population $\left\{ \left\langle g_j^{(1)} \right\rangle \right\}$; specifically, as $\left\langle g_j^{(l)} \right\rangle$

is a function of $\left\langle h_i^{(1)} \right\rangle$, we can express the mean in terms of the measure of the latter. Starting from Eq. (147), we have

$$\lim_{N_1 \to \infty} \mathrm{Var}_{\mathcal{W}}\left(\mu_c\right) \overset{\boldsymbol{a}}{>} \sigma_w^2 \int_0^\infty \left\langle g_j^{(l)} \right\rangle^2 p_{\left\langle h_j^{(1)} \right\rangle}(x)\, dx \overset{\boldsymbol{b}}{>} \sigma_w^2 \int_0^\infty \left\langle h_j^{(1)} \right\rangle^2 p_{\left\langle h_j^{(1)} \right\rangle}(x)\, dx \overset{\boldsymbol{c}}{=} \tag{148}$$

$$= \frac{\sigma_w^2}{2} \int_{-\infty}^\infty \left\langle h_j^{(1)} \right\rangle^2 p_{\left\langle h_j^{(1)} \right\rangle}(x)\, dx \overset{\boldsymbol{d}}{=} \frac{\sigma_w^4}{2} |\boldsymbol{K}|^2 , \tag{149}$$

where inequality $\boldsymbol{a}$ follows from the positivity of the integral in Eq. (147), while inequality $\boldsymbol{b}$ derives from the fact that the average value of a rectified Gaussian is, by construction, greater than the mean value of the original Gaussian; equality $\boldsymbol{c}$ follows from the parity of the integrand, and finally, equality $\boldsymbol{d}$ is deduced from the analysis on the linear system (particularly Eq. (142)).

Since, by construction, the variance of a rectified Gaussian is always less than that of the original Gaussian, starting from Eq. (39), applying expectation over the population of activated hidden layer nodes (as just done) and using the result derived for the linear architecture (Eq. (144)), we have:

$$\lim_{d, N_1 \to \infty} \mathrm{Var}_\chi\left(O^{(c)}\right) < \sigma_w^4 . \tag{150}$$

Finally, combining Eq. (148) and Eq. (150), it follows that

$$\lim_{|\boldsymbol{K}| \to \infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty . \tag{151}$$

□

G.2. Effect of max pooling

Theorem G.2. *Consider a dataset with Gaussian-distributed, i.i.d. components, where $\xi_b^{(a)} \sim \mathcal{N}(0,1)$, and $\xi_b^{(a)}$ represents the b-th component of the a-th input vector. Inputs are processed through an MLP with a single hidden layer ($L = 1$), mapping the input to output according to Eq. (21), Eq. (22), and Eq. (23). The weights are initialized following the Kaiming normal scheme, i.e., $w_{ij}^{(l)} \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{N_{l-1}}\right)$, with zero bias weights. In a setting with max pooling layer and ReLU activation function, an increase in the kernel size, m, leads to an asymptotic amplification of the Initial Guessing Bias (IGB). Formally:*

$$\lim_{m \to \infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty. \tag{152}$$

Proof. Consider the set $\{g_i^{(1)}\}_{i=1}^m$ such that

$$p_{g_i^{(1)}}^{(\chi)}(x) = \frac{1}{2}\delta(x) + \Theta(x)\mathcal{N}\left(x; 0, \sigma_w^2\right), \tag{153}$$

for a fixed set $\mathcal{W}$. Now, define the set of random variables $\{\omega_i\}_{i=1}^m$, where $n = \log(m)$, as follows:

$$\omega_i \equiv \frac{g_i^{(1)}}{\sqrt{\log(m)}} = \frac{g_i^{(1)}}{\sqrt{n}}. \tag{154}$$

Utilizing the relation

$$p_\omega^{(\chi)}(\omega(x))\, d\omega = p_{g^{(1)}}^{(\chi)}(x)\, dg^{(1)}, \tag{155}$$

we can express

$$p_{\omega_i}^{(\chi)}(x) = \frac{1}{2}\delta(x) + \Theta(x)\mathcal{N}\left(x; 0, \frac{\sigma_w^2}{n}\right). \tag{156}$$

Now consider the number of elements in the set $\{\omega_i\}$ that fall in the interval $[\tilde{\omega}, (\tilde{\omega} + d\tilde{\omega})]$, $\tilde{\omega} > 0$, denoted as $\#(\tilde{\omega})$. Focusing on its expected value, we obtain

$$\mathbb{E}\left(\#(\tilde{\omega})\right) = m\mathbb{E}\left(\mathbb{I}(\omega \in [\tilde{\omega}, (\tilde{\omega} + d\tilde{\omega})])\right) \sim m\sqrt{\frac{n}{2\pi\sigma_w^2}} e^{-\frac{\omega^2 n}{2\sigma_w^2}} d\tilde{\omega} = \sqrt{\frac{n}{2\pi\sigma_w^2}} e^{-n\left(\frac{\omega^2}{2\sigma_w^2} - 1\right)} d\tilde{\omega}. \tag{157}$$

Furthermore, since the probability of obtaining a value from within the interval $[\tilde{\omega}, (\tilde{\omega} + d\tilde{\omega})]$, $p_{\omega}^{(\chi)}(\tilde{\omega})\, d\tilde{\omega}$, is small, it follows a Poisson law, implying that the variance is equal to the mean. Defining

$$s(\omega) \equiv \frac{\omega^2}{2\sigma_w^2} - 1, \tag{158}$$

we observe that:

- If $s(\omega) < 0$, the average number of draws with value ω is exponentially close to 0. From (158), this occurs (considering the positive interval) when $\omega > \sqrt{2\sigma_w^2}$. In this case, applying the Markov inequality, we can conclude that, *w.h.p.* the actual number of draws is effectively 0. Specifically, according to Markov inequality,

$$\mathbb{P}\left(\#(\tilde{\omega}) \geq 1\right) \leq \mathbb{E}\left(\#(\tilde{\omega})\right). \tag{159}$$

  Thus, *w.h.p.* as $n \to \infty$, all draws have values less than $\sqrt{2\sigma_w^2}$.

- If $s(\omega) > 0$, the expectation value of the number of draws is nonzero. Using the Markov inequality again, we show that, also in this scenario, the actual number of draws concentrates around the non-null expectation value as $n \to \infty$. Specifically,

$$\begin{aligned}\mathbb{P}\left(\left|\frac{\#(\tilde{\omega})}{\mathbb{E}\left(\#(\tilde{\omega})\right)} - 1\right| \geq k\right) &= \mathbb{P}\left(\left(\frac{\#(\tilde{\omega})}{\mathbb{E}\left(\#(\tilde{\omega})\right)} - 1\right)^2 \geq k^2\right) \leq \frac{\mathbb{E}\left(\left(\#(\tilde{\omega}) - \mathbb{E}\left(\#(\tilde{\omega})\right)\right)^2\right)}{k^2\left(\mathbb{E}\left(\#(\tilde{\omega})\right)\right)^2} \\ &\leq \frac{\mathrm{Var}\left(\#(\tilde{\omega})\right)}{k^2\left(\mathbb{E}\left(\#(\tilde{\omega})\right)\right)^2} \simeq \frac{\cancel{\mathbb{E}\left(\#(\tilde{\omega})\right)}}{k^2\left(\mathbb{E}\left(\#(\tilde{\omega})\right)\right)^{\cancel{2}}} \propto \frac{e^{-ns(\tilde{\omega})}}{k^2},\end{aligned} \tag{160}$$

  where we used the property of the Poisson distribution, where the variance equals the mean. As n increases, the probability of deviation from the mean exponentially decreases when $s(\tilde{\omega}) > 0$, so that *w.h.p.*, $\frac{\#(\tilde{\omega})}{\mathbb{E}(\#(\tilde{\omega}))}$ is arbitrarily close to 1.

More refined analyses are possible to characterize the extreme value statistics. For further information on this topic, refer to Schehr & Majumdar (2014) and the references therein. However, for the purposes of our demonstration, we have sufficient elements to proceed.

We have shown that, as $m \to \infty$, $\max\{\omega_i\}$ converges to a deterministic value, implying that

$$\lim_{m\to\infty} \frac{\sqrt{\mathrm{Var}\left(\max\{\omega_i\}\right)}}{\mathbb{E}\left(\max\{\omega\}\right)} = 0. \tag{161}$$

It is important to note that this condition remains invariant under any scaling factor over the random variable ω_i, as both the standard deviation and the mean have identical dimensions. In particular, from Eq. (154), it follows:

$$\lim_{m\to\infty} \frac{\sqrt{\mathrm{Var}_\chi\left(\rho^{(1;m)}\right)}}{\left\langle \rho^{(1;m)} \right\rangle} = \frac{\cancel{\log(m)}\sqrt{\mathrm{Var}\left(\max\{\omega\}\right)}}{\cancel{\log(m)}\mathbb{E}\left(\max\{\omega\}\right)} = 0. \tag{162}$$

Combining Eqs. (70) and (74) with Eq. (162), we obtain

$$\lim_{m\to\infty} \gamma(\mathcal{A}, \psi(\chi)) = \frac{\left\langle \rho^{(1;m)} \right\rangle^2}{\mathrm{Var}_\chi\left(\rho^{(1;m)}\right)} = \infty. \tag{163}$$

□

In the proof of Thm. G.2, we focused on the statistical behavior of the maximum value in a growing set of random variables (r.v.s), particularly emphasizing positive event occurrences. This analysis remains applicable if we substitute the ReLU activation function with a linear one, as ReLU does not change the distribution within the positive domain of its support. This implies that the amplification of Initial Guessing Bias (IGB) in models with max pooling is independent of the choice of activation function, manifesting even without any activation function. This finding is not surprising, given that App. D.1 demonstrates that the key factor in the emergence of IGB is the statistics of nodes post-pooling layer, in particular $\left\langle \rho^{(1;m)} \right\rangle$. Hence, when considering models with pooling layers, it's crucial to evaluate the joint impact of the activation function and the pooling layer to accurately assess the presence and intensity of IGB.

G.3. Deep architectures

In App. E.1, we discussed how to derive $p_{f_0}(x)$ for a neural network with a single hidden layer. In this section, we will discuss the extension of the computation to a network with an arbitrary number of hidden layers. After that, we will focus on the impact that depth has on IGB. In particular, in App. G.3.3 we will show that network depth does not cause IGB, but it amplifies it when the network's depth increases.

G.3.1. MULTI-LAYER PERCEPTRON

The strategy that we will follow is similar to the one presented for the single-hidden-layer counterpart (see App. E.1). We will discuss here the case of MLP with ReLU activation function to focus on the effects of network depth. Extending the results discussed in the single-layer case to deeper networks, including variations such as the inclusion of Max-Pooling, is straightforward. We will propagate the signal across the network layers, keeping track of the changes in the distributions $p^{(\chi)}_{h_i^{(l+1)}}(x)$ and $p_{\left\langle h_i^{(l+1)} \right\rangle}(x)$. Using the CLT and its extensions considerably simplifies this propagation process because to keep track of changes in the distribution it is sufficient to see how few quantities vary.[12] Compared to the analysis in App. E.1, however, there is an important complicating element. While the elements in the set $\{h_i^{(1)}\}$ follow the same distribution (regardless of the choice of the activation function), from the second layer onward the activation function can induce a breaking in symmetry among the layer nodes, in the sense that they will follow, fixed a given configuration for the network weights, different distributions. This symmetry breaking, as we shall see, can cause an accentuation of IGB. To show this point, we will consider in our analysis the ReLU activation function.[13] Starting from the same setting described in App. E.1, we will derive a set of iterative equations to propagate across multiple layers.

- **layer-1**
  As shown in App. E.1 and in App. D.2, in the limit $d \to \infty$,

$$h_i^{(1)} \equiv \sum_j w_{ij}^{(0)} \xi_j \xrightarrow{\text{CLT}} p^{(\chi)}_{h_i^{(1)}}(x) \xrightarrow{d\to\infty} \mathcal{N}(0, \sigma_w^2) \xrightarrow{\text{ReLU}} p^{(\chi)}_{g_i^{(1)}}(x) \xrightarrow{d\to\infty} \Theta(x)\mathcal{N}\left(x; 0, \sigma_w^2\right) + \frac{1}{2}\delta(x)\,. \tag{164}$$

  All the nodes $\{h_i^{(1)}\}$ follow the same distribution, and so do the nodes $\{g_i^{(1)}\}$. In particular, this means that

$$\lim_{D\to\infty} \left\langle g_i^{(1)} \right\rangle = \left\langle g^{(1)} \right\rangle, \ \forall i\,. \tag{165}$$

- **layer-2**
  We start again from a combination of *r.v.*s

$$h_i^{(2)} \equiv \sum_j w_{ij}^{(1)} g_j^{(1)}\,. \tag{166}$$

  It is easy to prove that the generic *r.v.* $\left(w_{ij}^{(1)} g_j^{(1)}\right)$ involved in the sum satisfies the conditions of Eq. (29) and Eq. (30). So, in the limit $N_1 \to \infty$, we again have a convergence to a normal distribution. In particular,

$$p^{(\chi)}_{h_i^{(2)}}(x) \xrightarrow{N_1\to\infty} \mathcal{N}\left(x; \left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}, \text{Var}_\chi\left(g^{(1)}\right) S_{\boldsymbol{w}_i^2}\right) = \mathcal{N}\left(x; \left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}, \text{Var}_\chi\left(g^{(1)}\right) \overline{S_{\boldsymbol{w}_i^2}}\right)\,. \tag{167}$$

[12] This is because a Gaussian distribution is completely determined by the first two cumulants.

[13] We do not include the max pooling to isolate the effect of depth from that of max pooling. Note that absence of max pooling is equivalent of a max pooling with minimum kernel size.

In the last step of Eq. (167), as done for $p^{(\chi)}_{O^{(c)}}(x)$ in Sec. E.1, we used the concentration result derived in App. C.2.1. In particular that the distribution of the *r.v.* $S_{\boldsymbol{w}_i} \equiv \sum_{j=1}^{N_1} w_{ij}^{(1)}$ stays stable in the limit $N_1 \to \infty$,[14] while the distribution of $S_{\boldsymbol{w}_i^2} \equiv \sum_{j=1}^{N_1} \left(w_{ij}^{(1)}\right)^2$ asymptotically narrows around the mean value. Note that, as the distribution of $S_{\boldsymbol{w}_i}$ does not asymptotically concentrate, and since $\left\langle g^{(1)} \right\rangle \neq 0$, each node in the set $\{h_i^{(2)}\}$ will follow a different distribution. In particular we will have a set of normally distributed *r.v.*s, centred on different random points, $\{\left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}\}_i$. After passing through the ReLU activation function, we will have

$$p^{(\chi)}_{g_i^{(2)}}(x) \xrightarrow{N_1\to\infty} \Theta(x)\,\mathcal{N}\left(\left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}, \mathrm{Var}_\chi\left(g^{(1)}\right)\overline{S_{\boldsymbol{w}_i^2}}\right) + \left(\frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{\left\langle g^{(1)} \right\rangle S_{\boldsymbol{w}_i}}{\sqrt{2\mathrm{Var}_\chi\left(g^{(1)}\right)\overline{S_{\boldsymbol{w}_i^2}}}}\right)\right)\delta(x)\,. \tag{168}$$

- **layer-3**
  We can repeat the approach of the previous layer for the new set of variables,

$$h_i^{(3)} \equiv \sum_j w_{ij}^{(2)} g_j^{(2)}\,, \tag{169}$$

getting

$$p^{(\chi)}_{h_i^{(3)}}(x) \xrightarrow{N_2\to\infty} \mathcal{N}\left(x; \sum_{j=1}^{N_2} w_{ij}^{(2)}\left\langle g_j^{(2)} \right\rangle, \sum_{j=1}^{N_2}\left(w_{ij}^{(2)}\right)^2 \mathrm{Var}_\chi\left(g_j^{(2)}\right)\right)\,. \tag{170}$$

Note that in this case we cannot take out from the sums $\left\langle g_j^{(2)} \right\rangle$ and $\mathrm{Var}_\chi\left(g_j^{(2)}\right)$ since the nodes $\{g_j^{(2)}\}$ are not identically distributed. In App. C.2 we analyze the differences in $p^{(\chi)}_{h_i^{(3)}}(x)$ between the different nodes $\{h_i^{(3)}\}$. In particular, we show that

$$p^{(\chi)}_{h_i^{(3)}}(x) \xrightarrow{N_2\to\infty} \mathcal{N}\left(x; \left\langle h_i^{(3)} \right\rangle, \mathrm{Var}_\chi\left(h_i^{(3)}\right)\right)\,, \tag{171}$$

$$p_{\left\langle h_i^{(3)} \right\rangle}(x) \xrightarrow{N_2\to\infty} \mathcal{N}\left(x; 0, \sigma_w^2\overline{\left\langle g_j^{(2)} \right\rangle^2}\right)\,, \tag{172}$$

$$p_{\mathrm{Var}_\chi\left(h_i^{(3)}\right)}(x) \xrightarrow{N_2\to\infty} \delta\left(x - \sigma_w^2\overline{\mathrm{Var}_\chi\left(g_j^{(2)}\right)}\right)\,. \tag{173}$$

- **layer-l**
  The steps described for the propagation of the *layer-3* provide us with an iteration scheme that we can follow for a generic layer $l \geq 3$. In fact, knowing the statistics of the previous layers, we can compute

$$\overline{\mathrm{Var}_\chi\left(g_j^{(l-1)}\right)} = \int_{\mathbb{R}} \mathrm{Var}_\chi\left(g_j^{(l-1)}\right)(x)\;\; \mathcal{N}\left(x; 0, \sigma_w^2\overline{\left\langle g_j^{(l-2)} \right\rangle^2}\right) dx\,, \tag{174}$$

$$\overline{\left\langle g_j^{(l-1)} \right\rangle^2} = \int_{\mathbb{R}} \left\langle g_j^{(l-1)} \right\rangle^2(x)\;\; \mathcal{N}\left(x; 0, \sigma_w^2\overline{\left\langle g_j^{(l-2)} \right\rangle^2}\right) dx. \tag{175}$$

From Eq. (174) and Eq. (175) we can compute then

$$p^{(\chi)}_{h_i^{(l)}}(x) \xrightarrow{N_{l-1}\to\infty} \mathcal{N}\left(x; \left\langle h_i^{(l)} \right\rangle, \sigma_w^2\overline{\mathrm{Var}_\chi\left(g_j^{(l-1)}\right)}\right)\,, \tag{176}$$

$$p_{\left\langle h_i^{(l)} \right\rangle}(x) \xrightarrow{N_{l-1}\to\infty} \mathcal{N}\left(x; 0, \sigma_w^2\overline{\left\langle g_j^{(l-1)} \right\rangle^2}\right)\,. \tag{177}$$

[14] Fluctuations and mean stay both $\mathcal{O}(1)$.

Finally we can use (177) to iterate the set of (174) and (175) for the layer l, *i.e.*

$$\overline{\mathrm{Var}_{\chi}\left(g_j^{(l)}\right)} = \int_{\mathbb{R}} \mathrm{Var}_{\chi}\left(g_j^{(l)}\right)(x)\ \ p_{\left\langle h_j^{(l)}\right\rangle}(x)\,dx\,, \tag{178}$$

$$\overline{\left\langle g_j^{(l)}\right\rangle^2} = \int_{\mathbb{R}} \left\langle g_j^{(l)}\right\rangle^2(x)\ \ p_{\left\langle h_j^{(l)}\right\rangle}(x)\,dx\,. \tag{179}$$

- **layer-$(L+1)$**
  Arriving at the output layer, following the same iterative scheme we will have

$$\begin{cases} p^{(\chi)}_{O^{(c)}}(x) \xrightarrow{N_L\to\infty} \mathcal{N}\left(x; \left\langle O^{(c)}\right\rangle, \sigma_w^2 \overline{\mathrm{Var}_{\chi}\left(g_j^{(L)}\right)}\right) \\ p^{(\chi)}_{\mu_c}(x) \xrightarrow{N_L\to\infty} \mathcal{N}\left(x; 0, \sigma_w^2 \overline{\left\langle g_j^{(L)}\right\rangle^2}\right) \end{cases}. \tag{180}$$

These two distributions are the only ingredient that we need to replicate the steps described at the end of App. E.1 to get $\mathbb{P}^{\mathcal{W}}(f_0)$.

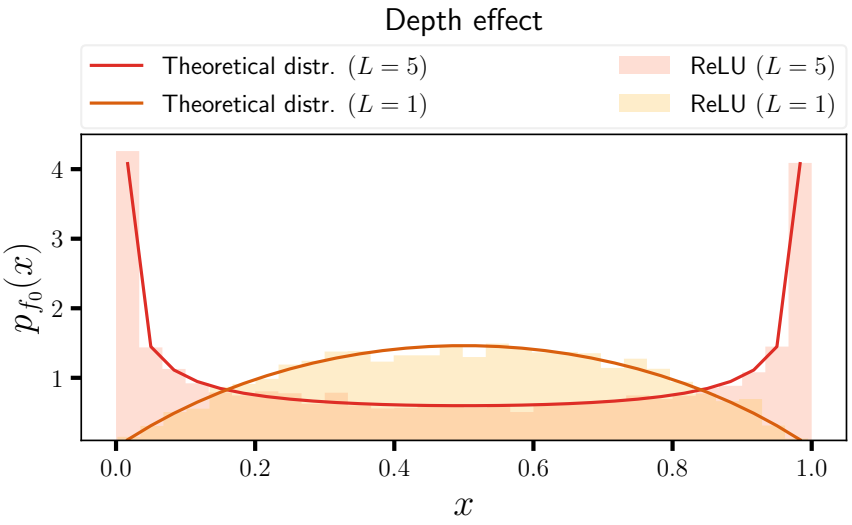

Figure G.1. Effects induced by the depth of the network. Simulations on an MLP with L hidden layers and ReLU activation function (without any pooling layers) are compared for two different depth values. The theoretical curves are superimposed on the empirical distributions, demonstrating a good agreement between the predictions of our analysis and empirical observations. For a single hidden layer network $(L=1)$ we have a stable distribution (*i.e.* that does not narrow around $\overline{f_0}=0.5$ for $D\to\infty$) but still peaked on $\overline{f_0}=0.5$. Increasing the number of hidden layers the probability mass moves from the center to the extremes of the support. For these simulations we used the `GB` dataset on model `MHLP` (see App. I.4 for more details).

G.3.2. Inclusion of Non-Null Biases

In the analysis presented in App. G.3.1, we considered a setting with null biases for the sake of clarity. However, the analysis can be extended to initializations where the biases are set to non-null values. For instance, considering an MLP with Gaussian-initialized bias parameters, the pre-activations of a generic hidden layer can be described as:

$$h_i' = h_i + b_i\,. \tag{181}$$

Here, h_i represents node i's pre-activation in the absence of biases, which, as per our analysis, follows a normal distribution.[15] The term b_i denotes the bias parameter. Given that h'_i is a sum of two independent Gaussian random variables, it too will exhibit a normal distribution. This allows for a seamless extension of our analysis. Similarly, if the bias parameters are initialized with a constant value ($b_i = c$), the analysis can be adapted accordingly. Adding a constant to h_i merely shifts the distribution's center, preserving the Gaussian profile for h'_i and reaffirming the robustness of our findings regarding IGB across different initialization strategies.

G.3.3. Amplification of IGB with depth

Here we prove that the distribution of $\mathbb{P}^{\mathcal{W}}(f_0)$ converges to a delta-peaked distribution in a multi-layer perceptron with ReLU activation when the number of layers goes to infinity. More precisely:

Theorem G.3. *Consider a dataset with Gaussian-distributed, i.i.d. components, where $\xi_b^{(a)} \sim \mathcal{N}(0,1)$, and $\xi_b^{(a)}$ represents the b-th component of the a-th input vector. Inputs are processed through an MLP, mapping the input to output according to Eq. (21), Eq. (22), and Eq. (23). The weights are initialized following the Kaiming normal scheme, i.e., $w_{ij}^{(l)} \sim \mathcal{N}\left(0, \frac{\sigma_w^2}{N_{l-1}}\right)$, with zero bias weights. Let us consider a multi-layer perceptron with L hidden layers, ReLU activation function, and without pooling layers. Let us assume that, in the $\lim_{N_l\to\infty} \forall l \in [0,\dots,L]$,* w.h.p.*:*

$$Var_\chi\left(h^{(l)}\right) > 0,\ \forall l\,, \tag{182}$$

and in particular

$$\lim_{l\to\infty} Var_\chi\left(h^{(l)}\right) > 0, \tag{183}$$

where $Var_\chi\left(h^{(l)}\right)$ indicate the variance of $\mathbb{P}^\chi\left(h^{(l)} \mid \mathcal{W}\right)$. Then

$$\lim_{L\to\infty} \gamma(\mathcal{A}, \psi(\chi)) = \infty\,. \tag{184}$$

To prove Eq. (184), we will show that

$$\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l+1)}\right\rangle\right) > (K+\epsilon)\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l)}\right\rangle\right)\,, \tag{185}$$

$$\mathrm{Var}_{\chi}\left(h^{(l+1)}\right) < (K-\epsilon)\mathrm{Var}_{\chi}\left(h^{(l)}\right)\,, \tag{186}$$

where K is a constant. Once proved the above relations we can, indeed, write

$$\frac{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l+2)}\right\rangle\right)}{\mathrm{Var}_{\chi}\left(h^{(l+2)}\right)} > \left(\frac{1}{K}\right)^l \frac{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(2)}\right\rangle\right)}{\mathrm{Var}_{\chi}\left(h^{(2)}\right)}(K+\epsilon)^l > l\underbrace{\left(\frac{1}{K}\frac{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(2)}\right\rangle\right)}{\mathrm{Var}_{\chi}\left(h^{(2)}\right)}\epsilon\right)}_{\equiv\frac{1}{C}},$$

$$\implies \frac{\mathrm{Var}_{\chi}\left(h^{(l+2)}\right)}{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l+2)}\right\rangle\right)} < C\frac{1}{l}. \tag{187}$$

Therefore if we consider an infinite depth network, *i.e.* $L\to\infty$, we have:

$$\frac{\mathrm{Var}_{\chi}\left(h^{(L)}\right)}{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(L)}\right\rangle\right)} = \mathcal{O}\left(\frac{1}{L}\right) \implies \frac{\mathrm{Var}_{\mathcal{W}}\left(\mu_c\right)}{\mathrm{Var}_{\chi}\left(O^{(c)}\right)} = \mathcal{O}\left(L\right) \implies \lim_{L\to\infty}\frac{\mathrm{Var}_{\mathcal{W}}\left(\mu_c\right)}{\mathrm{Var}_{\chi}\left(O^{(c)}\right)} = \infty\,. \tag{188}$$

Proof. Let us start from Eq. (185). As a first step we prove that, for $N_{l-1}\to\infty$,

$$\frac{1}{2}\mathrm{Var}_{\chi}\left(h^{(l)}\right) > \overline{\mathrm{Var}_{\chi}\left(g^{(l)}\right)} \equiv \overline{\left\langle\left(g^{(l)}\right)^2\right\rangle} - \overline{\left\langle g^{(l)}\right\rangle^2}. \tag{189}$$

[15] We omitted the layer index l on the variables h_i, b_i, h'_i to lighten the notation.

To prove this inequality we will now focus separately on the two terms of the *r.h.s.*. First we analyze $\overline{\left\langle g^{(l)} \right\rangle^2}$. From Eq. (176)

$$\text{Var}_{\mathcal{W}}\left(\left\langle h^{(l+1)} \right\rangle\right) = \sigma_w^2 \overline{\left\langle g_j^{(l)} \right\rangle^2}. \tag{190}$$

$$\overline{\left\langle h_j^{(l)} \right\rangle^2} = \int_{-\infty}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, x^2 dx = \int_{-\infty}^{0} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, x^2 dx + \int_{0}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, x^2 dx\,. \tag{191}$$

We now use the symmetry of $p_{\left\langle h_j^{(l)} \right\rangle}(x)$ (which is a Gaussian centered in 0), *i.e.*

$$p_{\left\langle h_j^{(l)} \right\rangle}(x) = p_{\left\langle h_j^{(l)} \right\rangle}(-x)\ , \tag{192}$$

to rewrite Eq. (191) as

$$\frac{1}{2}\overline{\left\langle h_j^{(l)} \right\rangle^2} = \int_{0}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, x^2 dx \equiv \int_{0}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, \left(\mu_+(x) + \mu_-(x)\right)^2 dx\,, \tag{193}$$

where we defined

$$\begin{aligned} \mu_+(\mu) &= \int_{0}^{\infty} \mathcal{N}\left(x; \mu, \text{Var}_\chi\left(h^{(l+1)}\right)\right) x dx > 0\,, \\ \mu_-(\mu) &= \int_{-\infty}^{0} \mathcal{N}\left(x; \mu, \text{Var}_\chi\left(h^{(l+1)}\right)\right) x dx < 0\,. \end{aligned} \tag{194}$$

We can thus rewrite

$$\frac{1}{2}\overline{\left\langle h_j^{(l)} \right\rangle^2} = \int_{0}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, \left(\mu_+^2 + \mu_-^2 + 2\underbrace{\mu_+\mu_-}_{\leq 0}\right) dx. \tag{195}$$

Note that $\mu_+\mu_- = 0$ implies either $\mu_+ = 0$ or $\mu_- = 0$ *i.e.* a distribution $p^{(\chi)}_{h_j^{(l)}}(x)$ with non-positive or non-negative support. Since $p^{(\chi)}_{h_j^{(l)}}(x)$ is a Gaussian distribution this may only happen in the limit of its variance going to 0 *i.e.* if the Gaussian shrink into a Dirac delta distribution. By hypothesis we excluded this possibility (Eq. (182) and Eq. (183)); therefore $\mu_+\mu_- < 0$ and we can rewrite

$$\frac{1}{2}\overline{\left\langle h_j^{(l)} \right\rangle^2} < \underbrace{\int_{0}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, \left(\mu_+(x)^2 + \mu_-(x)^2\right) dx}_{\equiv I}\ . \tag{196}$$

Now let us focus on the integral I of Eq. (196). We note that

$$-\mu_-(x) = -\int_{-\infty}^{0} \mathcal{N}\left(\tilde{x}; x, \text{Var}_\chi\left(h^{(l+1)}\right)\right) \tilde{x} d\tilde{x} == \int_{0}^{\infty} \mathcal{N}\left(y; -x, \text{Var}_\chi\left(h^{(l+1)}\right)\right) y dy = \mu_+(-x)\,, \tag{197}$$

where in the second step we just changed the integration variable $\tilde{x} \to y \equiv -\tilde{x}$. We can therefore rewrite

$$I = \int_{0}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, \left(\mu_+(x)^2 + \mu_+(-x)^2\right) dx = \int_{-\infty}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x)\, \mu_+(x)^2 dx = \overline{\left\langle g_j^{(l)} \right\rangle^2}, \tag{198}$$

where for the second step we used again Eq. (192). In summary, we showed that

$$\overline{\left\langle g_j^{(l)} \right\rangle^2} > \left(\frac{1}{2}\right) \overline{\left\langle h_j^{(l)} \right\rangle^2}. \tag{199}$$

We now turn our attention to the second term appearing in the *r.h.s.* of Eq. (189), *i.e.* $\overline{\left\langle \left(g^{(l)}\right)^2 \right\rangle}$:

$$\begin{aligned}\overline{\left\langle \left(g^{(l)}\right)^2 \right\rangle} &= \int_{-\infty}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x) \left\langle \left(g^{(l)}\right)^2 \right\rangle (x)\, dx = \int_0^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x) \left(\left\langle \left(g^{(l)}\right)^2 \right\rangle (x) + \left\langle \left(g^{(l)}\right)^2 \right\rangle (-x) \right) dx = \\ &= \int_0^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x) \left(\left\langle \left(g^{(l)}\right)^2 \right\rangle_+ (x) + \left\langle \left(g^{(l)}\right)^2 \right\rangle_- (x) \right) dx,\end{aligned} \tag{200}$$

where, analogously to Eq. (194), we defined

$$\begin{aligned}\left\langle \left(g^{(l)}\right)^2 \right\rangle_+ (x) &= \int_0^{\infty} \mathcal{N}\left(y; x, \mathrm{Var}_\chi\left(h^{(l+1)}\right)\right) y^2 dy\,, \\ \left\langle \left(g^{(l)}\right)^2 \right\rangle_- (x) &= \int_{-\infty}^{0} \mathcal{N}\left(y; x, \mathrm{Var}_\chi\left(h^{(l+1)}\right)\right) y^2 dy.\end{aligned} \tag{201}$$

From the above definitions, we see that

$$\left\langle \left(g^{(l)}\right)^2 \right\rangle_+ (x) + \left\langle \left(g^{(l)}\right)^2 \right\rangle_- (x) = \left\langle \left(h^{(l)}\right)^2 \right\rangle (x); \tag{202}$$

therefore substituting into Eq. (200) we get

$$\overline{\left\langle \left(g_j^{(l)}\right)^2 \right\rangle} = \int_0^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x) \left\langle \left(h_j^{(l)}\right)^2 \right\rangle (x)\, dx = \frac{1}{2} \int_{-\infty}^{\infty} p_{\left\langle h_j^{(l)} \right\rangle}(x) \left\langle \left(h_j^{(l)}\right)^2 \right\rangle (x)\, dx = \frac{1}{2} \overline{\left\langle \left(h_j^{(l)}\right)^2 \right\rangle}. \tag{203}$$

Finally, let us consider

$$\begin{aligned}\overline{\mathrm{Var}_\chi\left(g^{(l)}\right)} &\equiv \overline{\left\langle \left(g^{(l)}\right)^2 \right\rangle} - \overline{\left\langle g^{(l)} \right\rangle^2} = \frac{1}{2} \overline{\left\langle \left(h^{(l)}\right)^2 \right\rangle} - \overline{\left\langle g^{(l)} \right\rangle^2} < \\ &< \frac{1}{2} \overline{\left\langle \left(h^{(l)}\right)^2 \right\rangle} - \frac{1}{2} \overline{\left\langle h^{(l)} \right\rangle^2} = \frac{1}{2} \overline{\mathrm{Var}_\chi\left(h^{(l)}\right)} \xrightarrow{N_{l-1} \to \infty} \frac{1}{2} \mathrm{Var}_\chi\left(h^{(l)}\right).\end{aligned} \tag{204}$$

where in the last step we used the concentration result discussed in App. C.2.1, *i.e.* that the distribution of $\mathrm{Var}_\chi\left(h^{(l)}\right)$ asymptotically narrows around its mean value, becoming, *w.h.p.* independent of the realization $\mathcal{W}$.
Now we have all the ingredient we need; in fact, from (176) we know that

$$\mathrm{Var}_\chi\left(h^{(l)}\right) = \sigma_w^2 \overline{\mathrm{Var}_\chi\left(g_j^{(l-1)}\right)}. \tag{205}$$

Therefore, from (204), we can conclude

$$\mathrm{Var}_\chi\left(h^{(l+1)}\right) < \left(\frac{\sigma_w^2}{2} - \epsilon_l'\right) \mathrm{Var}_\chi\left(h^{(l)}\right), \tag{206}$$

with $\epsilon_l' > 0$. Similarly, combining (177) with (199) we get

$$\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l+1)} \right\rangle\right) > \left(\frac{\sigma_w^2}{2} + \epsilon_l\right) \mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l)} \right\rangle\right), \tag{207}$$

with $\epsilon_l > 0$. Calling

$$\epsilon \equiv \inf_l \epsilon_l\,, \tag{208}$$

we can finally write

$$\begin{aligned}&\frac{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l+2)} \right\rangle\right)}{\mathrm{Var}_\chi\left(h^{(l+2)}\right)} > \left(\frac{2}{\sigma_w^2}\right)^l \frac{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(2)} \right\rangle\right)}{\mathrm{Var}_\chi\left(h^{(2)}\right)} \left(\frac{\sigma_w^2}{2} + \epsilon\right)^l > l \underbrace{\left(\frac{2}{\sigma_w^2} \frac{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(2)} \right\rangle\right)}{\mathrm{Var}_\chi\left(h^{(2)}\right)} \epsilon\right)}_{\equiv \frac{1}{C}} \\ &\implies \frac{\mathrm{Var}_\chi\left(h^{(l+2)}\right)}{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(l+2)} \right\rangle\right)} < C \frac{1}{l}.\end{aligned} \tag{209}$$

Therefore if we consider an infinite depth network, *i.e.* $L \to \infty$, we have:

$$\frac{\mathrm{Var}_{\chi}\left(h^{(L)}\right)}{\mathrm{Var}_{\mathcal{W}}\left(\left\langle h^{(L)}\right\rangle\right)} = \mathcal{O}\left(\frac{1}{L}\right) \Longrightarrow \frac{\mathrm{Var}_{\mathcal{W}}\left(\mu_c\right)}{\mathrm{Var}_{\chi}\left(O^{(c)}\right)} = \mathcal{O}\left(L\right) \Longrightarrow \lim_{L\to\infty} \frac{\mathrm{Var}_{\mathcal{W}}\left(\mu_c\right)}{\mathrm{Var}_{\chi}\left(O^{(c)}\right)} = \infty\,. \tag{210}$$

□

We observe a fundamental distinction compared to the previous cases. Network depth can amplify the level of IGB in systems where it is already present. However, depth alone does not induce IGB. Considering a model with a linear activation function and no max pooling after the first hidden layer, and using Gaussian data centered at zero, we have $\left\langle \rho^{(1;m)} \right\rangle = \left\langle h_m^{(1)} \right\rangle = 0$. Thus, there is no Node Symmetry Breaking (NSB) in the second layer; in other words, the nodes in the second layer are Gaussian *r.v.*s centered at zero. It can be easily shown that, by iterating this process through each layer, the same scenario will be repeated in each subsequent layer up to the output layer. Therefore, NSB does not occur even in the output nodes, resulting in the absence of IGB in the system.

H. Extension to multi-class problems

We want now to extend the analysis to the multi-class case, *i.e.* to problems with $N_C > 2$. Following the framework we introduced earlier, we can easily extend the computation. We will again consider the distribution among the N_C classes of the dataset elements after initialization. In particular, we can define N_C values associated with the fraction of dataset elements classified as belonging to each of the N_C classes, $\{f_i\}_{i=0}^{N_C-1}$. We could also consider the distribution of the sorted frequencies; in other words, in each experiment, we order the classes according to the corresponding frequencies and not by their label. To distinguish these frequencies (and their statistics) from the set $\{f_i\}_{i=0}^{N_C-1}$, we use indices $\tilde{i}$, and call this new set $\left\{f_{\tilde{i}}\right\}_{\tilde{i}=0}^{N_C-1}$. In particular $f_{\tilde{0}}$ indicates the biggest frequency among the N_C, $f_{\tilde{1}}$ the second one and so on. Finally we add the set (or subset) cardinality, M, on the output nodes set $\left\{O_M^{(c)}\right\}_{c=0}^{M-1}$.[16] Analogously to the set of fractions, we define the set of ranked output nodes $\left\{O_M^{(\tilde{c})}\right\}_{\tilde{c}=0}^{M-1}$ In the following, similarly to Sec. E.1 we will derive the distribution of f_0.

We will make in the derivation of f_0 the following approximation: *We will repeat the same approach of the binary setting comparing the generic class "*0*" with the class with bigger mean output value among the* $N_C - 1$ *remaining classes.*

The first ingredient that we need is therefore the statistics of $O_{N_C-1}^{(\tilde{0})}$. From the analysis of previous sections, we know that $p^{(\chi)}_{O_{N_C-1}^{(\tilde{0})}}(x)$ is asymptotically Gaussian. The mean value of this Gaussian variable will follow the distribution of the maximum among $N_C - 1$ drawn from the Gaussian distribution $p_{\mu_c}(x)$ as, by definition,

$$\left\langle O_{N_C-1}^{(\tilde{0})} \right\rangle \equiv \max_{c\in\{1,\ldots,N_C\}} \left\langle O^{(c)} \right\rangle\,. \tag{211}$$

Maximum over $N_C - 1$ Gaussian draws We will follow the same approach used in App. D.2 . We will again start from Eq. (85). In this case,

$$p_X\left(x\right) \equiv p_{\mu_c}\left(x\right) = \mathcal{N}\left(x; 0, \hat{\sigma^2}\right)\,, \tag{212}$$

where we used a generic $\hat{\sigma^2}$ to indicate the variance. We will perform the computation in this general settings; then to analyze the specific cases we can just substitute the right variance for the specific case (see App. D.2).

$$F_X(y) \equiv \int_{-\infty}^{y} p_X\left(x\right) dx = \int_{-\infty}^{y} \mathcal{N}\left(x; 0, \hat{\sigma^2}\right) dx = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{y}{\sqrt{2}\hat{\sigma}}\right)\right)\,. \tag{213}$$

Now, putting all pieces together in Eq. (85), we get our target distribution,

$$p_Y\left(y\right) \equiv p_{\left\langle O_{N_C-1}^{(\tilde{0})} \right\rangle}\left(y\right) = \left(N_C - 1\right)\mathcal{N}\left(y; 0, \tilde{\sigma}^2\right)\left(\frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{y}{\sqrt{2}\tilde{\sigma}}\right)\right)\right)^{N_C-2}\,. \tag{214}$$

[16] Here, the cardinality of the set indicates the number of classes over which the statistics are computed. The statistics of the *r.v.* change with the number of classes (or nodes), i.e., with the cardinality.

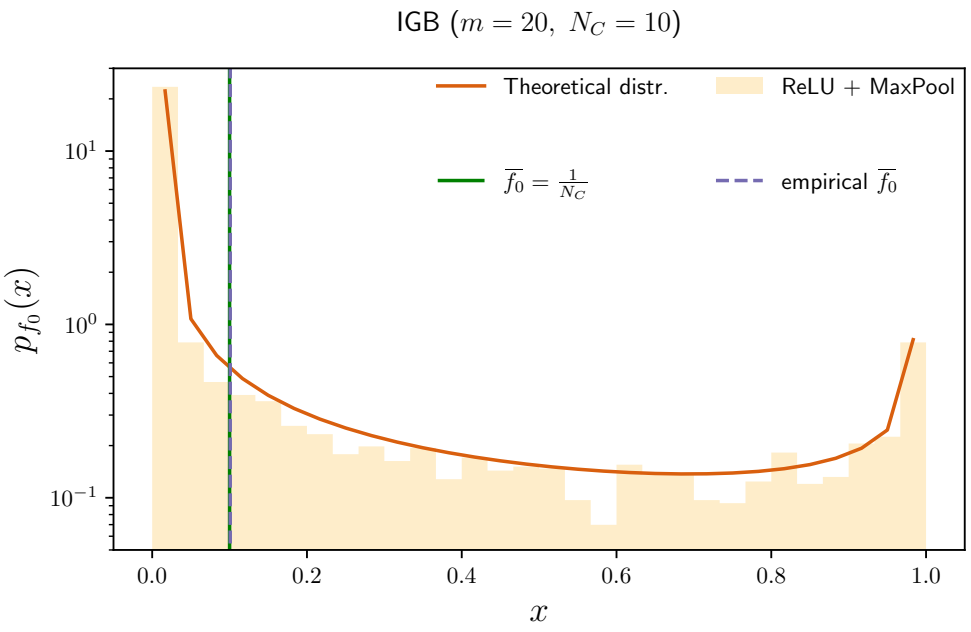

Figure H.1. The distribution $p_{f_0}(x)$ in a multi-class problem with $N_C = 10$. The empirical mean value is reported; note that despite the difference in the distribution the mean value is the same that we would observe using a linear activation function $\overline{f_0} = \frac{1}{N_C}$. For these simulations we used the dataset `GB` on the model `SHLP` with max pooling ($m = 20$). (see App. I.4 for more details).

Now we can proceed following again the steps presented at the end of App. E.1. The only difference we will have is in Eq. (109); now we will have

$$p_{f_0}(x) = \int_{-\infty}^{\infty} p_{\left\langle O^{(0)}_{N_C} \right\rangle}(\tilde{x})\, p_{\left\langle O^{(\tilde{0})}_{N_C-1} \right\rangle}\left(\tilde{x} - \hat{\Delta}(x)\right) d\tilde{x}\,, \tag{215}$$

where $\left\langle O^{(\tilde{0})}_{N_C-1} \right\rangle$ the biggest mean output among the $N_C - 1$. Note that $\hat{\Delta}(f_0)$ is a function of f_0. Substituting Eq. (212) and Eq. (214) we have:

$$p_{f_0}(x) = \int_{-\infty}^{\infty} \mathcal{N}\left(\tilde{x}; 0, \hat{\sigma^2}\right) (N_C - 1)\mathcal{N}\left(\tilde{x}; \hat{\Delta}(x), \hat{\sigma^2}\right) \left(\frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\tilde{x} - \hat{\Delta}(x)}{\sqrt{2}\hat{\sigma}}\right)\right)\right)^{N_C-2} d\tilde{x}\,. \tag{216}$$

H.1. Increasing number of classes exacerbates IGB

Having a large number of classes increases the probability that the center of the distribution related to one of these classes is an outlier of the distribution $p_{\mu_c}(x)$. However, since $p_{\mu_c}(x)$ is Gaussian (a fast-decaying distribution), the typical value of the maximum increases slowly. Therefore, an extremely large number of classes, N_c, will be needed to observe significant changes.

We can analyze this quantitatively. The typical value of the maximum out of N_c Gaussian random variables of variance ϵ grows as $\sqrt{(2\epsilon \log(N_c))}$ (Gumbel, 1958; Hartarsky et al., 2019). In the absence of IGB, ϵ is the variance of $p_{\mu_c}(x)$ and scales inversely with the dataset size, i.e., $\sim 1/D$ (see e.g., App. D.2 and Fig. D.1). This implies that outliers will only appear if the number of classes far exceeds the number of datapoints, a situation that does not occur in real life.

In summary, if there is no IGB, a large enough dataset ensures that increasing the number of classes does not affect the presence of IGB.

Conversely, if IGB is present, a sufficiently large number of classes may result in high imbalance even with a relatively small value of $\gamma(\mathcal{A}, \psi(\chi))$.

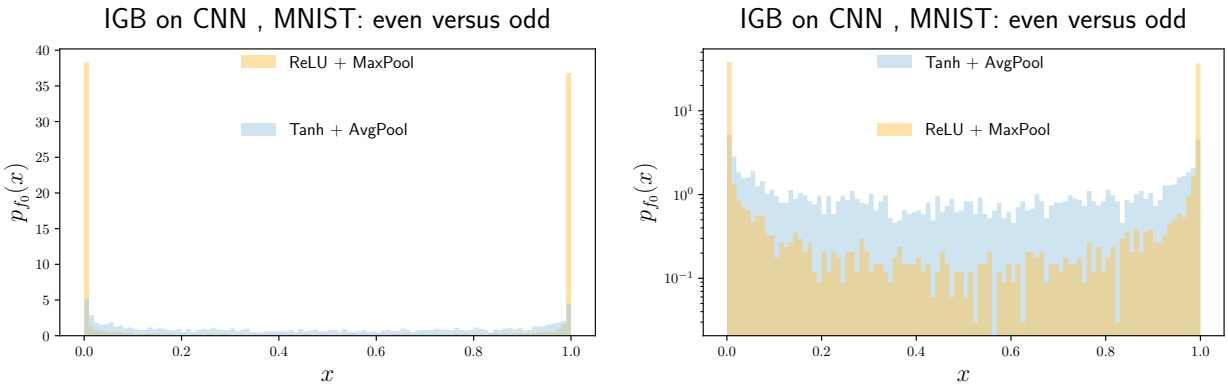

Figure I.1. Comparison of the distribution $p_{f_0}(x)$ between two untrained neural networks fed with MNIST dataset; the ten classes of the dataset were merged in two macro groups: even and odd. The two architectures employed in the comparison differ in the choice of activation function and pooling. For these simulations we used the `E&O` dataset on the model `CNN-B`. (see App. I.4 for more details). On the right, the same plot in a logarithmic scale to better visualize the differences in the low-density regions.

I. Experiments

The analysis presented is able to identify IGB and analytically describe the phenomena in a variety of settings (*e.g.*, MLPs with different activation functions, arbitrary depth, and the presence or absence of pooling layers). However, IGB is significantly broader in scope, as it may be observed in a wide range of dataset and architectural combinations. Although our theoretical analysis does not go into these scenarios (a further extension of the analytical conclusions to more sophisticated designs will be the focus of future research), we provide some representative instances to demonstrate the breadth of scenarios where IGB is significant in real-world circumstances. The experiments presented, while not exhaustive, aim to provide insights and emphasize three fundamental points that will be developed in a separate publication:

- IGB is amplified in structured data, particularly in cases of strongly correlated data (App. I.1).
- IGB is a ubiquitous phenomenon, that happens with most combinations of datasets and architectures (App. I.1, I.2).
- The differences induced by IGB on the initial state of the network have an impact on its dynamics, rendering it qualitatively distinct, especially in the initial phase, compared to the case without IGB (App. I.2).

I.1. Experiments on real data

In this section, we show some empirical evidence of IGB on combinations of models and data which are not covered by our theory, such as CNN architectures and the MNIST, CIFAR10 and CIFAR100 datasets.[17] In App. I.2, we will also include experiments on additional architectures (ResNet, MLP-mixer and Vision Transformers). The experiments we will discuss do not presume to show in a complete and exhaustive way the range of models and cases in which IGB can occur; this question is outside the scope of our work. Rather, they are some didactic examples intended to show how, beyond the assumptions used in our analysis to quantitatively treat the phenomenon, IGB also emerges in a broader context. We first present and discuss the experiments. For technical details for the reproducibility we refer the reader to App. I.4.

I.1.1. A PROTOTYPE FOR HIGHLY CORRELATED DATA: MNIST

In the analysis we presented in the main part of the paper, we focused on a simple data structure, to underline effects induced by architecture design: a remarkable conclusion from this setting is the following:
In a classification problem, an untrained network with biases set to zero and fed with i.i.d. data-points (i.e. different classes are identically distributed) often starts the training process with a strong bias toward one of the classes.
For our theoretical analysis, we placed ourselves in a setting where the effect of IGB is minimal and unaffected by the data

[17] MNIST is a particularly insightful dataset for IGB, because the correlations between pixels are high. This is opposite to what we cover with our theory, which is based on i.i.d. inputs.

structure. This ensures that the sources of IGB, which our analysis links to architectural design elements, do not stem from dataset characteristics. Below, we present experiments in which these hypotheses on data are violated, to demonstrate how the phenomenon manifests itself (more prominently) in real datasets, where:

- Data-points belonging to different classes will not be identically distributed.
- Components of a single data-point will not be independent; for example pixels of an image will clearly be correlated.
- Similarly, correlations between different data-points (belonging to the same class) is possible.

MNIST constitutes a good candidate to represent a scenario of strong correlations in the data. In fact, in this dataset the images are characterized by a small percentage of nonzero pixels; this leads to the formation of areas, within the same image, characterized by similar values and therefore strongly correlated. In addition, image areas are strongly correlated between different elements. For example, we are unlikely to observe the writing of a number on the edges of an image; therefore, different images will have similar values of pixels along the edges. When considering images of the same class, this correlation is even more pronounced. From the MNIST dataset, we defined two macro-classes, even and odd, by merging the 10 starting classes according to their parity. We then propagated the binary dataset, obtained in this way, through two untrained convolutional neural networks. The difference between the two networks lies in the choice of activation and pooling function employed. In particular the dataset is propagated through `CNN-B` (network details in App. I.4). The `CNN-B` model is a CNN where the last convolutional layer is directly connected to the outputs. We choose it this way, because, as earlier shown, IGB cannot arise without hidden layers. This indicates that the observation of IGB is caused by the convolutional layers or, to put it more precisely, it emerges from the propagation of structured data through the convolutional layers.

We show this in Fig. I.1, which contains two important messages:

♢ Unlike the cases presented in our theoretical analysis, neither of the two choices has total absence of IGB. This is consistent with the intuition that correlations in the data cause or amplify IGB.

♢ While we observe presence of IGB in both cases, this is more pronounced for the architecture with ReLU and max pooling, consistent with what we saw in our study.

These observations show how IGB is a rather universal phenomenon, related to the combined effect of data structure and the design of the neural network.

I.1.2. A FURTHER EXAMPLE: CIFAR10

We now show a multi-class example, on the same network as in App. I.1.1 (`CNN-B`) we propagated CIFAR10. The results are reported in Fig. I.2. As with MNIST, the results are qualitatively consistent with the conclusions of our theory: the network with ReLU and max pooling displays a stronger IGB than its counterpart with $\tanh$ and AvgPooling. Note, also, how in both cases, symmetry between classes is preserved at the ensemble level. In both cases, in fact, we get $\overline{f_c} = 1/N_C$. To show this we show in Fig. I.2 two vertical lines at the mean values calculated on the empirical distributions (histograms), each surrounded by its own uncertainty, estimated through the standard error.

I.1.3. HIGH CARDINALITY DATASET: CIFAR100

We also provide another example of a multi-class dataset. In this case, we choose Cifar-100 as a prototype of a high-cardinality dataset (with a high number of classes) to demonstrate the presence of IGB in this scenario as well. In the absence of IGB, the distribution should be peaked at $f_0 = 1/N_C = 1/100$. However, the histogram shows a peak at 0 (most of the time, the generic class does not receive any assigned elements). The gap between the peak at 0 and the rest of the distribution, particularly evident in the log-log scale plot, indicates a resolution limit due to the finite number of dataset elements ($D = 10^4$).

I.1.4. ADDITIONAL ARCHITECTURES

We also provide empirical evidence of IGB in different architectures. To improve the legibility of the paper, we show results for `ResNet`, `ViT` and `MLP-mix` in Sec. I.2. While App. I.2 is devoted to the dynamics und IGB, the reader can recognize

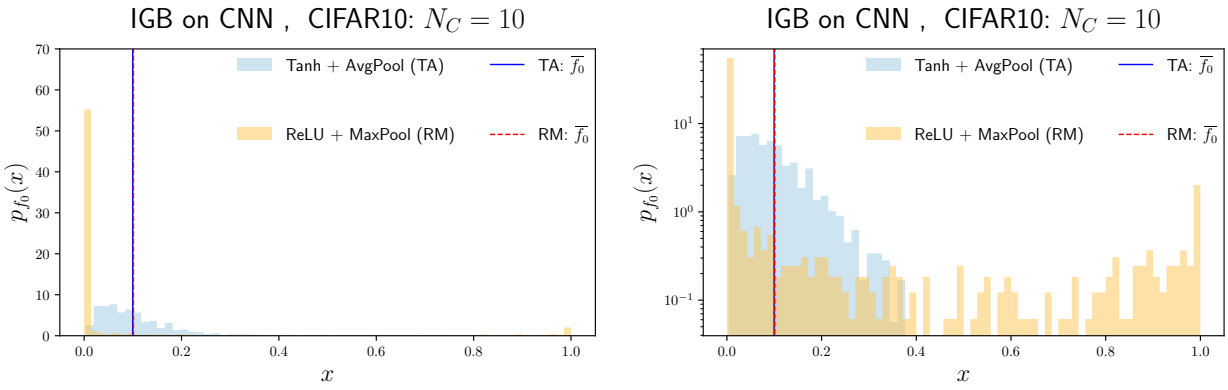

Figure I.2. Comparison of the distribution $p_{f_0}(x)$ between two untrained neural networks fed with CIFAR10 dataset (with 10 classes). The two architectures employed in the comparison differ in the choice of activation function and pooling. In the absence of IGB, the distributions would concentrate around the vertical line. For these simulations we used the dataset `C10` on the model `CNN-B` described below (see App. I.4). On the right, the same plot in a logarithmic scale to better visualize the differences in the low-density regions.

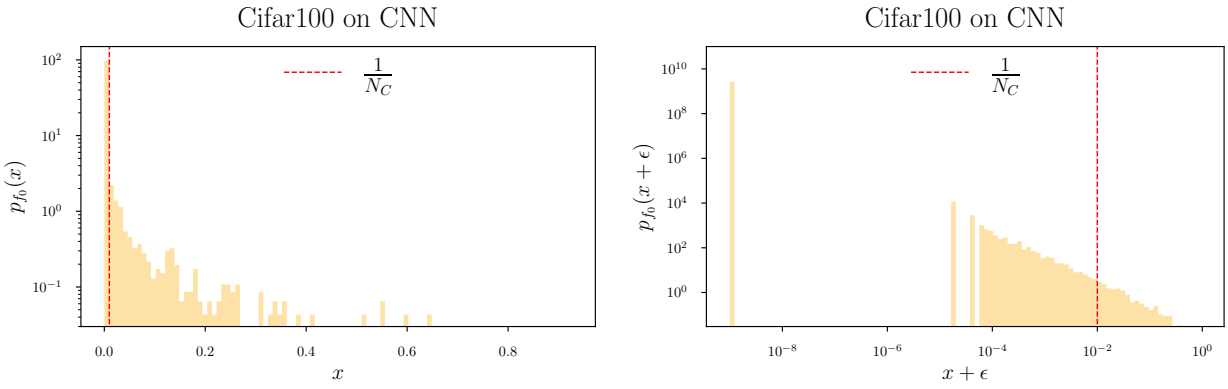

Figure I.3. `CNN-A` (ReLU + max pool case) with `C100` in this case ($N_C = 100$). The peak at $f_0 = \frac{1}{N_C}$, corresponding to the No IGB case (in the absence of IGB, the distribution should concentrate around this value), is reported as a reference. The plot on the right contains the same data as that on the left, but has a logarithmic x axis. We added a small numer, $\epsilon = 1e-9$, to the value of f_0 in order to show the peak at $f_0 = 0$.

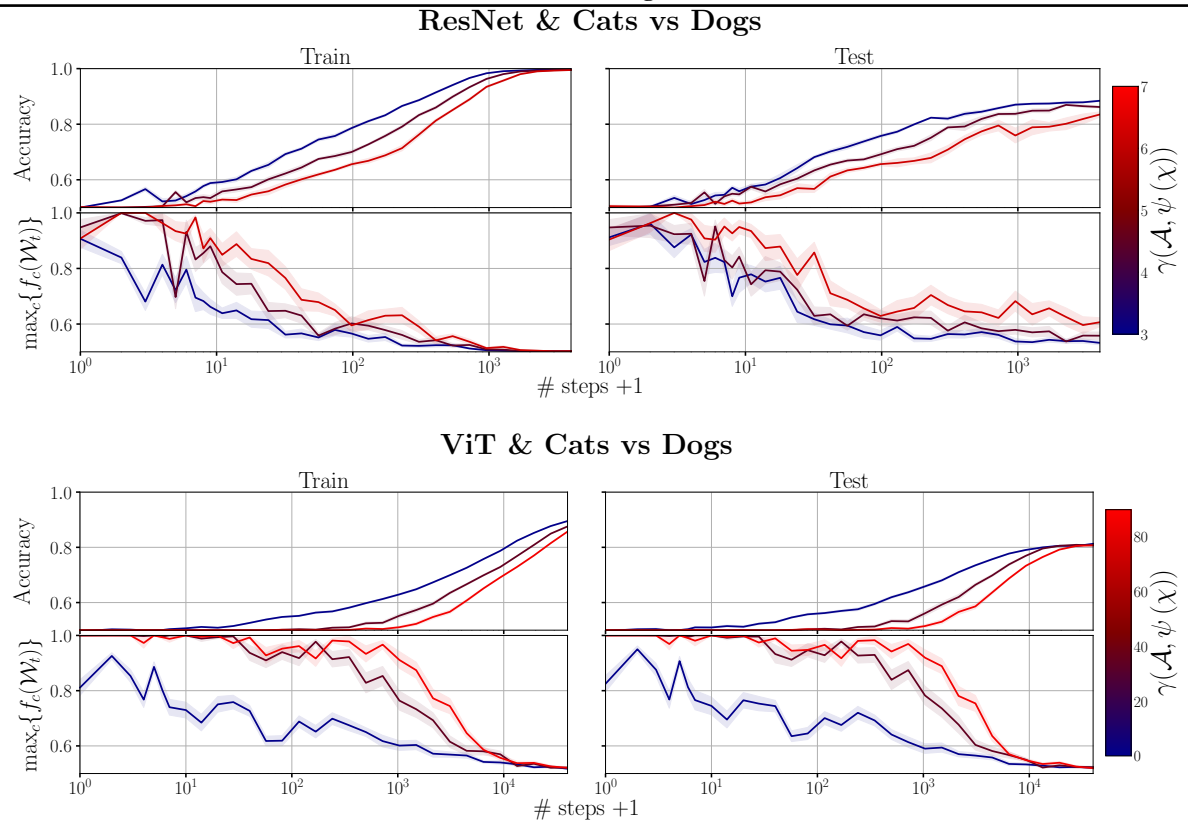

Figure I.4. **Right**: Comparison of the trend of $\max_i\{f_i\}$ with that of accuracy during the learning dynamics, varying with the level of guessing bias at initialization (IGB). The curves show a consistent pattern with the diagram on the shown in Fig. 2. The simulations were conducted on a ResNet (top) and Vision Transformer (bottom) using a binary dataset (dogs *vs* cats from CIFAR) as input; more details on the setting are provided in App. I.4.

the presence of IGB in Figs. I.4 and I.5. Specifically, IGB is identified by checking the quantity $\max_c\{f_c(\mathcal{W}_t)\}$, for $t = 0$.; we can observe that at time $t = 0$, *i.e.*, at initialization, each of these curves shows $\max_c\{f_c\} \neq 1/N_C$.[18]

I.2. Experiments on other architectures & effects on the training dynamics

The experiments presented in App. I.1 illustrate how Initial Guessing Bias (IGB) can emerge in CNNs. This section provides examples demonstrating the breadth of settings and architectures where this phenomenon is observed. In particular, beyond illustrating the emergence of IGB in additional architectures (ResNet, Vision Transformer, and MLP-mixer), we show the impact of the phenomenon on the dynamics.

While an exhaustive review is beyond the scope of this study, we present key experimental results that emphasize the practical significance of IGB, suggesting its relevance for further investigation. Given the observation of IGB in various settings, including advanced architectures that achieve state-of-the-art performance, it seems that IGB does not profoundly affect the final convergence. However, qualitative differences from non-IGB cases and similarities to class imbalance situations indicate potential challenges during training associated with IGB.

Experiments in Sec. 3 show that the time required to absorb IGB increases with the level of IGB itself. Since the absorption of IGB is necessary for performance improvement in training balanced datasets, a higher level of IGB translates to slower convergence. Similar results are observed across various settings: in Fig. I.4, for binary classification extended to other architectures (ResNet and Vision Transformers), and in Fig. I.5, the experiments are repeated for multi-class cases with analogous outcomes.

[18]The reader might have particular interest in the $K = 0$ case, where the data is centered and IGB cannot be attributed to data preprocessing.

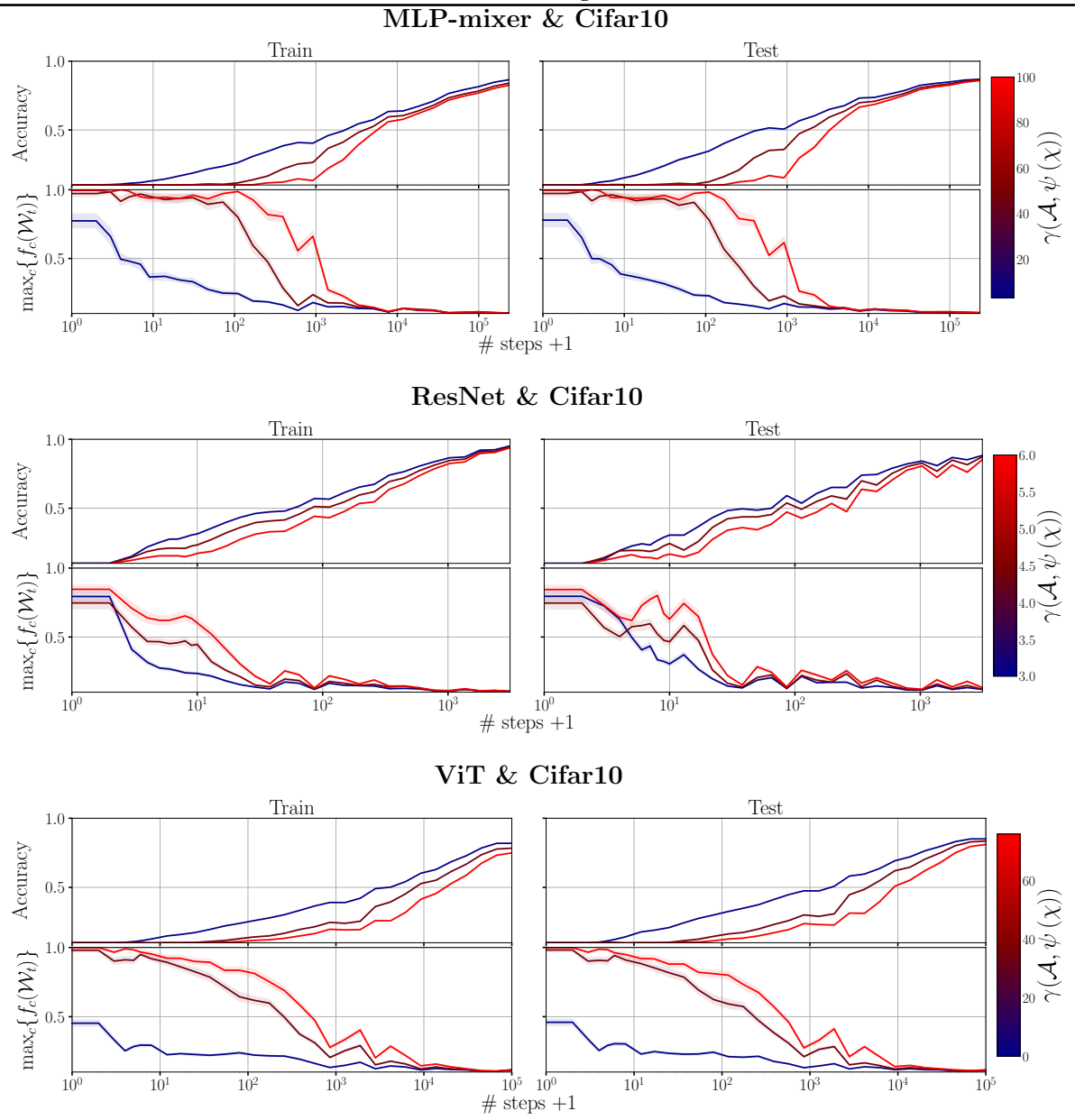

Figure I.5. **Right**: Comparison of the trend of $\max_i\{f_i\}$ with that of accuracy during the learning dynamics, varying with the level of guessing bias at initialization (IGB). The curves show a consistent pattern with the diagram on the shown in Fig. 2. The simulations were conducted on different architectures (MLP- mixer (top), ResNet (middle) and Vision Transformer (bottom)) using a multi-class dataset (`C10`) as input; more details on the setting are provided in App. I.4.

As discussed in App. G, there are various ways to increase the level of IGB. In the experiments shown (Fig. 2, Fig. I.4, Fig. I.5), IGB is tuned through data standardization, where inputs are pre-processed as:

$$\psi\left(\xi_b^{(a)}\right) = \xi_b^{(a)} + K\,. \tag{217}$$

The experiments are repeated for different values of K (specifically $K \in \{0, 2, 4\}$). Increasing $|K|$, the level of IGB also rises, measured using $\gamma(\mathcal{A}, \psi(\chi))$. This choice of IGB amplification allows for comparison without altering the architectural design or data structure (inputs are simply shifted by a constant value).

Even in the case of $K = 0$, the system exhibits IGB, as indicated by $\max_i\{f_i\} \neq 1/N_C$ at initialization. Also, it is worth noticing that the susceptibility of networks to IGB varies, as seen from the comparison of the $\gamma(\mathcal{A}, \psi(\chi))$ range in the

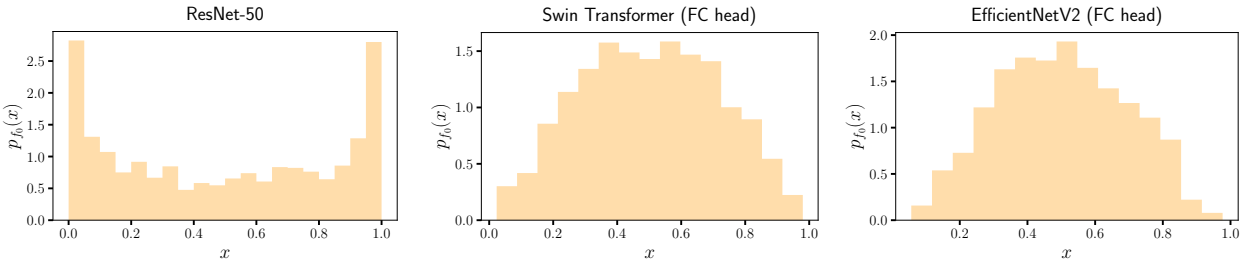

Figure I.6. The distribution $p_{f_0}(x)$ on pre-trained models. From left to right are shown `PT-ResNet`, `PT-SwinT-FC`, `PT-ENetV2-FC` on a binary dataset (Cats and Dogs from `C10`).

colormaps of different experiments. Despite using the same set of K values, the resulting IGB level varies significantly across networks. The IGB values, dependent on the combined effect of $\psi(\chi)$ and $\mathcal{A}$, show substantial variation among the considered architectures, suggesting the presence of regulatory or amplifying elements for IGB within these architectures.

I.3. IGB in Pre-Trained Models

Thus far, the analysis and experiments presented have focused on randomly initialized DNNs. While the results in App. I.2 provide valuable insights into the effects of IGB on the learning dynamics of untrained networks, we now demonstrate that even pre-trained models exhibit the presence of IGB.

Transfer learning is becoming an increasingly prevalent paradigm (Han et al., 2021), where large models are pre-trained on extensive datasets and then fine-tuned for specific, different tasks. In practice, before initiating the fine-tuning phase, the final layer, referred to as the head, is replaced by untrained layer(s). This replacement ensures compatibility with the new task, such as modifying the number of output nodes in the classifier (i.e., the last fully connected layer) to match the number of classes in the new task.

While the head could, in principle, consist solely of the classifier, other configurations with more complex structures for the head are possible (Ren et al., 2023). Using a more articulated head is a common practice, especially in situations where the head is the only part of the network trained during fine-tuning. Approaches that avoid full-model fine-tuning have recently received more attention, as they offer reduced computational costs and address privacy concerns (Xiao et al., 2023), while also ensuring better performance in the presence of out-of-distribution data with large distribution shifts (Kumar et al., 2022).

Fig. I.6 shows the presence of IGB on pre-trained models at the beginning of the fine-tuning phase. The experiments are conducted on different architectures; in each of them the last fully connected layer (classifier) is replaced to match the new number of classes and reinitialized. Interestingly, using a deeper untrained structure as head leads to an amplification of IGB as shown in Fig.I.7; the same architectures from experiments in Fig. I.6 are considered with the only difference that the fully connected head is now replaced by an untrained MLP.

I.4. Reproducibility

Here we provide technical details about the experiments, to allow for reproducibility. The code used for the experiments presented in this work are available at `https://github.com/EmanueleFrancazi/IGB-Algorithms`.

Datasets

- **Gaussian Blob** (`GB`): Consistent with the analysis presented, for most of the experiments shown we used a Gaussian blob as input. Specifically, all elements of the dataset are *i.i.d.* and each individual element consists of a random vector of $d = 3072$ *i.i.d.* normally distributed components, *i.e.*:

$$\xi_b^{(a)} \sim \mathcal{N}(0, 1) \ .$$

  Note that the value of d is chosen so that the random vectors we generate have the same dimension of `CIFAR10`.

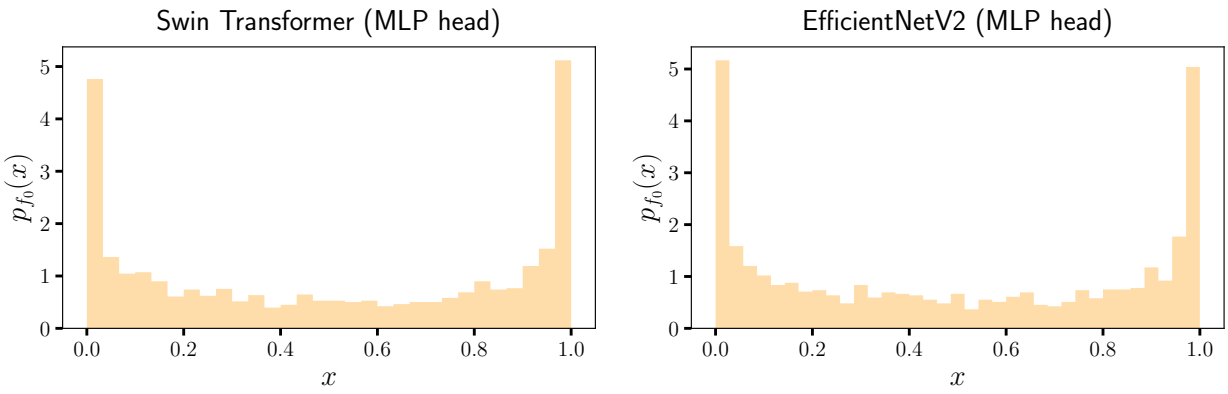

Figure I.7. The distribution $p_{f_0}(x)$ on pre-trained models. From left to right are shown `PT-SwinT-MLP`, `PT-ENetV2-MLP` on a binary dataset (Cats and Dogs from `C10`)

- **CIFAR10** (`C10`): We use `CIFAR10` (`https://www.cs.toronto.edu/~kriz/cifar.html`) (Krizhevsky et al., 2009) as an example of a real multi-class dataset. Before the start of the simulation we perform the standardization of the dataset: the pixel values are rescaled in the interval $[0, 1]$ and then shifted by the mean value and rescaled by the standard deviations (calculated on each channel).
- **CIFAR100** (`C100`): We use `CIFAR100` (`https://www.cs.toronto.edu/~kriz/cifar.html`) (Krizhevsky et al., 2009) as an example of high cardinality dataset, *i.e.* a dataset with a big number of classes.
- **MNIST** (`E&O`): We use `MNIST` (`http://yann.lecun.com/exdb/mnist/`) (Deng, 2012) to reproduce binary experiments on real data. The binary dataset is defined by merging the starting classes into two macro groups according to the parity of digits; thus, we will have the `even` number class $\{0, 2, 4, 6, 8\}$ and the `odd` number class $\{1, 3, 5, 7, 9\}$.

Models We here provide details on the architectures we used for our experiments. Our description of the models does not include the loss functions because the loss function is irrelevant in untrained networks.

- **MLP**: Our analysis provides theoretical predictions for MLP. In order to support the results of the study, we considered two different MLP networks in the proposed experiments:
  - **MLP with a single hidden layer** (`SHLP`): The number of nodes nodes in the hidden layer varies between $N_1 = 100$ for networks without Pooling and $N_1 = 500$ for networks equipped with max pooling layer. Different activation functions have been coupled to the networks to show the differences; details regarding the choice of these elements are given case by case.
  - **MLP with multiple hidden layers** (`MHLP`): In this case we have L hidden layers, each one composed by $N = 100$ nodes. As in the previous case, the activation function is an element we varied to set a comparison and underline the differences coming from this choice.
- **CNN**: To show how IGB manifests outside the setting employed in the quantitative treatment presented we propose, some experiments on convolutional neural networks (CNNs) as an alternative to the MLPs discussed above. In particular, we used:
  - `CNN-A`: This architecture was used for simulations related to the histogram in Fig. 1; two classes from CIFAR10 (three-channel images) were selected for these experiments. Starting from the input layer we have:
    - a first convolutional layer with: out channels=16 (Number of channels produced by the convolution), $m = 5$ (Size of the convolving kernel), stride=1, padding=2. The output of the layer is then passed through an activation function and a pooling layer. The choice of these elements varies to compare two different scenarios (as explained in Fig. 1). In both cases, however, we set common parameters for the pooling layer, *i.e.* $m = 2$ (kernel size), stride=2.

    - Next comes a second convolutional layer with the same parameters as the previous one, except for the number of output chanels; in this case we have out channels=64. Again the convolutional layer is followed by an activation function and a pooling layer (same as the first layer). The parameters of the pooling layer in this case are *i.e.* $m = 4$ (kernel size), stride=4. The processed signal is then connected with a weights layer to the output layer.
  - `CNN-B`: We also consider a second CNN architecture deeper than `CNN-A`. Specifically starting from the input layer:
    - we start with a sequence of five convolutional layers (each followed by an activation function and pooling layer). Except for the number of output channels the rest of the parameters are fixed the same for each of these layers, in particular we have for the convolutional layer $= 5$, stride=1, padding=4. For the pooling layer, however, $m = 5$, stride=1. Finally, the number of output channels for the various layers is $[16, 32, 32, 64, 32]$.
    - This is followed by an additional convolutional layer defined by the following parameters: out channels=16, $m = 5$, stride=1, padding=2. This is accompanied by the activation function and a Pooling layer whose parameters are: $m = 2$, stride=2.
    - Finally, a last sequence of convolutional layers, activation function and pooling layer precedes the output layer. The only parameter that differs from the sequence that precedes it is the kernel size of the pooling layer; specifically in this case $m = 4$. The processed signal is then connected with a weights layer to the output layer.

- **MLP-mixer**: We propose the MLP-mixer (`MLP-mix`) introduced in (Tolstikhin et al., 2021) as an example of a more advanced MLP architecture. We use the architectural design documented in `https://github.com/omihub777/MLP-Mixer-CIFAR/blob/main/README.md`.

- **ResNet**: We propose ResNet34 (`ResNet`), introduced in He et al. (2016) as an example of an architecture equipped with skip connections. We use the architectural design documented in `https://www.kaggle.com/code/kmldas/cifar10-resnet-90-accuracy-less-than-5-min`.

- **Vision Transformer**: We propose ViT (`ViT`), introduced in Dosovitskiy et al. (2020) as an example of an architecture equipped with multi-head attention mechanism. We use the architectural design documented in `https://github.com/tintn/vision-transformer-from-scratch/blob/main/vit.py`.

- **Pre-Trained Resnet**: ResNet50 (`PT-ResNet`) pretrained on Imagenet (Deng et al., 2009) (for more details see `https://pytorch.org/vision/stable/models/generated/torchvision.models.resnet50.html#torchvision.models.resnet50`). The head is substituted by an untrained classifier (fully connected layer) randomly initialized with Kaiming normal weights and null bias.

- **Pre-Trained Swin Transformer**: swin tiny architecture pretrained on Imagenet (Deng et al., 2009) (for more details see `https://pytorch.org/vision/main/models/generated/torchvision.models.swin_t.html#torchvision.models.swin_t`
  - `PT-SwinT-FC`: The classifier (last fully connected layer) is randomly re-initialized with Kaiming normal weights and null bias.
  - `PT-SwinT-MLP`: The classifier is substituted by a multi-layer perceptron (5 layers) with ReLU activation function randomly initialized with Kaiming normal weights and null bias.

- **Pre-Trained EfficientNet**: EfficientNetV2-S architecture pretrained on Imagenet (Deng et al., 2009) (for more details see `https://pytorch.org/vision/main/models/generated/torchvision.models.efficientnet_v2_s.html#torchvision.models.efficientnet_v2_s`)
  - `PT-ENetV2-FC`: The classifier (last fully connected layer) is randomly re-initialized with Kaiming normal weights and null bias.
  - `PT-ENetV2-MLP`: The classifier is substituted by a multi-layer perceptron (5 layers) with ReLU activation function randomly initialized with Kaiming normal weights and null bias.

Note that in both the CNN architectures we use, the final convolutional layer is directly connected to the output, without any additional fully-connected hidden layer (as the ones described by our theory). This indicates that the observed IGB is also also a feature of CNNs independently of the presence of fully-connected hidden layers at the end of the network (which as we proved would also cause IGB).

Dynamics The simulations concerning the dynamics (Sec. 3 App. I.2) use three different architectures (`ResNet`, `MLP-mix`, `ViT`) following the settings proposed in their respective repositories. For the `ViT`simulations, the dataset was augmented following https://github.com/omihub777/ViT-CIFAR/blob/main/autoaugment.py in order to slightly improve performance compared to the baseline documented in the repo. We highlight that the proposed architectures were not selected as representations of the state of the art, but rather as examples of well-documented architectures capable of achieving good performance quickly (*e.g.* https://www.kaggle.com/code/kmldas/cifar10-resnet-90-accuracy-less-than-5-min). In this way, we not only show the prevalence of IGB, observed in each of these architectures introduced before our work. But, at the same time, the proposed examples do not require excessive resources, facilitating reproducibility and making the results of the work more accessible.

J. Limitations and ethics

Limitations Our work focuses on systems simple enough to clearly show the main aspects of the phenomenon and at the same time complex enough to investigate non-trivial effects induced, for example, by network depth. A comprehensive picture emerges that clarifies the effect of some particular elements of the architecture and their connection with IGB. On the other hand, the observation of IGB is not restricted to the subset of networks/datasets considered in the study. Although the treatment of more articulated setups is outside the scope of the study, whose main goal is to present the phenomenon (which to the best of our knowledge has never been reported in the literature) in a clear manner, the characterization of IGB for more realistic systems remains an interesting question.

Ethics By informing model selection, data preparation and initial conditions, our results can improve the training of machine learning models. Better-performing machine learning models allow to better address wide ranges of problems, but can also be adapted for potentially harmful applications (Hutson, 2021; Qadeer & Millar, 2021).

The CIFAR datasets, are subsets of the 80 million tiny images, which are formally withdrawn since it contains some derogatory terms as categories and offensive images (http://groups.csail.mit.edu/vision/TinyImages/). However, note that none of the experiments described in the paper was performed on the overall tiny images dataset: the said derogatory images are not present in CIFAR10 nor CIFAR100.