Journal of Artificial Intelligence Research 82 (2025) 503-562 Submitted 08/2024; published 02/2025

PAC-Chernoff Bounds: Understanding Generalization in the Interpolation Regime

Andrés R. Masegosa ARMA@CS.AAU.DK
Department of Computer Science
University of Aalborg

Luis A. Ortega LUIS.ORTEGA@UAM.ES
Machine Learning Group
Department of Computer Science
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Abstract

This paper introduces a distribution-dependent PAC-Chernoff bound that exhibits perfect tightness for interpolators, even within over-parameterized model classes. This bound, which relies on basic principles of Large Deviation Theory, defines a natural measure of the smoothness of a model, characterized by simple real-valued functions. Building upon this bound and the new concept of smoothness, we present an unified theoretical framework revealing why certain interpolators show an exceptional generalization, while others falter. We theoretically show how a wide spectrum of modern learning methodologies, encompassing techniques such as ℓ_2-norm, distance-from-initialization and input-gradient regularization, in combination with data augmentation, invariant architectures, and over-parameterization, collectively guide the optimizer toward smoother interpolators, which, according to our theoretical framework, are the ones exhibiting superior generalization performance. This study shows that distribution-dependent bounds serve as a powerful tool to understand the complex dynamics behind the generalization capabilities of over-parameterized interpolators.

1. Introduction

In modern machine learning, model classes have such large capacity that optimizers virtually always retrieve a model *interpolating* the training data; that is, with null training error (Zhang et al., 2017). These models are called *interpolators* and it is well known that there are many within a large model class (Livni et al., 2014) and that some of them have a remarkably small generalization error (difference between "training error" and "expected or test error") while others do not (Feldman & Zhang, 2020).

The machine learning community has made a great effort during the last years to understand why modern learning algorithms retrieve, most of the time, interpolators with a small generalization error (Nagarajan & Kolter, 2019b; Bartlett et al., 2020). Most of these attempts often focus on providing generalization bounds. These bounds provide an upper limit on the expected error, tying it to variables related to the training dataset and the model produced by a learning algorithm. Such bounds typically resemble the following form,

$$L(\mathcal{A}(D)) \leq \hat{L}(\mathcal{A}(D), D) + \mathcal{C}(\mathcal{A}(D), D, \delta)\,, \tag{1}$$

where D represents the training dataset, independently and identically distributed (i.i.d.) from a data-generating distribution ν; $\mathcal{A}$ denotes the learning algorithm, which generates a hypothesis (a model) from a given data set (for example, via the gradient descent method); L and $\hat{L}$ indicate the expected and empirical errors of a hypothesis, respectively; $\mathcal{C}$ denotes a complexity measure; and the inequality holds with high probability, at least, $1-\delta$ over draws of datasets from the distribution ν.

Examples of bounds falling inside this scheme include Vapnik–Chervonenkis (VC) bounds (Vapnik & Chervonenkis, 2015; Bartlett et al., 2019), Rademacher bounds (Mohri et al., 2018), (Norm and margin)-based bounds (Nagarajan & Kolter, 2017; Neyshabur et al., 2015) and Sharpness-based measures (Nagarajan & Kolter, 2019a; Keskar et al., 2017a). Even PAC-Bayes with a fixed prior (McAllester, 1999) can be included here if we consider algorithms retrieving a distribution over the models.

However, many recent works are increasingly suggesting that these bounds are probably vacuous in the over-parameterized regime and, in consequence, unable to explain generalization in this setting. Zhang et al. (2017) was the first one to provide quite convincing empirical evidence of this in the context of deep neural networks. Nagarajan and Kolter (2019b) later provided a series of experiments and theoretical results in the same direction. These works also show that even in the case where the complexity measure $\mathcal{C}$ depends directly on the algorithm $\mathcal{A}$ rather than simply in its output $\mathcal{A}(D)$ (for example, assuming the algorithm always retrieve models whose parameters' norm is lower than a given constant) are also vacuous. Recently, Wang et al. (2024) showed how any near-interpolator exhibits rapid parameter norm growth, which implies that existing data-dependent parameter-norm-based bounds are necessarily loose, concluding that *"explaining the generalization capability of near-interpolators will require new tools"*. More generally, Gastpar et al. (2024) managed to show that, under some conditions resembling over-parameterization, there is no generalization bound like Equation (1) tight for all data-generating distributions, and that algorithm-dependent bounds are also provably vacuous. For all these reasons, the emerging conclusion is that:

Bounds that solely depend on the training data are provably vacuous for over-parameterized model classes and are unable to explain generalization.

The aforementioned works (Nagarajan & Kolter, 2019b; Nagarajan, 2021; Gastpar et al., 2024) directly or indirectly advocate for the need of exploring alternative bounds that not solely depend on the training data but also use information of the data-generating distribution. These bounds are usually referred to as *distribution-dependent bounds*; where the complexity term $\mathcal{C}$ explicitly depends on the data-generating distribution ν. Distribution-dependent bounds have been used before in different contexts (Zhang, 2006; Catoni, 2007), however, to the best of our knowledge, none of these bounds is known to be tight or, at least, non-vacuous for over-parameterized interpolators. Even more importantly, these kind of bounds have not been used before to analyze the generalization of over-parametrized interpolators.

1.1 Our Contribution

In this study, we investigate the use of distribution-dependent bounds within a finite hypothesis space. Our aim is to understand how modern learning techniques manage to produce models that both interpolate the data and have a small generalization error. More precisely, Theorem 4.1 introduces a distribution-dependent PAC-Chernoff bound that is *perfectly tight* for any algorithm retrieving models interpolating the training data, even when the model class is over-parameterized. This bound has the following standard form

$$L(\mathcal{A}(D)) \leq \hat{L}(\mathcal{A}(D), D) + \mathcal{C}(\mathcal{A}(D), n, \nu, \delta)\,,$$

where, now, the complexity term $\mathcal{C}$ also depends on the data-generating distribution ν. Proposition 4.4 shows that with high-probability $1-\delta$, that

$$\mathcal{C}(\mathcal{A}(D), n, \nu, \delta) \leq L(\mathcal{A}(D)) \leq \hat{L}(\mathcal{A}(D), D) + \mathcal{C}(\mathcal{A}(D), n, \nu, \delta)\,,$$

which ensures that the proposed bound is *perfectly tight for interpolators*, that is to say, to models whose empirical loss is null or small enough to be considered negligible or within an acceptable error margin, denoted as $\hat{L}(\mathcal{A}(D), D) \leq \epsilon$.

As we show in this work, these bounds are directly connected to Large Deviation Theory (LDT) (Ellis, 2012) because their complexity measure $\mathcal{C}(\mathcal{A}(D), n, \nu, \delta)$ directly depends on the so-called *rate function* (also known as the Cramér-Chernoff function), which is the central element of LDT. We employ the *rate function* to present a new characterization of the *smoothness* of a model using distribution-dependent measures. According to Theorem 4.6, this approach enables a precise characterization of *which interpolators better generalize*, addressing an outstanding open question in machine learning.

Despite being based on oracle quantities, we show that the theoretical framework built around this complexity measure based on the rate function allows to create an unified explanation of a wide range of learning techniques used in modern machine learning. Namely, ℓ_2-norm, distance from initialization and input-gradient regularization (Section 6), invariant architectures, data augmentation (Section 7) and over-parameterization (Section 8). Under this framework, we show why each of these learning techniques produce interpolators that generalize well and why and when they complement each other (for example, why ℓ_2-norm regularization is also effective in combination with the use of invariant architectures).

We *do not claim* that the proposed theoretical analyses of each of these learning techniques are better than existing ones at individual level. For example, we do not claim that our explanation about why ℓ_2-norm promotes generalization is better than existing ones, for example, in the context of linear regression (Bartlett et al., 2020). However, we *do claim* that using the rate function and PAC-Chernoff bounds, which are distribution-dependent quantities, it is possible to jointly explain, up to some degree, the generalization of interpolators, the role of over-parameterization and why many widely used modern learning techniques find interpolators that generalize. To the best of our knowledge, this is the first theoretical framework capable of achieving insights that covers such a wide range of learning techniques. At the same time, this work shows that distribution-dependent bounds are a promising research direction for understanding the generalization of modern machine learning methods.

2. Preliminaries

First of all, let us introduce the notation and the assumptions of this work. Let D denote a given *training data set* with size $n > 0$, which we assume has been i.i.d. generated from an unknown distribution, denoted by $\nu(\boldsymbol{y}, \boldsymbol{x})$. We assume we have a model class parameterized by a parameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. We denote for any model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, its loss as $\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$, its expected loss as $L(\boldsymbol{\theta}) = \mathbb{E}_\nu[\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})]$, and its empirical loss as $\hat{L}(\boldsymbol{\theta}, D) = \frac{1}{n}\sum_i \ell(\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\theta})$. Finally, the subset of models having a null variance (constant predictions) is defined as $\boldsymbol{\Theta}_0 := \{\boldsymbol{\theta} \in \boldsymbol{\Theta} : \mathbb{V}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})) = 0\}$.

We also consider that our model class *lives* inside a finite-precision computing machine. In that sense, we assume our model class is represented by p parameters. Then, we will have k^p different models, under a numeric precision of $\log_2(k)$ bits (for example, the value of k would be $k = 2^{32}$ for single floating point precision and $k = 2^{64}$ for double floating point precision). This model class defines a finite grid (using a discretization approach) contained in $\mathbb{R}^p$, but it does not imposed any restriction in the definition of the loss, which can be defined in a continuous space. In Section 10, we discuss how the results presented here could be potentially extended to infinite model classes by using recently introduced PAC-Bayes-Chernoff bounds (Casado et al., 2024). Let us now introduce the specific assumptions of this work:

Assumption 1. *The loss is lower-bounded; more precisely, the essential infimum of the loss is finite and positive. That is,* $m_{\boldsymbol{\theta}} := \operatorname{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\,\in\,\mathrm{supp}(\nu)} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \geq 0 \quad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$. *And the expected loss is always finite,* $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}\ \ L(\boldsymbol{\theta}) < \infty$.

This assumption assures that the loss function is lower-bounded and the generalization error is finite (not necessarily upper-bounded). The first part of the assumption is naturally satisfied in many standard problems; for example, in multiclass classification problems with softmax activation and the cross-entropy loss and regression problems with mean squared error, as it is clear that $m_{\boldsymbol{\theta}} = 0$. The second part of the assumption ($L(\boldsymbol{\theta}) < \infty$) is mainly introduced for the sake of simplicity in the mathematical exposition. In fact, this could be relaxed in many of the theoretical results of this work, under some considerations. However, under the scope of this work, that is studying the generalization error of interpolators, assuming a finite expected error, and thus, a finite generalization error, is a reasonable simplification for the exposition of the theoretical results.

Some extra considerations regarding this assumption raise in the case of a Gaussian likelihood and log-loss, which can be considered another *usual setting* in machine learning. In this setup, the density has to be lower than one for any data sample and the variance of the Gaussian distribution can not be null in order to satisfy Assumption 1. However, this is not restrictive; lower than one Gaussian densities is usual in high dimensional Gaussians and can be ultimately imposed with a restriction on the variances. On the other hand, non-zero variances are usually *desirable* to ensure stability of machine learning models. In fact, variances are typically restricted to be positive or computed as the exponential of a logarithm-scaled variable, ensuring their positiveness.

As an example of a case not considered under Assumption 1, the loss function cannot be Gaussian-distributed, as $m_{\boldsymbol{\theta}} = -\infty$ in this case. In this regard an exponentially-distributed loss could be used verifying Assumption 1.

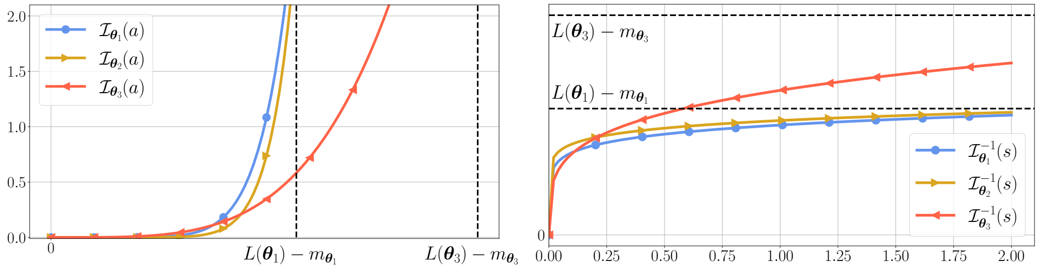

Figure 1: Illustration on different *rate function* (left) and *inverse rate function* (right) with the same or different domains of definition: three different models $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3 \in \boldsymbol{\Theta}$ are shown where $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ share the same definition interval for their rate functions.

3. The Rate Function

We start introducing the key function of this work, the so-called *rate function*, denoted by $\mathcal{I}_{\boldsymbol{\theta}}(a)$. The rate function places a central role in Large Deviation Theory (LDT) (Ellis, 2012), a branch of probability theory tightly connected to statistical mechanics, that deals with understanding the behavior of rare events or large fluctuations in random systems. The rate function is normally used to understand the underlying structure of these events and how their probabilities change as they move away from the typical or average behavior.

Mathematically, the rate function is defined as the *Legendre transform* of the *cummulant-generating function*, denoted by $J_{\boldsymbol{\theta}}(\lambda)$, which is the natural logarithm of the *moment-generating function* of a random variable, in this case $L(\boldsymbol{\theta}) - \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ with $\boldsymbol{y}, \boldsymbol{x} \sim \nu(\boldsymbol{y}, \boldsymbol{x})$.

Definition 1 (Rate Function). *For any model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, its rate function is a real-valued function $\mathcal{I}_{\boldsymbol{\theta}} : [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}) \to \mathbb{R}^+_0$, defined as*

$$\mathcal{I}_{\boldsymbol{\theta}}(a) = \sup_{\lambda > 0} \; \lambda a - J_{\boldsymbol{\theta}}(\lambda) \quad \forall a \in [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})\,,$$

where the cummulant-generating function is a real-valued function $J_{\boldsymbol{\theta}} : \mathbb{R}^+_0 \to \mathbb{R}^+_0$, defined as

$$J_{\boldsymbol{\theta}}(\lambda) = \ln \mathbb{E}_\nu \left[e^{\lambda(L(\boldsymbol{\theta}) - \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}))} \right] \quad \forall \lambda \geq 0\,.$$

The inverse of the rate function, denoted $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s)$, will also place a relevant role in this work.

Definition 2 (Inverse Rate Function). *For any model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the inverse rate function is a real-valued function $\mathcal{I}^{-1}_{\boldsymbol{\theta}} : \mathbb{R}^+_0 \to [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}]$, defined as*

$$\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s) = \inf_{\lambda > 0} \frac{J_{\boldsymbol{\theta}}(\lambda) + s}{\lambda} \quad \forall s \geq 0\,.$$

Note that, when $\mathbb{P}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}}) \neq 0$, for $\forall s \geq -\ln \mathbb{P}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}})$, $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s)$ is constantly equal to $L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$ and, in consequence, $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s)$ is a *generalized inverse* of $\mathcal{I}_{\boldsymbol{\theta}}(a)$ (Rockafellar, 1970). For the sake of simplicity, we note that we abuse notation and assume throughout the rest of work that for $a > L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$, $\mathcal{I}_{\boldsymbol{\theta}}(a) = \infty$. We also assume that binary operators apply to this value following common sense. According to the following proposition, both the rate function and its inverse are well defined for models satisfying Assumption 1.

Proposition 3.1 (Proof). *Under Assumption 1, $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ and $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(\cdot)$, are well defined. That is, $\forall a \in [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$, $\mathcal{I}_{\boldsymbol{\theta}}(a) < \infty$ and $\forall s \in \mathbb{R}^+_0$, $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s) < \infty$.*

Finally, the following result states some properties of the rate and inverse rate function, shedding some light on their monotony, curvature and shape. In combination with Figure 1, this result should provide some intuition on how this two functions behave in general.

Proposition 3.2 (Rockafellar (1970)). *For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the rate function $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ and the inverse rate function $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(\cdot)$ satisfy the following properties,*

(i) *$\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ is convex and $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(\cdot)$ is concave; both monotonically increasing.*

(ii) $\lim_{a\to 0} \frac{\partial}{\partial a}\mathcal{I}_{\boldsymbol{\theta}}(a) = 0$ *and* $\lim_{s\to 0} \frac{\partial}{\partial s}\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s) = +\infty$.

(iii) $\lim_{a\to (L(\boldsymbol{\theta})-m_{\boldsymbol{\theta}})^-} \mathcal{I}_{\boldsymbol{\theta}}(a) = -\ln \mathbb{P}(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}})$ *and* $\lim_{s\to -\ln \mathbb{P}(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})=m_{\boldsymbol{\theta}})} \mathcal{I}^{-1}_{\boldsymbol{\theta}}(s) = L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$.

(iv) *$\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ and $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(\cdot)$ are invariant to reparameterizations.*

Note that the rate function of a Gaussian random variable, like many other standard random variables, does not have a vertical asymptote. However, we should then consider that if the loss follows a Gaussian distribution, it will not satisfy the first condition in Assumption 1, as the essential infimum is not finite.

The relevance of the rate function is consequence of the following results; firstly, the classic Chernoff bound, defines how likely is to observe, over different i.i.d. data sets, an empirical loss $\hat{L}(D, \boldsymbol{\theta})$ that deviates from the expected loss $L(\boldsymbol{\theta})$ by a positive quantity $a > 0$.

Theorem 3.3 (Chernoff (1952)). *For any fixed $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $a > 0$, it satisfies*

$$\mathbb{P}_{D\sim\nu^n}\Big(L(\boldsymbol{\theta}) - \hat{L}(D, \boldsymbol{\theta}) \geq a\Big) \leq e^{-n\mathcal{I}_{\boldsymbol{\theta}}(a)}\,.$$

The above bound is relevant in this work because of two main factors that characterize its behavior w.r.t. the data set size n and the generalization error gap a: on one hand, Chernoff's bound is known to be quite loose in the mean of the variable (when $a \approx 0$) but tight on the tail (when $a \approx L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$). As a result, the bound is specially useful when talking about interpolators, where the value of $L(\boldsymbol{\theta}) - \hat{L}(D, \boldsymbol{\theta})$ is close to its maximum possible value ($a = L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$), as stated in the following result.

Proposition 3.4 (Proof). *For any fixed $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $n > 0$, it satisfies*

$$\lim_{a\to L(\boldsymbol{\theta})-m_{\boldsymbol{\theta}}} \mathbb{P}_{D\sim\nu^n}\Big(L(\boldsymbol{\theta}) - \hat{L}(D, \boldsymbol{\theta}) \geq a\Big) = \lim_{a\to L(\boldsymbol{\theta})-m_{\boldsymbol{\theta}}} e^{-n\mathcal{I}_{\boldsymbol{\theta}}(a)}\,.$$

On the other hand, Cramér's Theorem (Cramér, 1938) states that Chernoff's bound is exponentially tight for *large* n. Formally, this statement is written as follows,

Theorem 3.5 (Cramér (1938); Ellis (2012)). *For any fixed $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and any $a > 0$, it satisfies*

$$\lim_{n\to\infty} -\frac{1}{n} \ln \mathbb{P}_{D\sim\nu^n}\Big(L(\boldsymbol{\theta}) - \hat{L}(D, \boldsymbol{\theta}) \geq a\Big) = \mathcal{I}_{\boldsymbol{\theta}}(a)\,.$$

Inception	Crop	L2	Train Acc.	Test Acc.	Test NLL	ℓ_2-norm
Standard	no	no	99.99%	84.36%	0.65	304
Crop	yes	no	99.94%	86.89%	0.58	309
L2	no	yes	100.0%	86.60%	0.49	200
L2-Crop	yes	yes	99.98%	88.45%	0.42	130
Random	no	no	100.0%	10.13%	5.52	311
Initial	-	-	10.00%	10.00%	2.30	593

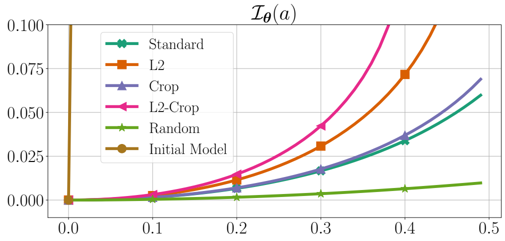

Figure 2: Metrics of Inception models on Cifar10 using ℓ_2 regularization and/or random cropping (Crop), and randomly sampled class labels (Random). The corresponding *rate functions* are shown on the right.

In LDT, the above asymptotic result is intuitively interpreted using the following equality,

$$\mathbb{P}_{D\sim\nu^n}\Big(L(\boldsymbol{\theta}) - \hat{L}(D,\boldsymbol{\theta}) \geq a\Big) = e^{-n\mathcal{I}_{\boldsymbol{\theta}}(a)+o(n,a)}\,, \tag{2}$$

which shows that the exact expression of $\mathbb{P}_{D\sim\nu^n}(L(\boldsymbol{\theta}) - \hat{L}(D,\boldsymbol{\theta}) \geq a)$ is defined by the rate function, up to a sub-exponential term, that is negligible when n is *large*, because $\lim_{n\to\infty}\frac{o(n,a)}{n} = 0$ and, in consequence, it does not have any meaningful effect. Then, according to LDT, when n is large, the rate function would be the key quantity describing the generalization error of a model, that is, statistical behavior of the difference between the expected $L(\boldsymbol{\theta})$ and the empirical loss $\hat{L}(D,\boldsymbol{\theta})$.

> The Chernoff bound, governed by the rate function $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$, is tighter for larger datasets (when $n\to\infty$) and for models interpolating the data ($a \approx L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$), which are the settings of modern machine learning.

The Chernoff bound states that the generalization error a model is governed by its rate function. The rate function is an *oracle* and *distribution-dependent* quantity because it depends on the data-generating distribution ν. In machine learning, this distribution ν is always unknown, which may be considered as a problem. However, as discussed in the introduction, we will show how the use of a distribution-dependent quantity allows to explain, for example, why data-augmentation works, why we need over-parameterized model families, and why we should consider invariant architectures, like convolutional neural networks. Furthermore, in this work, we also show how the rate function can be easily estimated from an independent data set, if it is used as a proxy for the data-generating distribution. This is a common practice in machine learning by making use of a separated *validation data set* to get estimates of $L(\boldsymbol{\theta})$ without incurring in *data snooping*. In Appendix A, we show that the rate function $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ can be easily estimated by *running* the model once on this validation data set and using log-sum-exp operations to estimate $J_{\boldsymbol{\theta}}(\lambda)$; and using a simple grid search to optimize the value of λ in Definition 1. *The fact that the cumulant and rate function can be easily estimated and plotted (as in Figure 2) is, in our opinion, a relevant finding of this work.* This will allow us to empirically illustrate and validate the predictions made by our theoretical framework.

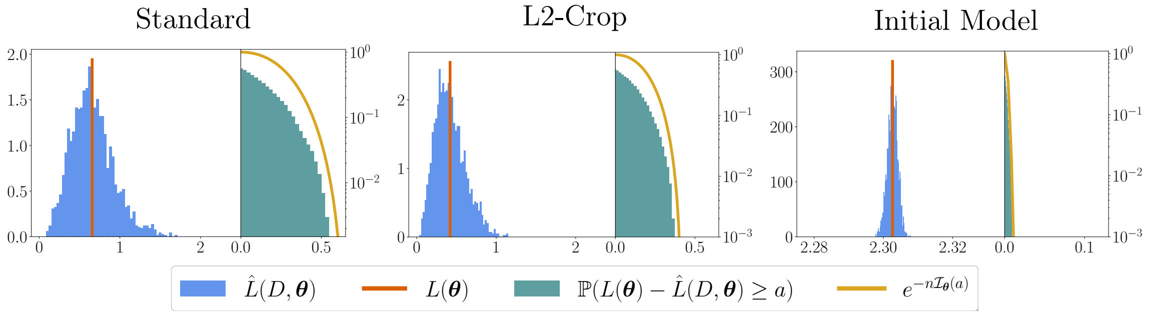

Figure 3: Illustrations on the distribution of $\hat{L}(D,\boldsymbol{\theta})$ for data sets of size $n = 50$, its concentration around the generalization error $L(\boldsymbol{\theta})$ and Chernoff's inequality for increasing values of $a \in [0, L(\boldsymbol{\theta}))$. All these quantities have been approximated using the test set of Cifar10 data set for 3 of the Inception models used in Figure 2.

Let us briefly illustrate why the rate function is useful to understand the generalization of interpolators. Figure 2 displays an estimation of the rate function $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ for some neural network models used in Zhang et al. (2017). This work demonstrated that, within the same model class, interpolators that merely memorize the training data (as represented by "Random" in the figure, which has been learned using a random-labeled training data set) can co-exist with others that generalize exceptionally well, as the ones learned using weight-norm regularization (labeled as "L2") and/or data-augmentation techniques (random-cropping labeled as "Crop"). According to Chernoff's bound, models with a larger rate function are less likely to have significant disparities between their expected and empirical losses; in other words, the empirical loss $\hat{L}(D,\boldsymbol{\theta})$ is more concentrated around its mean $L(\boldsymbol{\theta})$. Figure 3 illustrates this fact for three of these models by plotting histograms that showcase the distribution of $\hat{L}(D,\boldsymbol{\theta})$ across various data sets of size $n = 50$ (retrieved from the test set). From the histograms, it is clear that the concentration of $\hat{L}(D,\boldsymbol{\theta})$ varies among the models; the initial model, defined by *Kaiming or He initialization* (Goodfellow et al., 2016), has a prominent rate function. For this model, $\hat{L}(D,\boldsymbol{\theta})$ is tightly concentrated around its mean, $L(\boldsymbol{\theta}) = \ln 10$, observable through the minute scale of its x-axis. Comparatively, the *Standard* model, trained using SGD and characterized by a smaller rate function, has a more dispersed distribution around its mean $L(\boldsymbol{\theta}) = 0.65$. This dispersion is notably wider than that of the *L2-Crop* model. The latter, also trained with SGD but incorporating both ℓ_2 regularization and data-augmentation, has a larger rate function than the *Standard* model.

The main point of Figures 2 and 3 is to illustrate how $\hat{L}(D,\boldsymbol{\theta})$ is a random variable that could be either highly or poorly concentrated around its mean $L(\boldsymbol{\theta})$, something which is captured by the rate function of the model. When a model interpolates the training data, we must think we are observing a realization of this random variable $\hat{L}(D,\boldsymbol{\theta})$ for a particular data set D. Consequently, when comparing two models that interpolate the training data, we should choose the one that is more concentrated around its mean (higher rate function); because it will likely have a smaller expected loss. For example, Figure 3 (left) and (center) describe the distribution of $\hat{L}(D,\boldsymbol{\theta})$ for two models, Standard and L2-Crop, respectively. As both models interpolate the training data, we should prefer L2-Crop because $\hat{L}(D,\boldsymbol{\theta})$ is more concentrated around its expected value for that model.

4. The Rate Function Characterizes the Generalization of Interpolators

In this section, we showcase the main results of this work, Theorems 4.1 and 4.6, which show how the (inverse) rate function can characterize the generalization performance of interpolators. Furthermore, we present a new notion of smoothness of a model which perfectly aligns with the following statement: *a (sufficiently) smoother interpolator generalizes better.*

4.1 Tight Distribution-Dependent Bounds for Over-parameterized Interpolators

As discussed in the introduction, current high-probability generalization bounds in the form of Equation (1) have been unable so far to explain the generalization of over-parameterized interpolators. Different works (Nagarajan & Kolter, 2019b; Nagarajan, 2021; Gastpar et al., 2024; Wang et al., 2024) have also shown that the problem lies in the impossibility of having tight generalization bounds solely depending on the training data for over-parameterized model classes. Gastpar et al. (2024) even concludes that *"a bound without explicit distributional assumptions is likely to be not tight"*. Thus, an open question raises:

Open Question 1

Are there tight distribution-dependent bounds for over-parameterized models?

The answer to this question does not seem to be straightforward given that Nagarajan & Kolter (2019b) also showed examples of distribution-dependent bounds (Nagarajan & Kolter (2019b)'s Definition 3.3) which are not tight. However, the following result shows an uniform-convergence bound which applies simultaneously over the model class,

Theorem 4.1 (Proof). **(PAC-Chernoff Bound).** *With h.p.* $1-\delta$ *over* $D \sim \nu^n$*, for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$*, simultaneously,*

$$L(\boldsymbol{\theta}) \leq \hat{L}(D, \boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) .$$

During the rest of this work, many results will resemble the same structure used in Theorem 4.1: "*with h.p.* $1-\delta$ *over* $D \sim \nu^n$*, for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$*, simultaneously...*". This kind of enunciating is common in PAC theory but might be confusing to other readers. We want to highlight that the above theorem states that *for any* $\delta \in (0,1)$*, the event* $\boldsymbol{Event}(\boldsymbol{\theta}, D, \delta) := L(\boldsymbol{\theta}) \leq \hat{L}(D, \boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$ *will hold for every* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ *unless you have drawn a "bad" data set* $D \sim \nu^n$*; that is, the event holds with probability at least* $1-\delta$*.* Mathematically:

$$\forall \delta \in (0,1), \quad \mathbb{P}_{D\sim\nu^n}\Big(\bigcap_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbf{Event}(\boldsymbol{\theta}, D, \delta)\Big) \geq 1-\delta\,.$$

Many results will resemble this structure through the text where the "event" might be an inequality ($A \leq B$) as in Theorem 4.1, or an implication ($A \implies B$) as in Proposition 4.4.

Going back to Theorem 4.1, in the above bound, $\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$ defines a complexity measure for the model $\boldsymbol{\theta}$ in the context of a model class defined by p parameters, the training data set of n samples and, according to the own definition of $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(\cdot)$ (see Definition 2), it also depends on the data-generating distribution ν. From Proposition 3.2, this complexity measure $\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$ monotonically grows with the size of the model class and monotonically decreases with the level of confidence δ and the size of the training data.

It is common to find models, within the same model class, that define the same loss function, $\ell(\cdot,\cdot,\boldsymbol{\theta}_1) = \ell(\cdot,\cdot,\boldsymbol{\theta}_2)$. For instance, in a Multi-Layer Perceptron (MLP), weights can be permuted in specific ways without altering model predictions, and models with zeros in the last layer produce the same predictions regardless of the weights in other layers. Consequently, when calculating the size of the model class, we can effectively "exclude" models that define the same empirical loss across different datasets. Let $\bar{\boldsymbol{\Theta}} \subseteq \boldsymbol{\Theta}$ be a subset of the model class where, if $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \bar{\boldsymbol{\Theta}}$, there exists a dataset $D \sim \nu^n$ such that $\hat{L}(D,\boldsymbol{\theta}) \neq \hat{L}(D,\boldsymbol{\theta}')$. The following result provides a refined bound, based on the size of $\bar{\boldsymbol{\Theta}}$.

Corollary 4.2 (Proof)**.** *With h.p. $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{|\bar{\boldsymbol{\Theta}}|}{\delta}\right) .$$

For the remainder of this work, we keep the term k^p in all of our results. However, in each case, we can replace k^p with the typically smaller term $|\bar{\boldsymbol{\Theta}}|$ by applying the result above. The following result shows that, when n goes to infinity, the proposed complexity measure converges with rate $1/\sqrt{n}$ to the (scaled) standard deviation of the loss function.

Theorem 4.3 (Proof)**.** *For any $\delta \in (0,1)$ and any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, it verifies that*

$$\lim_{n\to\infty} \sqrt{n}\,\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) = \sqrt{2\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))\ln\frac{k^p}{\delta}} .$$

The relevant property of this novel complexity measure is that it is a *perfectly tight proxy* of the expected loss $L(\boldsymbol{\theta})$ for models that interpolate the training data, $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$, even if the model class is over-parameterized,

Proposition 4.4 (Proof)**.** *With h.p. $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$\textit{if} \quad \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \quad \textit{then} \quad 0 \leq L(\boldsymbol{\theta}) - \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) \leq \epsilon . \tag{3}$$

Note that in the above result , the event that holds with probability $1-\delta$ is the implication in Equation 3. Using there results, we can derive an algorithm-distribution-dependent bound that is *perfectly tight* for any data distribution and any learning algorithm; even for over-parameterized model classes. However, for this bound to be tight it is imperative that the model interpolates the training data. Formally, for any fixed $\epsilon > 0$, with $2^{\mathcal{X}\times\mathcal{Y}}$ denoting the power set of all possible data sets of any size; let $\mathcal{A}_\epsilon : 2^{\mathcal{X}\times\mathcal{Y}} \rightarrow \boldsymbol{\Theta}$ be an algorithm that takes any data set and returns a model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ such that its training loss is lower than ϵ. That is, for any training data set $D \in 2^{\mathcal{X}\times\mathcal{Y}}$, $\mathcal{A}_\epsilon$ verifies that $\hat{L}(\mathcal{A}_\epsilon(D), D) \leq \epsilon$. This kind of algorithms are quite common in machine learning where the model space $\boldsymbol{\Theta}$ is big enough that there exists models capable of memorizing random labels (Zhang et al., 2017, Theorem 1). Then, for any $\mathcal{A}_\epsilon$ algorithm, with h.p. $1-\delta$ over $D \sim \nu^n$,

$$L(\mathcal{A}_\epsilon(D)) \leq \hat{L}(\mathcal{A}_\epsilon(D), D) + \mathcal{I}^{-1}_{\mathcal{A}_\epsilon(D)}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) \leq L(\mathcal{A}_\epsilon(D)) + \epsilon .$$

This provides an answer to the Open Question 1 using this newly presented bound:

> PAC-Chernoff bounds are perfectly tight for (over-parameterized) interpolators.

The question now is whether this distribution-dependent bound is useful to understand the generalization of algorithms retrieving (over-parameterized) interpolators. A first insight from the above bound is that the (inverse) rate function of the retrieved interpolator defines its generalization error. To the best of our knowledge, this is the first (distribution-dependent) complexity measure characterizing the generalization error an interpolator even in the context of a over-parameterized model class. In the next section, we explore further how the rate function can be used to precisely characterize which are the interpolates that better generalize.

4.2 Smoother Interpolators Generalize Better

As mentioned in the introduction, an open question in machine learning is the following,

Open Question 2

Given two models $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$, both successfully interpolating the training data, which of them generalizes better?

In this section, we introduce a formal result that provides the following answer to the above question: *the smoother interpolator is the one that achieves better generalization, given that it is sufficiently smoother*. The condition of being *smoother* is defined in terms of the rate function of the models. We contend that a model $\boldsymbol{\theta}$ is *smoother* than a model $\boldsymbol{\theta}'$ if, for any $a > 0$, observing a deviation larger than a between the expected (test) loss and the empirical (train) loss is consistently smaller for $\boldsymbol{\theta}$ than for $\boldsymbol{\theta}'$. This is formalized as follows,

Definition 3. *Given a data-generating distribution ν and a loss function ℓ, a model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is β-smoother than a model $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$ if*

$$\forall a \in (0, \beta] \quad \mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a)\,.$$

Notice that if $\boldsymbol{\theta}$ is β-smoother than $\boldsymbol{\theta}'$, it is β'-smoother for any $\beta' \in (0, \beta]$. Furthermore, if $\boldsymbol{\theta}$ is β-smoother than $\boldsymbol{\theta}'$, it verifies that $\boldsymbol{\theta}'$ cannot be β'-smoother than $\boldsymbol{\theta}$ for any $\beta' > 0$. This properties allow to reasonably compare degrees of smoothness between models in the same or different model spaces. It is important to notice that the concept of smoothness is defined in the context a given data-generating distribution and given loss function.

According to the above definition, the higher the rate function, the smoother the model when comparing it to others. Furthermore, according to Chernoff's bound (see Theorem 3.3 and Equation (2)), smoother models will have an empirical loss $\hat{L}(D, \boldsymbol{\theta})$ more concentrated around its expected value $L(\boldsymbol{\theta})$. In consequence, it is more unlikely to observe higher differences between $\hat{L}(D, \boldsymbol{\theta})$ and $L(\boldsymbol{\theta})$ in smoother models. In fact, the following result states that there is a correspondence between the variance of the model's loss and the newly introduced notion of smoothness.

Proposition 4.5 (Proof). *For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$,*

$$\mathbb{V}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})) \leq \mathbb{V}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}')) \iff \exists \beta > 0 \text{ s.t. } \boldsymbol{\theta} \text{ is } \beta\text{-smoother than } \boldsymbol{\theta}'\,.$$

This result shows the relation between the smoothness and the variance the loss function between a pair of models $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. As a result, we can *intuitively* understand this notion of smoothness as a *generalization of the variance of the loss of a model.*

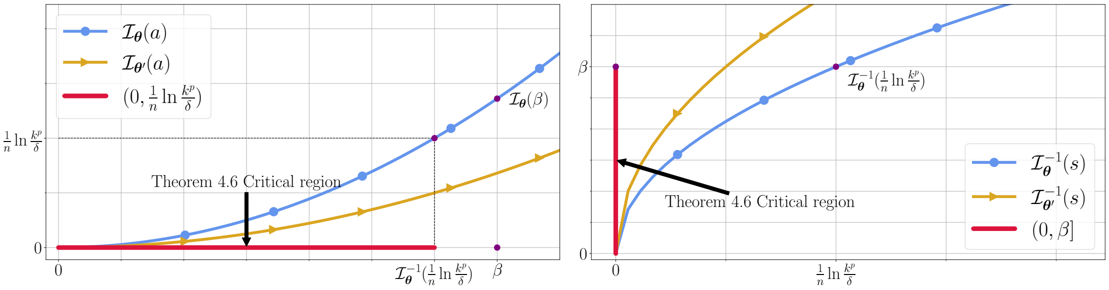

Figure 4: Visual illustration on Theorem 4.6. The rate and inverse rate function of two models $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \boldsymbol{\Theta}$ is shown with $\boldsymbol{\theta}$ being β-smoother than $\boldsymbol{\theta}'$ with $\beta > \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$. This highlights the critical region that must fall below β for the theorem to hold.

Using this definition of smoothness and the results presented in Section 4.1, we can easily derive a result providing an answer to which interpolator generalizes better,

Theorem 4.6 (Proof)**.** *For any $\epsilon \geq 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^p$ and $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$, simultaneously,*

$$\textit{if } \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \textit{ and } \boldsymbol{\theta} \textit{ is } \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\textit{-smoother than } \boldsymbol{\theta}'\textit{, then, } L(\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}') + \epsilon\,.$$

Note that, due to Proposition 4.5, the smoothness condition given by "$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$-smoother" in Theorem 4.6 can be understood as an inequality between the variances of the loss function under the hypothesis that n is sufficiently large. In fact, if $\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) \leq \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}'))$, there exists $\beta > 0$ such that $\boldsymbol{\theta}$ is β-smoother than $\boldsymbol{\theta}'$. Then, if n is large enough to verify that $\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) \leq \beta$, we got that with h.p., if $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$, then, $L(\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}') + \epsilon$. In short, Theorem 4.6 is a generalization of the following result: if n is large enough, with h.p.,

$$\text{if } \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \text{ and } \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) \leq \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}')), \text{ then, } L(\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}') + \epsilon\,.$$

Theorem 4.6 states that an interpolator generalizes better than another (with h.p.) if it is *sufficiently smoother* in terms of its rate function. Figure 4 illustrates the premise of this theorem. We should note that the above result holds even for the *log-loss*, which is the *default* loss used for *training*, and for over-parameterized model classes. This result is specially useful when ϵ is very small or null, as it states, with h.p., that *smooth interpolators* generalize better, up to an ϵ, than other less smooth models; independently on whether these interpolate the data or not. Furthermore, the above result verifies that the higher the probability $1-\delta$ or the number of parameters p, the stronger the smoothness condition needs to be; and the opposite for larger n. In that sense, we could have interpolators which are β-smoother than others but have worse generalization performance, because they are not smooth enough in order to apply Theorem 4.6.

Interpolators with a larger rate function $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ or, equivalently, smoother interpolators, are the ones that better generalize.

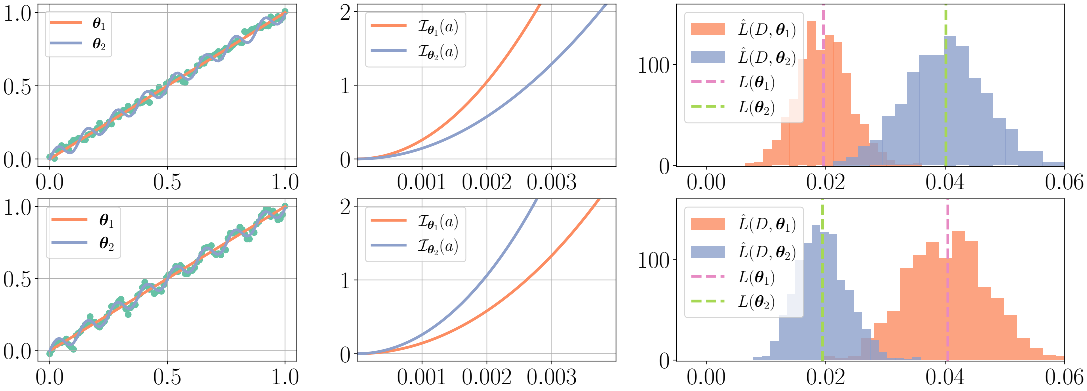

Figure 5: Visualization on the presented notion of smoothness. Two models, a linear model $\boldsymbol{\theta}_1$ and a complex model $\boldsymbol{\theta}_2$, are evaluated on two data-generating distributions ν_1 (first row) and ν_2 (second row). The first column shows the data and models, the second shows rate functions, and the third displays $\hat{L}(D, \boldsymbol{\theta})$ distributions.

Figure 5 shows a synthetical example to highlight how the notion of smoothness depends on the *specific data-generating distribution*, and why one model is *smoother* than another only relative to such distribution. The example considers two different data generating distributions ν_1 (first row, which adds random noise to a linear function) and ν_2 (second row, adds random noise to a complex sinusoidal function). The considered loss is the mean squared error. The introduced notion of smoothness using the rate function shows how $\boldsymbol{\theta}_1$ (a linear model) is *smoother* than $\boldsymbol{\theta}_2$ (a more complex model) under ν_1. In fact, under ν_1, the distribution of $\hat{L}(D, \boldsymbol{\theta}_1)$ is more concentrated around a smaller mean value. However, the second row of this example also shows how the more complex model $\boldsymbol{\theta}_2$ can be smoother than a linear model $\boldsymbol{\theta}_1$ under a different data-generating distribution ν_2 for exactly the same reasons. With this example we want to highlight that a good notion of smoothness (as the one presented here) should consider the data in which the models are being evaluated, rather than just the complexity of the function they induced.

The results from this section clearly indicate that the generalization error of a interpolator is defined by its level of smoothness and its rate function. The question is whether this theoretical characterizations, which relies on distribution-dependent quantities, is useful for understanding the inner workings of current learning techniques and complex phenomenons appearing in deep learning. In Section 5, we show how the PAC-Chernoff bound and our smoothness criteria are powerful enough to analyze the so-called double-descent phenomenon (Belkin et al., 2019). In Section 6, we examine the relationship between the smoothness of a model and widely used regularization techniques such as the parameter norm of a model, distance from initialization, input-gradient norm, and Lipschitz constant. Many of these quantities have also been previously used as measures of smoothness or complexity of a model (Neyshabur et al., 2017a). In Section 7, we show why invariant architectures and data-augmentation induce smoother interpolators. Finally, in Section 8, we examine how over-parametrization is a necessary condition for having smoother interpolators. In Section 10, we also elaborate on the need to stablish connections our definition of smoothness with current existing definitions (Keskar et al., 2017b; Shalev-Shwartz & Ben-David, 2014).

Size	Train Acc.	Test Acc.	Test NLL	Bound
4k	43.82%	42.74%	1.56	2.93
32k	70.84%	59.86%	1.17	2.01
85k	86.69%	62.73%	1.35	1.77
163k	97.65%	63.97%	1.82	1.94
266k	99.84%	65.19%	2.15	2.19
395k	100.0%	65.74%	2.28	2.30
548k	100.0%	67.27%	2.10	2.11
727k	100.0%	69.88%	1.84	1.86
931k	100.0%	69.11%	1.83	1.84
1161k	100.0%	69.82%	1.72	1.74

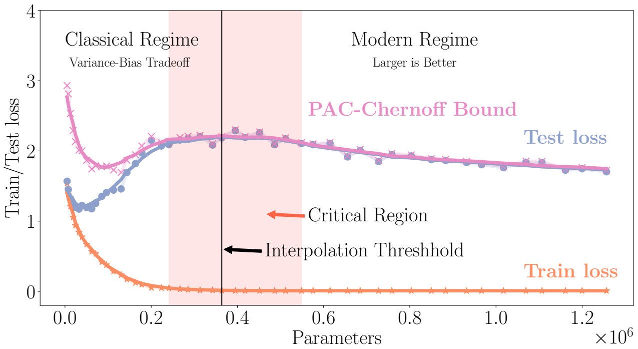

Figure 6: Illustration of the *double-descent phenomenon*: Evolution of the train loss, test loss and the proposed PAC-Chernoff Bound of nested convolutional neural networks with increasing number of parameters. Increasing the number of parameters promotes models with worse population loss until the critical region; further increasing the number of parameters decreases the population error. The proposed bound is perfectly tight for interpolators. The point where models reach the interpolation regime is marked with a black line. The training and test loss of models are marked with points and their smoothed curve is shown. Only a few of the total models are shown in the table.

5. Understanding Double-Descent with PAC-Chernoff Bounds

The existing literature consistently demonstrates that interpolators with a larger number of parameters tend to perform better. A finding that defies traditional beliefs from classical statistical learning theory. Traditionally, it was assumed that increasing the number of parameters in a model would lead to higher overfitting and, consequently, poor generalization performance. However, this perspective has been challenged by the phenomenon of the double-descent curve (Belkin et al., 2019), which exemplifies a shift in dynamics as models enter the interpolation regime. In this regime, an increase in parameters paradoxically leads to improved performance. This counter-intuitive behavior suggests that once models begin to interpolate, their generalization performance improves as they grow larger.

Figure 6 illustrates this phenomenon by showing the training and the test loss of a sequence of convolutional networks with a growing number of parameters (implemented by adding more channels to the layers of a simple convolutional model). More precisely, models with a number of parameters ranging from 4k to 1161k[1].

Figure 6 also displays the evolution of our PAC-Chernoff bound for the whole range of models. As theoretically shown, this bound is perfectly tight for interpolators. In the classical regime, the bound also presents the so-called *double descent phenomena*, even thought its tightness is not guaranteed. This highlights how distribution-dependent bounds are powerful enough to capture the complex dynamics emerging in the interpolation regime.

[1] All models were found by running stochastic gradient descent on Cifar10's training data, until training loss reaches 0.01 or until it did not improve in two consecutive epochs of training.

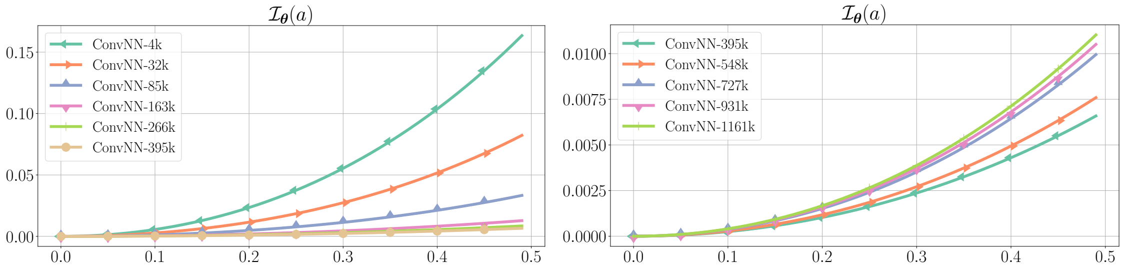

Figure 7: Rate functions obtained by interpolating CNNs with 4k-1161k parameters on Cifar10. Models correspond to those in the table of Figure 6. Parameters are increased by increasing the number of channels in the architecture, resulting in *nested* model classes. Two images are shown where models are divided so that the rate function decrease on the left figure and increase on right one.

Open Question 3

Why does the generalization performance of interpolators improve with an increasing number of parameters?

The complexity term of our own PAC-Chernoff bound $\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{kp}{\delta}\right)$ monotonically increases with the size of the mode class, which, in principle, would contradict the *double-descent phenomena* on the bound and the fact that generalization error is reduced as we increase the number of models. The following result, viewed as an extension of Theorem 4.6, offers a partial explanation for this puzzling phenomena. More precisely, it demonstrates that if the interpolators become progressively *smoother* (higher rate function), their generalization error will be reduced, despite being part of a model class characterized by a greater number of parameters.

Theorem 5.1 (Proof). *Let $\boldsymbol{\Theta} \subset \boldsymbol{\Theta}'$ be two nested model classes with $p < p'$ parameters respectively. For any $\epsilon > 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, for any $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously*

$$\text{if } \hat{L}(D,\boldsymbol{\theta}') \leq \epsilon\,,\; \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \;\text{ and } \boldsymbol{\theta}' \text{ is } \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\big(\tfrac{1}{n}\ln\tfrac{kp'}{\delta}\big)\text{-smoother than } \boldsymbol{\theta}$$
$$\Downarrow$$
$$L(\boldsymbol{\theta}') \leq L(\boldsymbol{\theta}) + \epsilon\,.$$

Figure 7 (right) illustrates the rate functions of the interpolators from Figure 6, highlighting how *larger interpolators* become increasingly smoother, meaning their rate functions are progressively higher (i.e., they are progressively smoother). This explains how it is possible to obtain larger interpolators with smaller generalization error. In fact, by reversing the implication of Theorem 5.1, we can infer that if a larger interpolator exhibits better generalization performance, then, it cannot be *less smooth* than a smaller interpolator.

Corollary 5.2 (Proof). *Let $\mathbf{\Theta} \subset \mathbf{\Theta}'$ be two nested model classes with $p < p'$ parameters respectively. For any $\epsilon > 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, for any $\boldsymbol{\theta}' \in \mathbf{\Theta}'$, $\boldsymbol{\theta} \in \mathbf{\Theta}$, simultaneously,*

$$\hat{L}(D,\boldsymbol{\theta}') \leq \epsilon,\ \ \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \text{ and } L(\boldsymbol{\theta}') + \epsilon < L(\boldsymbol{\theta})$$
$$\Downarrow$$
$$\boldsymbol{\theta} \text{ is not } \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp'}{\delta}\right)\text{-smoother than } \boldsymbol{\theta}'\,.$$

Consequently, Open Question 3 can be reformulated in terms of model smoothness rather than generalization performance.

Open Question 4

Why does the smoothness of interpolators improve with an increasing number of parameters?

Based on Theorem 5.1 and Corollary 5.2, addressing Open Question 4 is equivalent to answering Open Question 3. In other words, if we could answer why the rate function of the interpolators is increasingly higher as we increase the number of parameters in nested model classes, Theorem 5.1 will effectively explain why the double descent phenomena, and in particular, why the generalization performance of interpolators enhances as the number of parameters grows. However, to the best of our knowledge, there is no direct answer to this behavior without incurring in additional hypothesis on the matter.

Our *rationale* for explaining this phenomenon posits that, by increasing the number of parameters, larger neural networks are better at capturing invariances in the data, a notion that has been supported both theoretically (Bronstein et al., 2021) and empirically (Goodfellow et al., 2009) by the machine learning community. Additionally, as will be shown in Section 7, this capability of capturing invariances leads to *smoother* models with higher rate functions. Combining these two factors, Theorem 5.1 can help to elucidate why the generalization performance of interpolators improves with an increase in parameters. The experimental setup presented in Figures 6 and 7 support this understanding.

Finally, it is also quite interesting to look at the models on the *left part* of the *interpolation threshold* of Figure 6. Surprisingly, the sequence of rate functions, displayed in Figure 7 (left), get increasingly lower as the size of the mode class increases, just the opposite of what happens in Figure 7 (right). In the classical regime, increasing the number of parameters leads to less smooth models with higher generalization error. However, explaining this behavior in the non-interpolation regime is out of the scope of this work, as the presented PAC-Chernoff bound is *not tight* in that regime and no theoretical guarantees can be derived in such framework. However, it is worth noticing that, even with the lack of tightness guarantees, the proposed bound is *close enough* to showcase the double descent phenomena itself.

In the following section, we will connect the proposed oracle bound with many existing regularization techniques, allowing to explain, in an unified framework, the effectiveness in terms of generalization.

6. Explicit Regularization

Structural risk minimization (Shawe-Taylor et al., 1998) is a learning principle based on *regularizers* that penalize the complexity of a model when minimizing the training loss. Let $r(\boldsymbol{\theta})$ denote a regularizing function, the learning objective is then given by,

$$\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \hat{L}(D,\boldsymbol{\theta}) + r(\boldsymbol{\theta})\,. \tag{4}$$

High-probability bounds of the form given in Equation (1), $L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{C}(\boldsymbol{\theta}, D, \delta)$, have been traditionally used to justify the use of regularizers and to derive novel ones. According to these bounds, an obvious choice for regularizes is the complexity term, $r(\boldsymbol{\theta}) = \mathcal{C}(\boldsymbol{\theta}, D, \delta)$. Then, the structural risk minimization problem given in Equation (4) would correspond to minimize a high-probability upper bound over the expected loss $L(\boldsymbol{\theta})$. Many regularizing methods have been studied from this perspective. One of the most commons one is being the ℓ^2-norm regularization and other variations based on spectral norms, which recurrently show up within the complexity term of many generalization bounds (Bartlett et al., 2017). However, as discussed in the introduction, these bounds are known to be loose in the context of over-parameterized model classes interpolating the data. In consequence, the rationale of using their associated complexity measures $\mathcal{C}(\boldsymbol{\theta}, D, \delta)$ as regularizes is no longer as strong as it used to be in learning setups where the model class is not over-parameterized.

Open Question 5

Is there a regularizer $r(\boldsymbol{\theta})$ that ensures near-optimal performance for over-parameterized model classes interpolating the training data?

The following result provides an answer to Open Question 5. It shows that the complexity measure of the PAC-Chernoff bound given in Theorem 4.1 can be used as a regularizer. That is to say, using the inverse rate function $\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{kp}{\delta}\right)$, we get an *optimal regularizer for over-parameterized interpolators.* More precisely, for any $\epsilon > 0$, let $\boldsymbol{\theta}^\star_\epsilon$ and $\boldsymbol{\theta}^\times_\epsilon$ denote the interpolator with the best generalization performance and the interpolator with the smallest inverse rate function, respectively,

$$\boldsymbol{\theta}^\star_\epsilon = \mathop{\arg\min}_{\boldsymbol{\theta}\,:\,\hat{L}(D,\boldsymbol{\theta})\leq\epsilon} L(\boldsymbol{\theta})\,, \qquad \boldsymbol{\theta}^\times_\epsilon = \mathop{\arg\min}_{\boldsymbol{\theta}\,:\,\hat{L}(D,\boldsymbol{\theta})\leq\epsilon} \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{kp}{\delta}\right)\,.$$

This result shows that, with h.p., the expected loss of the interpolator with the smallest inverse rate function $\boldsymbol{\theta}^\times_\epsilon$ is very close to the expected loss of the optimal interpolator, $\boldsymbol{\theta}^\star_\epsilon$.

Theorem 6.1 (Proof)**.** *For any $\epsilon > 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, $|L(\boldsymbol{\theta}^\star_\epsilon) - L(\boldsymbol{\theta}^\times_\epsilon)| \leq \epsilon$.*

In simpler terms, favoring models with a small inverse rate function is the same as favoring models with a larger rate function, which essentially means choosing *smoother models.* This finding highlights that the *smoothest interpolator* not only fits the data but also performs nearly optimally when it comes to generalization.

The inverse rate function $\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{kp}{\delta}\right)$ is an *optimal regularizer for over-parameterized interpolators.*

6.1 Connecting the Inverse Rate with Existing Regularization Techniques

We have shown that the inverse rate function, a distribution-dependent and oracle element, is an *optimal regularizer* for interpolators. However, in this section we will show how a wide range of existing regularizers are tightly connected to the inverse rate and, in consequence, to our definition of smoothness. We do not aim to fully explore these connections, and we do not claim that the new bounds that we derive are better than existing ones. Our focus here is to show how the (inverse) rate function of a model can be used to unify many existing regularization techniques which were previously assumed to be unrelated.

6.1.1 Norm ℓ_2 Regularization

This regularizer, also known as weight decay, is widely used by the ML community and it is known to improve generalization even in over-parameterized model classes interpolating the training data (Goodfellow et al., 2016). The following result defines a connection between the inverse rate of a model and its norm, under the widely used assumption that the loss function $\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is Lipschitz with respect to the parameters of the model $\boldsymbol{\theta}$ (Li & Orabona, 2019).

Proposition 6.2 (Proof)**.** *If the loss function $\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{\theta}$ with constant $M > 0$, then, for any $\boldsymbol{\theta}_0 \in \boldsymbol{\Theta}_0 = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid \mathbb{V}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})) = 0\}$, it verifies that*

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\right) \leq \sqrt{2Ma}\;\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2\,,$$

where $a = min\big(1, \frac{1}{n}\ln\frac{kp}{\delta}\big)$.

In many common ML models, the null vector is a model with null variance, that is, $\mathbf{0} \in \boldsymbol{\Theta}_0$, because the loss of this model is constant for any data sample. For example, in supervised classification problems with K labels, a neural network with null weights has a constant loss equal to $\ln K$. In these cases, the above result shows that by promoting models with small parameter norm we achieve models with small inverse rate. In consequence, the ℓ_2-norm is a proxy to minimize the inverse rate and works as a regularizer. Figure 2 illustrates for our running example how the use of ℓ_2-norm regularization leads to models with small parameter norm and larger rate function. This aligns with the existing evidence in the literature about the use of ℓ_2-norm for successfully regularizing interpolators.

However, in the context of over-parameterized models interpolating the training data, it is well known that the ℓ_2-norm of the parameters of the model does not correlate well with the generalization error (Jiang et al., 2020). That is easy to explain if we consider that the ℓ_2-norm just defines an upper-bound over the inverse rate, which, as shown in Section 4.1 is an optimal proxy for the generalization of interpolators. The gap of this upper-bound is the ultimate responsible of the failure of ℓ_2-norm as a proxy for the generalization error of an interpolator. In fact, an open question in ML is:

Open Question 6

When does the norm of an interpolator correlate with its generalization error?

The following result provides a characterization of a learning setup where these two elements are correlated.

Corollary 6.3 (Proof)**.** *Considering the log-loss, if it belongs to the exponential family with a constant base measure, that is, there exist $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^p$, $a : \boldsymbol{\Theta} \to \mathbb{R}$ and $k \in \mathbb{R}$ such that*

$$\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) := -\ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T s(\boldsymbol{x}, \boldsymbol{y}) - a(\boldsymbol{\theta}) + k \quad \forall (\boldsymbol{x}, \boldsymbol{y}) \sim \nu\,;$$

Then, $\forall \epsilon > 0$, exists $n_0 > 0$ such that $\forall n > n_0$:

$$\left| \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\right) - \sqrt{2\tfrac{1}{n}\ln\tfrac{kp}{\delta}}\sqrt{\boldsymbol{\theta}^T Cov_\nu(s(\boldsymbol{y}, \boldsymbol{x}))\boldsymbol{\theta}} \right| \leq \epsilon\,,$$

where $Cov_\nu(\cdot)$ is the covariance w.r.t. ν of the sufficient statistics of each $(\boldsymbol{x}, \boldsymbol{y})$ sample.

This result shows that under the exponential family and a large enough number of data samples, the inverse rate is close to the scaled Riemannian norm of the model parameters under the positive definite matrix $\text{Cov}_\nu(s(\boldsymbol{y}, \boldsymbol{x}))$ that defines the Riemannian metric. Note that, in this case, the Riemannian norm will be an optimal regularizer as Theorem 6.1 applies and the inverse rate is a perfectly tight proxy for the generalization error of interpolators as shown in Section 4.1. The ℓ_2-norm is obtained when the covariance matrix is a diagonal matrix with constant entries, a condition that requires the statistical independence of the components of the sufficient statistics $s(\boldsymbol{y}, \boldsymbol{x})$ with respect to the data-generating distribution.

In short, Proposition 6.2 shows how the ℓ_2-norm of the parameters of the model upper bounds its inverse rate and, then, it shows why biasing the optimizer towards models with smaller ℓ_2-norm tends to reduce the generalization error of the interpolators. However, Corollary 6.3 exemplifies a general setting to understand the gap in the inequality of Proposition 6.2 and how this gap can manifest if $\text{Cov}_\nu(s(\boldsymbol{y}, \boldsymbol{x}))$, a distribution-dependent quantity, is very different from a diagonal matrix.

6.1.2 Distance From Initialization

The distance from initialization is known to be related to the generalization error of an interpolator (Nagarajan & Kolter, 2017). In fact, it has been successfully used as a regularizer in Hu et al. (2020). In these works, an implicit assumption was that the initial models are produced through one of the well-established initialization schemes used in neural networks, such as Xavier or He initialization (Goodfellow et al., 2016). These parameter initialization schemes provide an initial set of weights which guarantee that the variance of the activations across the different layers remain constant. In consequence, the randomly initialized model makes almost constant predictions. If $\boldsymbol{\theta}_0$ denotes the initial parameter, we should see that $\mathbb{V}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}_0))$ is very small. Figure 3 (right) shows how this is the case for a randomly intialized Inception model using He initialization.

Under the assumption that the initial model obtained by common initializers belongs to $\boldsymbol{\Theta}_0$, Proposition 6.2 also shows a link between distance from initilizaton and the inverse rate function and, in consequence, with the generalization error of a model. The same reasoning used with ℓ_2-norm applies here: by promoting models with small distance from initialization we promote models with small generalization error.

Furthermore, as happens with the ℓ_2-norm, in the context of over-parameterized interpolators, distance from intialization is not a reliable proxy for the generalization error (Jiang et al., 2020). Again, this is easy to explain if we consider that distance from intialization is just an upper-bound over of the inverse rate. In consequence, an open question in ML is:

Open Question 7

When the distance from intialization of an interpolator perfectly correlates with its generalization error?

We provide again a characterization for the case where the cumulant generating function of the model $J_{\boldsymbol{\theta}}(\lambda)$ is a quadratic function or, equivalently, its second-order Taylor approximation around $\boldsymbol{\theta}_0 \in \boldsymbol{\Theta}_0$ is exact. In this case, as we show in the proof of Proposition 6.4, the cumulant generating function can be expressed as

$$J_{\boldsymbol{\theta}}(\lambda) = \frac{1}{2}\lambda^2 \big(\boldsymbol{\theta} - \boldsymbol{\theta}_0\big)^T \mathrm{Cov}_\nu\big(\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_0)\big)\big(\boldsymbol{\theta} - \boldsymbol{\theta}_0\big)\,, \tag{5}$$

where $\mathrm{Cov}_\nu(\cdot)$ is the covariance w.r.t. ν of the gradient of the log-likelihood of each $(\boldsymbol{x}, \boldsymbol{y})$ sample. Under this hypothesis, the inverse rate can be expressed as stated next.

Proposition 6.4 (Proof)**.** *For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, if the equality of Equation (5) holds, then*

$$\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) = \sqrt{2\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\sqrt{\big(\boldsymbol{\theta} - \boldsymbol{\theta}_0\big)^T \mathit{Cov}_\nu\big(\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_0)\big)\big(\boldsymbol{\theta} - \boldsymbol{\theta}_0\big)}\,.$$

When training highly over-parameterized models using gradient descent, it is well known that parameters remain close to their initial values. This phenomenon is called lazy training and is a standard assumption in many theoretical analysis of deep neural networks (Jacot et al., 2018). However, when lazy training takes place, the second-order Taylor approximation of the cumulant could become highly accurate. The above proposition then suggests how the Riemannian distance, induced by $\mathrm{Cov}_\nu\big(\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_0)\big)$, would be an optimal proxy for the generalization error of an interpolator in the lazy training regime. This reasoning also helps to understand the limitations in this regard of the Euclidean-distance.

6.1.3 Input-Gradient and Lipschitz Regularization

Input-gradient regularization is a technique used in deep learning to improve the robustness of neural network models (Gouk et al., 2021). This approach focuses on the gradients of the model's output with respect to its input, which are indicative of how sensitive the model's predictions are to changes in the input data.

The following result shows how the inverse rate function and the expected norm of the input-gradient of a model are also related. This result is based on the *log-sobolev inequalities* (Chafaï, 2004) and relies on the assumption that $\nu(\boldsymbol{y}|\boldsymbol{x})$ is deterministic, to simplify gradients with respect to the target variable, and that $\nu(\boldsymbol{x}, \boldsymbol{y})$ is a uniformly strictly log-concave density, an assumption satisfied by a wide range of distributions.

Proposition 6.5 (Proof)**.** *If $\nu(\boldsymbol{x}, \boldsymbol{y})$ is an uniformly strictly log-concave density and $\nu(\boldsymbol{y}|\boldsymbol{x})$ is deterministic, then $\exists M > 0$, such that, for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

$$\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \sqrt{\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\sqrt{M\mathbb{E}_\nu\Big[\big\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\big\|_2^2\Big]}\,.$$

The above result explains why models with small input-gradient norm tend to have a small generalization error. And why a small generalization error may coexist with a large input-gradient norm, if the gap in the above inequality is large.

From the above result, it is straightforward to derive a connection between the generalization of a model and its Lipschitz constant. If a model $\boldsymbol{\theta}$ is Lipschitz with respect to the input data $\boldsymbol{x}$ with Lipschitz constant denoted $\mathrm{Lip}(\boldsymbol{\theta})$, then, by definition, the following inequality is verified:

$$\forall(\boldsymbol{x},\boldsymbol{y}) \in \mathrm{supp}(\nu) \quad \|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\|_2^2 \leq \mathrm{Lip}(\boldsymbol{\theta})\,. \tag{6}$$

The above inequality can be combined with Proposition 6.5 to derive a new upper bound on the inverse of the rate function and on the generalization error of a model:

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \sqrt{\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\sqrt{M\mathrm{Lip}(\boldsymbol{\theta})}\,.$$

In consequence, models with a small Lipschitz constant, will tend to have a small generalization error. This aligns fully with the existing literature (Neyshabur et al., 2017b) and, also, explains the rationale behind regularization techniques that prevent overfitting by controlling the Lipschitz constant (Gouk et al., 2021).

6.1.4 Summary

In this section, we have theoretically characterized an optimal (distribution-dependent) regularizer for interpolators, even for over-parameterized model classes. This theoretical characterization can be used to understand why a wide range of commonly used regularization techniques promotes interpolators with a smaller generalization error. We are not aware of previous works able to characterize optimal regularizers for interpolators and establish such a wide range of connections with existing regularization methods in the machine learning community.

We have also analyzed, in some specific but not less relevant situations, when could some of these common regularization methods be optimal for interpolators; even in over-parameterized model classes. In this regard, our work complements other related theoretical results, like the ones proposed by Bartlett et al. (2020), where the Euclidean norm was shown to be an optimal regularizer for linear regression interpolators under some specific assumptions.

This section allows to draw connections between our definition of smoothness, as outlined in Section 4, and other existing definitions in the literature. We have shown how the *smoothest interpolator* has near-optimal performance in Proposition 6.1. The clearest connection with existing measures of smoothness arises from the characterization of the Lipschitz constant, as demonstrated by combining Equation (6) and Proposition 6.5. The Lipschitz constant is frequently cited in the literature as a measure of a model's *smoothness* (Gouk et al., 2021). In this work, we have shown how models with a small Lipschitz constant have a small inverse rate and, in consequence, are smoother according to our own definition.

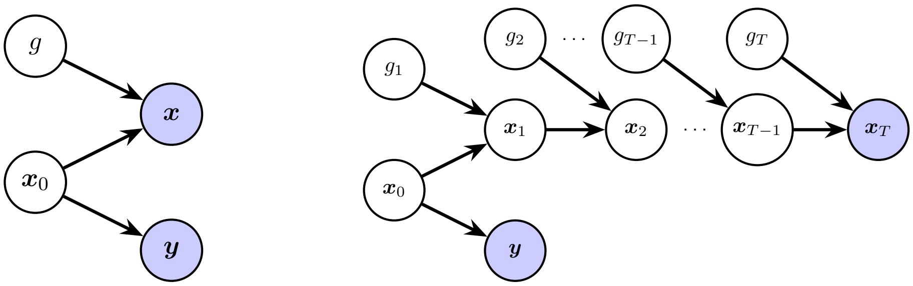

Figure 8: Graph representation of Assumption 2. The left side represents the original assumption with a single transformation and the right side represents the *generalized setting* with T transformations. Observed variables are marked with a blue-ish background.

7. Invariances

In many machine learning problems, we know that the input data $\boldsymbol{x}$ we observed have been *transformed*, usually as a result of the *measuring process*. For example, in the case of sensor readings, by random noise due to non-perfect sensors or, in the case of images, these may be rotated, blurred, etc, by the image capturing process (Bronstein et al., 2021). In this section, we discuss how widely used techniques in machine learning, like data augmentation (Shorten & Khoshgoftaar, 2019) and invariant architectures (Bronstein et al., 2021), can be jointly explained, using PAC-Chernoff bounds and the rate function, as techniques to improve learning under these *transformed* inputs.

Let G denote a set of input-data transformations such that any $g \in G$ denotes a function, $g : \mathcal{X} \to \mathcal{X}$, defining a specific transformation. Assume there exists an *unknown distribution* h over G. In image classification, G could be the set of all possible rotations, translations and/or reflections.

The central hypothesis in this section is that the generative process of the data follows the framework defined in the previous paragraphs.

Assumption 2. *The data-generating distribution has the following structure:*

$$(i)\ \boldsymbol{x}_0 \sim \nu_0(\boldsymbol{x}_0), \quad (ii)\ y \sim \nu(\boldsymbol{y}|\boldsymbol{x}_0), \quad (iii)\ g \sim h(g), \quad (iv)\ \boldsymbol{x} = g(\boldsymbol{x}_0)\,,$$

where $\boldsymbol{x}_0$ denotes the un-transformed and un-observed input, which is distributed according to ν_0, $\boldsymbol{x}$ denotes the observed (transformed) input. See Figure 8 (left) for a graph representation.

Under this assumption, the target value of an input is sampled *before* its transformation, that is, *transforming the inputs does not affect its label.* In statistical terms, under Assumption 2, we have that the target random variable $\boldsymbol{y}$ is conditionally independent of $\boldsymbol{x}$ given $\boldsymbol{x}_0$. This fact perfectly aligns with many common settings, where, for example, rotating, reflecting or cropping an image does not alter the label of the object been represented in it (Bronstein et al., 2021). This assumption is weaker than the usual ones used in many related works (Chen et al., 2020; Bronstein et al., 2021), where the conditional generating distribution is assumed to be invariant under any transformation. These works assume that $\forall g \in G$, $\nu(\boldsymbol{y}|\boldsymbol{x}) = \nu(\boldsymbol{y}|g(\boldsymbol{x}))$. This assumption does not hold, for example, when considering rotation over digits, as the *true label* of the image of a *six* rotated 180 degrees *changes to a nine.*

Consider the generalized scenario where several transformations can be applied, that is, $g \in G$ is the composition of other transformations $g(\boldsymbol{x}_0) = g_T \circ \cdots \circ g_1(\boldsymbol{x}_0) = \boldsymbol{x}$. We may denote $G_1, \ldots, G_T$ the sets of all possible individual transformations, with $G = \cup_t G_t$. Notice that the value of T is arbitrary and can differ from one transformation $g \in G$ to another; however, to simplify the notation we will use the same T where one may consider that some $g_i \in G_i$ are the identity. Then, we may consider the non-fully-transformed inputs $\boldsymbol{x}_t = g_t \circ \cdots \circ g_1(\boldsymbol{x}_0)$, with $0 < t < T$. The following result shows that the mutual information between the targets and the inputs, denoted $MI(\boldsymbol{y}\,;\boldsymbol{x})$, decreases the further the input is transformed.

Proposition 7.1 (Proof)**.** *Under Assumption 2, let $\boldsymbol{x}_t$ denote the random variable denoting a transformation of an input $\boldsymbol{x}_0$ using a composition of t transformations, then*

$$MI(\boldsymbol{y}\,;\boldsymbol{x}_t) \leq MI(\boldsymbol{y}\,;\boldsymbol{x}_{t-1}) \quad \forall t \in \mathbb{N}^+\,.$$

As a consequence, $MI(\boldsymbol{y};\boldsymbol{x}) = MI(\boldsymbol{y};\boldsymbol{x}_T) \leq MI(\boldsymbol{y};\boldsymbol{x}_0)$.

Going back to the main objective of this work, which is to study the generalization of interpolators, the rate function and PAC-Chernoff bounds can be directly applied to explain why learning on transformed data $\boldsymbol{x}$ makes interpolators have worse generalization error compared to “ideally” learning on the untransformed data $\boldsymbol{x}_0$, which is never observed. In the following sections, we will use this argument to explain the reason why invariant architectures and data augmentation promote models with better generalization errors by exploiting the fact that the inputs are transformed.

Open Question 8

Why models interpolating transformed data, as described in Assumption 2, have a higher generalization error?

Given a data-generating distribution, it is clear that the probability distribution of $\hat{L}(D, \boldsymbol{\theta})$ rules the generalization error of the model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Proposition 7.1 shows that learning on transformed variables reduces the information that the inputs $\boldsymbol{x}$ have about their target $\boldsymbol{y}$. As a result, one may expect that under transformed inputs, $\hat{L}(D, \boldsymbol{\theta})$ becomes less concentrated and with a higher expectation. Figure 9 illustrates how this is the case for a Multi-Layer Perceptron and an Inception model on Cifar-10. In Figure 9 (left) we can see the distribution of $\hat{L}(D_0, \boldsymbol{\theta})$, $\hat{L}(D_1, \boldsymbol{\theta})$ and $\hat{L}(D_2, \boldsymbol{\theta})$ for datasets of size 50; where D_0 is the unstransformed data, D_1 corresponds to random translations applied to D_0, and D_2 corresponds to random rotations applied to D_1. In short, these datasets are drawn from ν_0, ν_1 and ν_2 respectively. The figure shows how transforming the inputs once (translations) and twice (translations + rotations) increases the expectation and the scattering of the loss. This *scattering* can be recognized by the size of the boxplots. As shown in Figure 9 (center) and (right), the lower level of concentration of the distribution of $\hat{L}(\cdot, \boldsymbol{\theta})$ under transformed inputs is also reflected by lower rate functions. Note that the same model can have different rate functions when applied over different data-generating distributions because the rate function depends on both the model and the data-generating distribution. Figure 9 also shows that the effect of

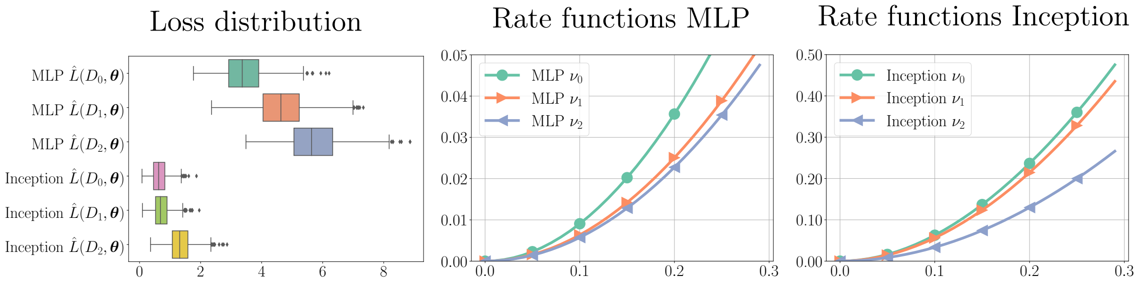

Figure 9: Illustration on the effect of transformed inputs. The distribution of $\hat{L}(D,\boldsymbol{\theta})$ is shown on a fixed Multilayer perceptron and a fixed Inception model with similar amount of parameters for three different learning setups. For *MLP* and *Inception*, $D_0 \sim \nu_0^{50}$ using Cifar10's test set; transformed inputs using random translations of 3 pixels, ($D_1 \sim \nu_1^{50}$); and, transformed inputs with additional rotations up-to 20° are done ($D_2 \sim \nu_2^{50}$).

using transformed inputs can also be *partly nullified* by using invariant architectures, as the case for Inception network which uses convolutional layers which are invariant to translations, ν_1, but not to rotations, ν_2. We will go deeper into this in the next subsection.

The observed changes in the distribution of the empirical loss $\hat{L}(\cdot,\boldsymbol{\theta})$ under transformed inputs can be formalized in our framework by considering some specific assumptions. Although these assumptions can not be easily verified in general, the following results intend to provide *theoretical justifications* for why transformed inputs make the distribution of the empirical loss $\hat{L}(\cdot,\boldsymbol{\theta})$ less concentrated and with higher expectation. Formally, let ν_t denote the marginal generating distribution of t-times transformed inputs $\boldsymbol{x}_t = g_t \circ \cdots \circ g_1(\boldsymbol{x}_0)$, $L^{\nu_t}(\boldsymbol{\theta})$ and $L^{\nu_{t+1}}(\boldsymbol{\theta})$ denotes the expected loss of the model $\boldsymbol{\theta}$ under the data-generating distributions ν_t and ν_{t+1}, respectively. Similarly, $\mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}(a)$ and $\mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a)$ denote the rate function of a model $\boldsymbol{\theta}$ under these two data-generating distributions. According to the following result, if the loss function is convex in $\boldsymbol{x}$ and the set of transformations satisfies that $\mathbb{E}_{g\sim h}[g(\boldsymbol{x})] = \boldsymbol{x}$, then, transformed inputs, as described by Assumption 2, make models have a higher expected loss.

Proposition 7.2 (Proof)**.** *Under Assumption 2, if $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is convex under $\boldsymbol{x}$ and $\mathbb{E}_{g_t}[g_t(\boldsymbol{x})] = \boldsymbol{x}$, then, $\forall\boldsymbol{\theta}\in\boldsymbol{\Theta},\ L^{\nu_{t+1}}(\boldsymbol{\theta}) \geq L^{\nu_t}(\boldsymbol{\theta})$.*

The following result helps to explain why the empirical loss is less concentrated when the input-data is transformed as described in Assumption 2. For any given input-data point $\boldsymbol{x}$, a given model incurs in some loss $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$, and when the same input-data point $\boldsymbol{x}$ is transformed by a randomly chosen transformation $g\sim h$, the model incurs in a different loss, $\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})$. This loss will be higher or lower than the loss over the untransformed input. The following result shows that the empirical loss under the transformed data is less concentrated if the relative change between the two losses, denoted by $\Delta(\boldsymbol{y},\boldsymbol{x},g,\boldsymbol{\theta}) = \ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta}) - \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$, is statistically independent of the loss itself, $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$. That is to say, how much the loss of the transformed-input changes does not depend of the loss of the unstransformed input itself.

Proposition 7.3 (Proof)**.** *Under Assumption 2, if the change observed in the loss after a transformation $\Delta(\boldsymbol{y},\boldsymbol{x},g,\boldsymbol{\theta}) = \ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta}) - \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is statistically independent of the loss itself, $\Delta(\boldsymbol{y},\boldsymbol{x},g,\boldsymbol{\theta}) \perp \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$, then*

$$\mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a) \leq \mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}(a) \quad \forall a > 0\,.$$

Summarizing, when input-data is transformed (Assumption 2), the input-data provides less information about the target variable (Proposition 7.1). This causes the distribution of the empirical loss $\hat{L}(\cdot,\boldsymbol{\theta})$ to be less concentrated and with higher expected loss $L(\boldsymbol{\theta})$ (as empirically shown in Figure 9 and theoretically argued in Propositions 7.2 and 7.3).

> Transformed input-data makes the expected loss of the model $L(\boldsymbol{\theta})$ higher and the distribution of $\hat{L}(\cdot,\boldsymbol{\theta})$ less concentrated.

Finally, we re-state here the distribution-dependent bound given in Theorem 4.1 using the extended notation used in this section. With high probability over random draws $D_t \sim \nu_t^n$, simultaneously for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, then:

$$\text{if } \hat{L}(D_t,\boldsymbol{\theta}) \leq \epsilon \ \text{ then } \quad \left(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}\right)^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\right) \leq L^{\nu_t}(\boldsymbol{\theta}) \leq \left(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}\right)^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\right) + \epsilon\,. \tag{7}$$

The above bound easily explains why the presence of transformed input-data makes interpolators have larger generalization error. The presence of transformed input-data reduces the level of concentration of $\hat{L}(D_t,\boldsymbol{\theta})$ and, in consequence, makes the rate function smaller (as shown in Figure 9 and Proposition 7.3). Thus, the inverse rate function increases and so does $(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_t})^{-1}(\frac{1}{n}\ln\frac{kp}{\delta})$; as a result, given that ϵ is supposedly very small, the generalization error of the interpolator increases too.

> The PAC-Chernoff bound explains why interpolators of transformed inputs have higher generalization error.

7.1 Invariant Architectures

Invariant architectures refers to a kind of neural networks that remain unaffected by certain transformations in the input data (Bronstein et al., 2021). In case of images, these transformations might include scaling, rotation, translation, or other geometrical changes.

Definition 4. *Given a set of transformations G, a model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is called G-invariant if*

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = p(\boldsymbol{y}|g(\boldsymbol{x}),\boldsymbol{\theta}) \quad \forall g \in G \ \ \forall (\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X}\times\mathcal{Y}\,.$$

These invariant models, such as convolutional neural networks (CNNs) (LeCun et al., 1998), are crucial because they can recognize and process patterns irrespective of their position, scale, or orientation. This invariance is particularly important in tasks like image and speech recognition, where the essence of the data needs to be captured regardless of such variations. Invariant architectures have consistently shown superior performance compared to non-invariant models (Goodfellow et al., 2016).

The generalization error of invariant neural networks has been widely studied before. Different generalization bounds have been shown to have complexity terms which tend to shrink when the architecture of the model is invariant (Sokolic et al., 2017; Elesedy, 2022; Behboodi et al., 2022). Although these bounds provide theoretical grounds on why generalization error is reduced with invariant architectures, the problem lies in that they only show that the complexity measures of the respective bounds are reduced, not the generalization error. However, based on the arguments given in Section 1, there is a growing empirical (Zhang et al., 2017) and theoretical evidence (Nagarajan & Kolter, 2019b; Gastpar et al., 2024; Wang et al., 2024) that many of these bounds are non-tight, or, probably, even vacuous, and do not satisfy basic criteria such as the bound decreasing with larger dataset sizes, as specifically happens, for example, with the bound provided by Behboodi et al. (2022). In consequence, to the best of our knowledge, there are not widely accepted, specific generalization bounds that comprehensively captures the effect of the invariance of an architecture on a interpolator's generalization error, specially in over-parameterized model classes. An open question is then the following,

Open Question 9

Why do (over-parameterized) invariant interpolators generalize better?

Although the final generalization error of a model is also influenced by other factors such as the optimization process, we can provide a partial answer to this relevant open question by analyzing the rate function of invariant models. The key advantage of invariant architectures under transformed inputs (Assumption 2) is that they are *unaffected* by input-data transformations and, in consequence, they avoid all the problems described in the previous subsection. This can be formally stated in the following result,

Proposition 7.4 (Proof). *Under Assumption 2, if a model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is G_t-invariant then,*

$$L^{\nu_{t+1}}(\boldsymbol{\theta}) = L^{\nu_t}(\boldsymbol{\theta}) \quad \textit{and} \quad \mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a) = \mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}(a) \quad \forall a > 0\,.$$

As argued in the above section, transformed-input data makes the empirical loss $\hat{L}(\cdot, \boldsymbol{\theta})$ be less concentrated (with a lower rate function) and with a higher expected loss; however, the previous result shows that invariant architectures allow to bypass this problem, as the distribution of the empirical loss will be unaffected by transformed-input data.

Figure 9 clearly illustrates this point. Recall that $D_0 \sim \nu_0^{50}$ defines the distribution of the input data, and $D_1 \sim \nu_1^{50}$ the distribution after applying random translations over D_0. As expected, for a fixed MLP model, whose architecture is non invariant to translations, the expected loss significantly increases and the rate function significantly decreases. While for a fixed *Inception* model, the changes are minimal. Here, the *Inception* model does not suffer a severe downgrade from the presence of translations as the MLP. The reason why *Inception* does not report exactly the same results on D_0 and D_1, as predicted by Proposition 7.4 is because, even though convolutions and max-pooling are invariant to translations, in this experiment, translations were implemented by padding the image (for example with zeros), leading to slightly different results in the model output.

The PAC-Chernoff bound of Equation (7) neatly captures this phenomenon. As argued in the above section, the presence of transformed input-data implies that $\left(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_1}\right)^{-1}\left(\frac{1}{n}\ln\frac{kp}{\delta}\right) \geq \left(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0}\right)^{-1}\left(\frac{1}{n}\ln\frac{kp}{\delta}\right)$. However, if the model $\boldsymbol{\theta}$ is invariant to the input-data transformations present in the data-generating distribution ν_1, then, due to Proposition 7.4, the complexity term of our bound does not increase. This, in combination with the fact that our distribution-dependent PAC-Chernoff bound is perfectly tight for interpolators, provides, to the best of

Model	Train Acc.	Test Acc.	Test NLL
Inception	100.0%	74.08%	1.00
Inception-Shuffle	100.0%	42.46%	2.45
MLP	99.99%	51.69%	3.29
MLP-Shuffle	99.99%	51.12%	3.29
Initial Inception	10.00%	10.00%	2.30
Initial MLP	10.00%	9.96%	2.30

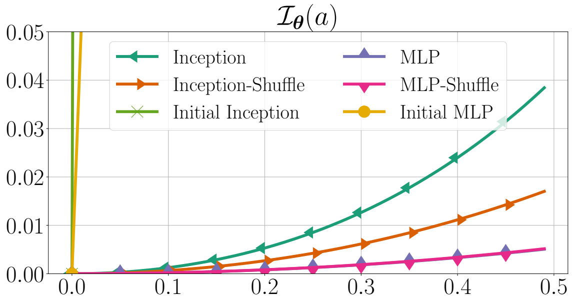

Figure 10: Results of the random shuffling experiment. Metrics of trained Inception and MLP models with and without a fixed random shuffling of the inputs in both train and test datasets. The rate function of all these models is shown in the right, each of them w.r.t. their corresponding shuffled or un-shuffled data distribution they where trained and evaluated on.

our knowledge, the best theoretical explanation of why invariant models interpolating the training data have a smaller generalization error.

The PAC-Chernoff bound explains why invariant interpolators have smaller generalization errors than non-invariant interpolators under transformed inputs.

The results in Figure 10 attempt to illustrate this behavior in a different way. The specific amount of invariances of an Inception network are unknown; however, we know at least this model is invariant to translations by the usage of convolutional layers. In this experiment, we attempt to remove this invariance by performing a pre-fixed random shuffling of the pixels of the images. One can understand this as an extra input-data transformation g_{T+1} that performs a permutation of the pixels on top of the data-generating distribution ν_T (we are implicitly assuming that Cifar10's data is generated by T unknown transformations of an unknown distribution ν_0). As a result of this, a convolutional network like *Inception* would no longer be invariant in the context of ν_{T+1}, at least, to image translations. According to the analysis at the beginning of the section, the empirical loss will be less concentrated under ν_{T+1} and, in consequence, the rate function will be lower with respect to ν_T. In the figure, we can see this effect in the rate functions of the learned Inception models, under ν_T (*Inception* label) and under ν_{T+1} (*Inception-Shuffle* label)[2].

The same experiment is performed to a MLP, which presents no local invariances; as a result and as expected, the rate function of this model is not affected by the presence of the random permutation. In fact, although the model is not invariant to shuffling, the architecture is; in the sense that for any MLP $\boldsymbol{\theta}$ and permutation g_{T+1}, there exists a MLP $\boldsymbol{\theta}'$ which makes the same predictions on ν_{T+1} that $\boldsymbol{\theta}$ does on ν_T (by permuting the weights in the first layer). Thus, a learning algorithm can find the "same" MLP under ν_T and ν_{T+1}. As a result, both MLP models present (nearly) the same rate function in the experiment. The table in this figure also shows how the generalization error is much higher for non-invariant models, as predicted by our theoretical analysis.

[2] Note that the rate functions are computed under the log-loss or negative log-likelihood (NLL).

7.2 Data Augmentation

Data augmentation is a widely-used technique in machine learning that enhances the robustness and generalization ability of models, particularly in the field of computer vision (Shorten & Khoshgoftaar, 2019). By applying various transformations like rotations, reflections, scaling, and translations to existing data samples, data augmentation artificially expands the dataset. This process helps the learned models to become invariant or less sensitive to these transformations, leading to improved performance on unseen data.

Recent research has delved deeply into the theoretical foundations of data augmentation (DA), with significant contributions coming from studies such as Chen et al. (2020) and Lyle et al. (2020). These works build on the assumption than the set of transformations G forms a *group* and that these transformations do not alter the data-generating distribution, also assuming *an equality in distribution assumption*. Within this context, these studies illustrate how DA can decrease variance and improve bounds on generalization error. Lyle et al. (2020) further presents a PAC-Bayes bound based on these premises, showing that data augmentation can lower the bound's complexity measure. However, these studies encounter a critical challenge: the *equality in distribution* assumption does not hold in reality, as it implies that any model $\boldsymbol{\theta}$ would yield identical expected losses on both transformed and non-transformed input data, a premise contradicted by evidence shown in Figure 9, where two models exhibit deteriorating performance as the data is increasingly transformed. Additionally, the PAC-Bayes bound introduced by Lyle et al. (2020) is not proven to be tight for interpolators. Thus, as highlighted in the introduction, merely demonstrating that data augmentation reduces the complexity term of a bound, which might be vacuous, does not assure a lower generalization error for an interpolator.

In this section, we demonstrate that our theoretical framework offers a more nuanced explanation. First, we will illustrate that data augmentation defines a new, more concentrated empirical loss. This finding is consistent with prior studies indicating that data augmentation decreases the variance of the empirical loss (Chen et al., 2020; Lyle et al., 2020). Second, we will explore how the complexity term of the PAC-Chernoff bound, introduced in Theorem 4.1, becomes smaller with the application of data augmentation. As this bound is also perfectly tight for interpolators trained on augmented datasets, we can deduce that the generalization error of interpolators under data augmentation is reduced, in opposite to the non-tight bounds used in earlier research (Lyle et al., 2020).

As argued, for example, by Chen et al. (2020), data augmentation is equivalent to optimize, using Monte-Carlo estimates, the so-called *data-augmented loss*, denoted as ℓ_G, which is the result of averaging the standard loss over all transformations obtained from a given input $\boldsymbol{x}$,

$$\ell_G(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = \mathbb{E}_{g\sim h}\left[\ell(\boldsymbol{y}, g(\boldsymbol{x}), \boldsymbol{\theta})\right] . \tag{8}$$

A pertinent question arises regarding whether the use of the augmented loss ℓ_G results in a more concentrated empirical loss (characterized by a larger rate function), given that more concentrated empirical losses lead to interpolators with a reduced generalization error.

The response to this inquiry is somewhat complex. Thus, in order to make the rest of the section easier to follow, we are summarizing now the key points of the discussion:

1. It is crucial to recognize that each transformed dataset, denoted D_T, corresponds to an original, unaltered dataset D_0, from which D_T is generated through random transformations $g_t \sim h_t$ for $t = 1, \ldots, T-1$. Two main objects are discussed:

   (a) The conventional empirical loss for transformed inputs $\hat{L}(D, \boldsymbol{\theta})$, here referred to as $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$ to highlight that $D_T \sim \nu_T^n$, and that we are using the standard non-augmented loss ℓ.

   (b) The augmented empirical loss for original, non-transformed inputs, referred to as $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$, with D_0 being the non-transformed dataset, $D_0 \sim \nu_0^n$, where $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta}) = \frac{1}{n}\sum_{\boldsymbol{y}_i, \boldsymbol{x}_i \in D_0} \ell_G(\boldsymbol{y}_i, \boldsymbol{x}_i, \boldsymbol{\theta})$ and ℓ_G as defined in Equation (8).

   It is important to keep in mind that $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ is an unknown quantity as D_0 is un-observed. However, $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ will play a fundamental role in explaining the underlying mechanics of data-augmentation.

2. Theorem 7.5 will showcase, using the rate function, that the augmented empirical loss for non-transformed inputs $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ exhibits a higher concentration than the empirical loss for transformed inputs $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$. As a result of this, and the fact that their expected value (population errors) are equal, minimizing $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ is a *better* optimization objective as observing lower generalization errors is more likely due to the Chernoff's bound in Theorem 3.3.

   However, in practice D_0 is unobserved, and DA involves minimizing $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ rather than $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$. Therefore, merely establishing that $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ is a better minimization objective than $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$ is insufficient for a nuance understanding of DA. The following points discuss the link between $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ and $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$.

3. Proposition 7.7 will show that $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ and $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ are equal when G is a group and h is an uniform distribution. As a result of this, an interpolator of $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ will interpolate for $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$, which exhibits a higher concentration than $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$.

   However, many DA techniques fall outside of this group hypothesis, such as rotations *within a finite range* or random-cropping.

4. Proposition 7.8 will reveal that both $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ and $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ share the same "optimal interpolators" (minimal error). In essence, if $\boldsymbol{\theta}$ serves as a perfect interpolator for $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$, then, it will similarly function as a perfect interpolator for $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$.

   As a conclusion on the discussion, such an interpolator is likely to exhibit a lower generalization error compared to those minimizing $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$, given that $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ is more concentrated than $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$. As a result of this and Chernoff's bound, an interpolator minimizing $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$ is more likely to exhibit a lower generalization error.

In the following result, we use $L^{\nu_0, \ell_G}(\boldsymbol{\theta})$ and $\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0, \ell_G}(a)$ to denote the expected loss and rate function, respectively, associated to the augmented loss ℓ_G under the un-transformed data-generating distribution ν_0. Recall that, under this notation, $L(\boldsymbol{\theta})$ would be equal to $L^{\nu_T, \ell}(\boldsymbol{\theta})$ and, similarly, $\mathcal{I}_{\boldsymbol{\theta}}(a)$ would be equal to $\mathcal{I}_{\boldsymbol{\theta}}^{\nu_T, \ell}(a)$, as they are implicitly defined over the standard loss ℓ and the (observed) transformed data-generating distribution ν_T, which is the data-generating distribution from which we observe the training data sets.

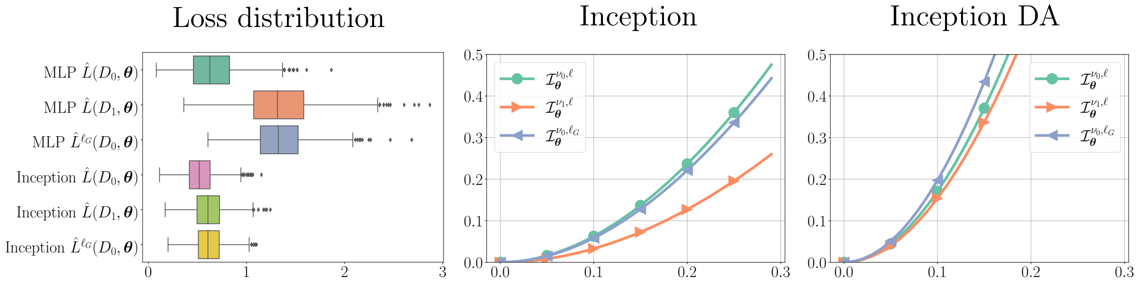

Figure 11: Illustration on the effect of data augmentation and transformed inputs. The distribution of $\hat{L}(D_0,\boldsymbol{\theta})$, $\hat{L}(D_1,\boldsymbol{\theta})$ and $\hat{L}^{\ell_G}(D_0,\boldsymbol{\theta})$ are shown on a fixed Multilayer perceptron and a fixed Inception model with similar amount of parameters for three different learning setups. For *MLP* and *Inception*, we use Cifar10's test set to simulate $D_0 \sim \nu_0^{50}$; transformed inputs, ($D_1 \sim \nu_1^{50}$), are simulated by applying random translations of 3 pixels, and rotations up-to 20° to Cifar10's test set. The augmented loss ℓ_G is computed using the same *augmentations* used to generate D_1. The obtained rate functions for the Inception model are shown in the middle figure. The last figure resembles the obtained rate functions with an Inception model trained using the augmented loss ℓ_G.

Theorem 7.5 (Proof)**.** *Under Assumption 2, it verifies that*

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad L^{\nu_T,\ell}(\boldsymbol{\theta}) = L^{\nu_0,\ell_G}(\boldsymbol{\theta}) \quad \textit{and} \quad \mathcal{I}_{\boldsymbol{\theta}}^{\nu_T,\ell}(a) \leq \mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G}(a) \;\; \forall a > 0\,.$$

Figure 11 illustrates the above result using a fixed MLP and a fixed Inception model over CIFAR-10. Figure 11 (left) shows how the distribution of the empirical loss over transformed data $\hat{L}^{\ell}(D_1,\boldsymbol{\theta})$ with $D_1 \sim \nu_1^{50}$ is more dispersed than the distribution of the empirical loss over the un-transformed data $\hat{L}^{\ell}(D_0,\boldsymbol{\theta})$ with $D_0 \sim \nu_0^{50}$ for both models. And, also, how the distribution of $\hat{L}^{\ell_G}(D_0,\boldsymbol{\theta})$ becomes more concentrated and with the same expectation than the distribution of $\hat{L}^{\ell}(D_1,\boldsymbol{\theta})$. Figure 11 (center) also shows how the rate functions of the Inception models on the three different setups changes as expected. We want to remark again that the models are fixed and the changes in the distribution of their empirical losses are due to changes in the data-generating distribution and/or the loss.

The following result is an adaptation of the PAC-Chernoff bound given in Theorem 4.1 for this setup, which describes the effect of using the data-augmented loss on the generalization error of interpolators.

Corollary 7.6 (Proof)**.** *With h.p.* $1-\delta$ *over* $D_0 \sim \nu_0^n$*, for all* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$*, simultaneously,*

$$\textit{if } \hat{L}^{\ell_G}(D_0,\boldsymbol{\theta}) \leq \epsilon \quad \textit{then} \quad \big(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G}\big)^{-1}\big(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\big) \leq L^{\nu_T,\ell}(\boldsymbol{\theta}) \leq \big(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G}\big)^{-1}\big(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\big) + \epsilon\,.$$

If we reformulate the above bound for interpolators under the standard loss,

$$\text{if } \hat{L}^{\ell}(D_T,\boldsymbol{\theta}) \leq \epsilon, \text{ then } \big(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_T,\ell}\big)^{-1}\big(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\big) \leq L^{\nu_T,\ell}(\boldsymbol{\theta}) \leq \big(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_T,\ell}\big)^{-1}\big(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\big) + \epsilon, \qquad (9)$$

and acknowledge that Theorem 7.5 also states that $(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_T,\ell})^{-1}(s) \geq (\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G})^{-1}(s)$ $\forall s > 0$, it becomes evident why an interpolator leveraging the augmented loss exhibits lower generalization error and thus superior generalization capability. According to this result, the process of learning by minimizing $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ is advantageous over minimizing $\hat{L}^{\ell}(D_T, \boldsymbol{\theta})$. As Theorem 7.5 states, the augmented loss results in a more concentrated empirical loss and a reduced inverse rate function. Consequently, as posited by Corollary 7.6, the generalization error associated with an interpolator is also diminished.

However, as previously highlighted, in the practice of data augmentation (DA), we do not minimize $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$ due to the lack of access to the unaltered dataset D_0; instead, we focus on minimizing $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$, which refers to the augmented loss calculated over the observed transformed dataset. If the set of transformations G forms a group and its associated probability distribution h is uniform —two prevalent assumptions in DA— it follows from the next result that minimizing $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ effectively amounts to minimizing $\hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$.

Proposition 7.7 (Proof)**.** *Under Assumption 2, if the transformations G define a group and its probability distribution h is uniform, then*

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = \hat{L}^{\ell_G}(D_0, \boldsymbol{\theta}) \,, \tag{10}$$

where D_0 is any un-transformed dataset and D_T is the corresponding transformed dataset obtained by applying a transformation $g \sim h$ to each of the samples in D_0.

When multiple transformations $g_1, \ldots, g_T$ are considered, the analysis can be seamlessly expanded into a sequence of equalities, provided that each corresponding set of transformations $G_1, \ldots, G_T$ forms a group and is governed by uniform distributions $h_1, \ldots, h_T$. From this angle, it becomes feasible to combine this approach with the insights from the previous section on invariant architectures, as these two methods could address different sets of chained transformations. For instance, convolutional neural networks naturally handle translations, whereas data augmentation can accommodate other types of transformations.

In summary, Theorem 7.5 shows that $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta})$ for a dataset $D_T \sim \nu_T^n$ is more concentrated than $\hat{L}(D_T, \boldsymbol{\theta})$ for a dataset $D \sim \nu_T^n$. Furthermore, the Chernoff bound presented in Corollary 7.6 indicates that interpolators operating with the augmented loss exhibit a reduced generalization error. To the best of our knowledge, this is the strongest result of this type explaining why interpolators under data augmentation (DA) have better generalization error, a result achieved through the application of distribution-dependent bounds.

> Theorem 7.5 shows that the augmented loss makes the empirical loss more concentrated. Then, the PAC-Chernoff bound of Corollary 7.6, explains why interpolators under the augmented loss have a smaller generalization error.

However, some transformations in data augmentation (DA) fail to form a group. For instance, rotations within the range of -20 to 20 degrees do not constitute a group because combining a rotation of 20 degrees with another of 10 degrees results in a 30-degree rotation, which exceeds the range. To establish a group of rotations, it is necessary to consider the full spectrum of rotations between -360 and 360 degrees, a range not typically employed in practice. Similarly, random cropping operations also do not create a group. In such scenarios, the analysis can still proceed under the assumption that $\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) \approx \hat{L}^{\ell_G}(D_0, \boldsymbol{\theta})$.

Therefore, by minimizing $\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta})$, we are, in effect, also indirectly minimizing the loss $\hat{L}^{\ell_G}(D_0,\boldsymbol{\theta})$, which is more concentrated and has interpolators with a reduced generalization error.

The subsequent result presents an alternative approach for handling transformations that do not inherently form a group. In such scenarios, it is necessary that for every transformation in the set G, an inverse transformation exists, denoted as $\forall g \in G$, $\exists g^{-1} \in G$. An example fulfilling this criterion is the set of rotations within the range of -20 to 20 degrees, where for any given rotation a in $[-20, 20]$, its inverse rotation $-a$ is also within the $[-20, 20]$ interval. Under these conditions, if $\boldsymbol{\theta}$ acts as a *perfect interpolator* with respect to $\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta})$ —the actual empirical loss targeted by the learning algorithm— then, $\boldsymbol{\theta}$ similarly qualifies as a *perfect interpolator* for $\hat{L}^{\ell}(D_0,\boldsymbol{\theta})$. Here, the term *perfect interpolator* is used to describe models that achieve the minimal possible empirical loss.

Proposition 7.8 (Proof)**.** *Given an un-transformed dataset D_0, for any transformed dataset D_T derived from D_0 under Assumption 2, if $m_{\boldsymbol{\theta}} := \operatorname{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, g(\boldsymbol{x}),\boldsymbol{\theta}) \quad \forall g \in G$, and $\forall g \in G$, $\exists g^{-1} \in G$. Then, it verifies that*

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \quad \hat{L}^{\ell_G}(D_T,\boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \hat{L}^{\ell}(D_0,\boldsymbol{\theta}) = m_{\boldsymbol{\theta}}\,.$$

This result implies that if a model achieves perfect interpolation on the transformed dataset D_T using the augmented loss, it concurrently achieves perfect interpolation on the original, untransformed dataset D_0 under the standard loss. This means that when the learning algorithm identifies an interpolator utilizing the augmented loss (as typically occurs in practice), it is essentially also identifying an interpolator for the untransformed dataset D_0. This scenario parallels the one encountered with invariant architectures, where the algorithm retrieves interpolators for the untransformed dataset. As highlighted at the outset of Section 7, interpolators for the untransformed dataset D_0 are associated with a lower generalization error compared to those for the transformed dataset D_T.

Figure 11 (right) illustrates the rate function of an inception model trained with data augmentation achieving almost perfect interpolation, where $\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta}) \approx m_{\boldsymbol{\theta}}$. It's particularly noteworthy to observe how this model's various rate functions are nearly indistinguishable, indicating that the model has nearly become invariant to rotations too. According to Proposition 7.4, if a model is invariant to rotations, then $\mathcal{I}^{\nu_0,\ell}(a) = \mathcal{I}^{\nu_1,\ell}(a)$, which aligns with the outcome of this experiment.

Within this theoretical framework, distinct from the methodologies referenced earlier, we can establish a connection between invariant architectures, data augmentation, and the discussions on explicit regularization presented in Section 6. Specifically, we can explain why these regularization techniques, in conjunction with invariant architectures or data augmentation, help to find interpolators with smaller generalization errors, a relationship already empirically showcased in Figure 2. This is achieved by extending the analysis from Section 6 to the new inverse rate functions $\left(\mathcal{I}^{\nu_0}_{\boldsymbol{\theta}}\right)^{-1}(\frac{1}{n}\ln\frac{k^p}{\delta})$ and $\left(\mathcal{I}^{\nu_0,\ell_G}_{\boldsymbol{\theta}}\right)^{-1}(\frac{1}{n}\ln\frac{k^p}{\delta})$, associated respectively with invariant architectures and data augmentation. For instance, implementing Proposition 6.2 on these inverse rate functions elucidates the benefit of ℓ_2-norm or distance-from-initialization regularization in the context of invariant architectures and/or data augmentation. Similarly, Proposition 6.5 clarifies the advantage of input-gradient normalization in these settings.

8. Over-Parameterization and Smooth Interpolation

Many theoretical contributions have established that over-parameterization is directly related to the strong performance of interpolators (Neyshabur et al., 2019; Bubeck et al., 2021). Specifically, the celebrated study by (Bubeck & Sellke, 2023)[3], building on an isoperimetry assumption, demonstrates that any model capable of interpolating training data beneath the noise threshold requires a (Euclidean) Lipschitz constant of the order at least $\sqrt{nd/p}$, where d represents the data's ambient dimension. We sketch this result using our notation: We say a model $\boldsymbol{\theta}$ is interpolating below the noise threshold if $\hat{L}(D,\boldsymbol{\theta}) \leq L^\star := \min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} L(\boldsymbol{\theta})$. Similarly, we denote $Lip(\boldsymbol{\theta})$ the Lipschitz constant of the model $\boldsymbol{\theta}$, as defined in Equation (6). Bubeck and Sellke (2023) (Theorem 1) can be informally stated as: under an isoperimetry assumption, for any $\epsilon \in (0, L^\star)$ and any $\delta \in (0,1)$, with high probability $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,

$$\text{if} \quad \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \quad \text{then} \quad Lip(\boldsymbol{\theta}) \geq \Omega\Big((L^\star - \epsilon)\sqrt{\tfrac{nd}{p}}\Big)\,. \tag{11}$$

This implies that to keep models with $O(1)$-Lipschitz constants that continue to interpolate the training data as the number of samples n grows, the number of parameters p must also increase. Bubeck and Sellke (2023) employs the Lipschitz constant as a measure of a model's *smoothness*. According to their findings, over-parameterization emerges as a crucial prerequisite for interpolators to exhibit small Lipschitz constants. Given that a lower Lipschitz constant enhances generalization, over-parameterization becomes a necessary condition for achieving interpolators with reduced generalization error.

In this section, we show how using PAC-Chernoff bounds we can arrive to a more nuanced understanding of the role that over-parameterization plays in generalization error of interpolators. In fact, the following result shows that over-parameterization is a requirement that naturally emerges from the PAC-Chernoff bound of Theorem 4.1. The following result derives a lower bound over the number of parameters of a model class based on the rate function of a model interpolating the data below the noise-threshold.

Theorem 8.1 (Proof)**.** *For any $\epsilon \in (0, L^\star)$ and any $\delta \in (0,1)$, with high probability $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$\textit{if} \quad \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \quad \textit{then} \quad p \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L^\star - \epsilon) + \ln \delta}{\ln k}\,.$$

The above bound connects the smoothness of an interpolator, measured by the rate function as discussed in Section 4, and the minimum number of parameters in the model class. As the size of the training dataset increases, the number of parameters must also increase linearly if we wish to maintain the same degree of *smoothness* in the model. This result generalizes the result of Bubeck and Sellke (2023) and *it does not require a isoperimetry assumption.*

Over-parameterization is a necessary condition for smooth interpolators.

[3] Outstanding Paper Award at NeurIPS'21.

The primary advantage of the aforementioned findings are their ability to establish a link between over-parameterization, interpolation, and our prior analyses about model's smoothness and its rate function. This enables us to show how Bubeck and Sellke (2023)'s analysis of *smooth interpolation*, using isoperimetry and Lipschitz continuity, represents merely one of several approximations to the characterization presented in Theorem 8.1. As described Section 6, the norm of parameters, the distance from initialization, the input-gradient norm, and the Lipschitz constant of a model, serve as proxies of the (inverse) rate function of a model and, consequently, of its smoothness. We first proceed to show how Bubeck and Sellke (2023)'s characterization of *smooth interpolation* can be directly deduced from Theorem 8.1 under the same isoperimetry assumption. This can also be derived under log-concave assumptions using the same techniques used in Proposition 6.5 and Equation (6).

Corollary 8.2 (Proof)**.** *If $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{x}$ with constant $Lip(\boldsymbol{\theta})$ and satisifies a c-isoperimetry assumption, then for any $\epsilon\in(0,L^\star)$ and any $\delta\in(0,1)$, with high probability $1-\delta$ over $D\sim\nu^n$, for all $\boldsymbol{\theta}\in\boldsymbol{\Theta}$, simultaneously,*

$$\textit{if}\quad \hat{L}(D,\boldsymbol{\theta})\leq\epsilon\quad\textit{then}\quad Lip(\boldsymbol{\theta})\geq\sqrt{\tfrac{nd}{2c(p\ln k-\ln\delta)}}(L^\star-\epsilon)\,.$$

We can similarly derive a bound with a giving norm or distance-from-initialization.

Corollary 8.3 (Proof)**.** *If the loss function $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{\theta}$ with constant $M>0$, then, for any $\boldsymbol{\theta}_0\in\boldsymbol{\Theta}_0=\{\boldsymbol{\theta}\in\boldsymbol{\Theta}\mid\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))=0\}\subset\boldsymbol{\Theta}$, any $\epsilon\in(0,L^\star)$ and any $\delta\in(0,1)$, with high probability $1-\delta$ over $D\sim\nu^n$, for all $\boldsymbol{\theta}\in\boldsymbol{\Theta}$, simultaneously,*

$$\textit{if}\quad \hat{L}(D,\boldsymbol{\theta})\leq\epsilon\quad\textit{then}\quad \|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2\geq\sqrt{\tfrac{n}{8M(p\ln k-\ln\delta)}}(L^\star-\epsilon)\,.$$

Corollary 8.2 and 8.3 imply that, as the number of samples n increases, to have *smooth* interpolators with a small Lipschitz constant, or a small parameter norm, or a small distance from initialization, we need to increase the number of parameters of the model class too. In any case, all these results are approximations of the general result given in Theorem 8.1. Note that a similar result could be derived in the context of input-gradient norms using Proposition 6.5.

> Interpolating with a small parameter norm, or a small distance from initialization, or a small input-gradient norm requires over-parameterization.

Remarkably, we can also link Theorem 8.1 with invariant architectures and data-augmentation given in Section 7. For example, Theorem 7.5, which says that the use of the data-augmented loss induced a higher rate function, in combination with Theorem 8.1, would explain why interpolating under data-augmentation will require, at some point, a higher number of parameters. Moreover, the same could be said about interpolating using more invariant architectures, as they define models with higher rate functions.

> Interpolating using data-augmentation and/or a model-invariant architecture requires over-parameterization.

To the best of our knowledge, we are not aware of any other results showing this kind of relationships between over-parameterization and smooth interpolators.

9. Related Work

In the context of uniform convergence bounds, many efforts have been made towards obtaining tighter generalization bounds (Kawaguchi et al., 2022; Bartlett et al., 2017; Neyshabur et al., 2017a; Golowich et al., 2018; Liang et al., 2019), by, for example, only considering the models effectively visited by the optimization algorithm. However, it is clear that these bounds are not actually meaningful (see Nagarajan & Kolter (2019b) for further references and discussions). The PAC-Bayes framework has also adapted to this new paradigm by exploiting properties of the models induced by the training data set (such as low spectral norm (Neyshabur et al., 2017b), noise stability (Arora et al., 2018b), de-randomization (Negrea et al., 2020), and compression (Arora et al., 2018b; Zhou et al., 2019)), but none of these bounds are shown to be empirically tight in the over-parameterized model classes and, for example, are unable to describe the interplay between invariant architectures and over-parameterization. In fact, very recently, Gastpar et al. (2024) proved that, for over-parameterized model classes, none of these bounds are tight for all data-generating distributions.

Our work mainly differs from these related works in the sense that we use *oracle complexity measures*, that is, PAC-Chernoff bounds and the rate function assumes access to the data-generating distribution. Consequently, we are able to tightly bound the generalization error of an interpolator, as elaborated in Section 4. This capability stems from our ability to derive robust results such as Theorem 4.6 even under unbounded losses. Furthermore, Theorem 4.6 and the PAC-Chernoff bound of Theorem 4.1 enables to leverage architectural invariances with respect to transformations inherent in the data-generating distribution.

Many other recent theoretical studies try to understand specific learning techniques in deep learning in an isolated and independent way to simplify the problem (e.g, (Vidal et al., 2020; Arora et al., 2016; Gilbert et al., 2017; Patel et al., 2016; Schoenholz et al., 2017; Poole et al., 2016; Tishby & Zaslavsky, 2015; Arora et al., 2018a; Gunasekar et al., 2018; Hardt & Ma, 2017; Soudry et al., 2018; Chen et al., 2020; Bubeck & Sellke, 2023)). As a result of such isolated approach, they are not fully able, most of the times, to establish links among different learning techniques, as done in this work. Furthermore, relying on the insights provided by the field of geometric deep learning (Bronstein et al., 2021), our work provides a theoretical explanation, from a statistical inference point of view, on why such invariances induce smoother model classes and better generalization.

10. Conclusions and Limitations

In this work, we discussed how a growing literature (Zhang et al., 2017; Nagarajan & Kolter, 2019b; Wang et al., 2024; Gastpar et al., 2024) is suggesting that bounds that solely depend on the training data are provably vacuous for over-parameterized model classes, and points to the need of exploring novel approaches to understand generalization in deep learning. Then, we argued in favor of distribution-dependent bounds, as bounds which directly depend on the data generating distribution, instead of the training data sample. A distribution-dependent PAC-Chernoff bound, outlined in Theorem 4.1, is introduced, which shows how this bound is perfectly tight for any interpolator. Based on the complexity measure of this bound, a novel definition of *smoothness* is proposed in Definition 3, which is able to provide a clear answer to the previously unanswered question of which interpolators generalize more effectively.

In Section 5, we showed how PAC-Chernoff bounds are able to capture the complex phenomenon of double-descent. Our bound is able to explain why is possible to have interpolators with a smaller generalization error and, at the same time, defined by a larger number of parameters.

In Section 6, we showed how the complexity measure of our PAC-Chernoff bound defines an regularizer that provides near-optimal performance for over-parameterized model classes interpolating the training data, as detailed in Theorem 6.1. During the rest of Section 6, we illustrated how a broad spectrum of existing regularization techniques can be shown to be proxies of this optimal regularizer. In certain cases, we even identified how some variations of existing regularizers can act as optimal surrogates for an interpolator's generalization error. This analysis serves to better understand the limitations of existing regularization techniques such as parameter norm, distance from initialization or input-gradient norm.

In Section 7, we described how having transformed inputs, as depicted in Assumption 2, make learning harder because interpolators have a higher generalization error, as shown by the PAC-Chernoff bound of Equation 7. The same PAC-Chernoff bound, in combination with other results, was also able to show why invariant architectures and data augmentation induce interpolator with smaller generalization error. We finished this section by discussing how this theoretical framework can establish a connection between invariant architectures, data augmentation, and the regularizers analyzed in Section 6.

Finally, in Section 8, we established how over-parameterization is an essential requisite for achieving smooth interpolators, with smoothness being definable across various dimensions, including parameter-norm, Lipschitz constant, or the employment of invariant architectures or data augmentation. Highlighting how this theoretical framework is able to establish connections among different, previously thought, unrelated approaches. We concluded the discussion by addressing the compatibility of larger model classes with smooth interpolators that exhibit superior generalization.

In summary, the primary insight from this study is that, distribution-dependent PAC-Chernoff bounds are a powerful tool to better understand the generalization capabilities of (over-parameterized) interpolators.

10.1 Discussion of Limitations

A limitation of this research is its assumption of a finite model class. We believe this limitation can be overcome by employing recently introduced PAC-Bayes-Chernoff bounds (Casado et al., 2024). This PAC-Bayesian bound is also a distribution-dependent bound, relying on a similar rate function. As a result, these findings could potentially be extended to infinite model classes. However, exploring such an extension is beyond the scope of this paper. Finally, this study has not delved into the significant role that stochastic gradient descent (SGD) plays in identifying interpolators with minimal generalization error.

Another limitation of our work is the lack of explicit connections between the smoothness definition introduced in this paper and other existing definitions in the literature. For instance, smoothness is often characterized using the largest eigenvalue of the Hessian matrix (Nesterov, 2003), the change in loss value within a perturbation radius (Keskar et al., 2017b), or Lipschitz continuity of the loss function (Shalev-Shwartz & Ben-David, 2014). While our notion of smoothness is tied directly to the PAC-Chernoff framework and its distribution-dependent complexity term, future work could explore how this concept aligns or diverges from these established definitions. Establishing such connections could provide additional insights into how our framework complements or generalizes existing theoretical constructs. We have chosen to focus here on the implications of our specific definition for generalization, leaving these broader comparisons for future research. This limitation underscores the potential for follow-up studies to bridge our smoothness definition with the broader landscape of smoothness concepts in machine learning theory.

Acknowledgements

AM acknowledges funding for cloud computing from Google Cloud for Researchers program, from Grant PID2022-139293NB-C31 funded by MCIN/AEI/10.13039/501100011033 and by ERDF, a way of making Europe. LO acknowledges financial support from PID2022-139856NB-I00 funded by MCIN/ AEI / 10.13039/501100011033 / FEDER, UE and from the Autonomous Community of Madrid (ELLIS Unit Madrid).

Appendix A. Experimental Settings

In this appendix we expand in the details of the conducted experiments. More precisely, the architectures used and the hyper-parameters and learning methods of each figure.

A.1 Learning Settings for Figures

A GitHub Repository with the conducted experiments can be found in https://github.com/Ludvins/2024_PAC-Chernoff-Bound. Regarding the experimental setting of this work, we have mainly used a small InceptionV3 (Szegedy et al., 2016) used in Zhang et al. (2017), and Cifar10 dataset (Krizhevsky et al., 2009). Before showing the specific specific architecture of our small Inception, taken from Zhang et al. (2017), we need to detail some of the modules that compose it:

1. Convolutional module: Convolutional layer, batch-normalization and ReLU activation.
2. Inception module with output channels $o_{1\times1}$ and $o_{3\times3}$: Consists on 2 different convolutional layers, one with kernel 1×1 and $o_{1\times1}$ output channels and another with kernel 3×3 and $o_{3\times3}$ output channels. The output of this layers is then concatenated, so the total number of output channels is $o_{1\times1} + o_{3\times3}$.
3. Downsample module: Convolutional module with kernel size 3, stride 2 and padding 0 and MaxPooling with kernel size of 3 and stride 2. The outputs of these two layers is concatenated.

With these elements, the architecture of our small InceptionV3 network is

1. Convolutional module with 96 output channels, kernel size 3, stride 1 and padding 0.
2. Inception Module with $o_{1\times1} = 32$ and $o_{3\times3} = 32$.
3. Inception Module with $o_{1\times1} = 32$ and $o_{3\times3} = 48$.

4. DownSample Module with $o_{3\times 3} = 80$.
5. Inception Module with $o_{1\times 1} = 112$ and $o_{3\times 3} = 48$.
6. Inception Module with $o_{1\times 1} = 96$ and $o_{3\times 3} = 64$.
7. Inception Module with $o_{1\times 1} = 80$ and $o_{3\times 3} = 80$.
8. Inception Module with $o_{1\times 1} = 48$ and $o_{3\times 3} = 96$.
9. DownSample Module with $o_{3\times 3} = 96$.
10. Inception Module with $o_{1\times 1} = 176$ and $o_{3\times 3} = 160$.
11. Inception Module with $o_{1\times 1} = 176$ and $o_{3\times 3} = 160$.
12. Adaptative Average Pooling layer with kernel 7×7.
13. Fully connected layer from 16464 to the number of classes (i.e, 10).

Where the total number of parameters of this model is 1.814.106.

Figure 2. For this experiments, all Inception models where trained using SGD with momentum 0.9 and learning rate 0.01 with exponential decay of 0.95. All models are trained for 30.000 iterations of batches of size 200 or until the train loss is under 0.005. These settings are selected to ensure that the random label model converges to an interpolator. Random cropping is employed using `RandomResizeCrop` function of `torchvision` with scale $(0.8, 1.0)$ and ratio $(0.9, 1.1)$. For ℓ_2 regularization, the multiplicative factor is 0.01.

Figure 3. For this figure, Standard, L2-Crop and Initial model from Figure 2 are used. Subsets of size $n = 50$ of CIFAR10's test split are used to approximate samples of the data generating distribution and build the histograms.

Figure 6. For this figure, we used a generalization of LeNet5 (three convolutional layers a two fully connected with ReLu activation and average pooling), where the number of channels of the convolutional layers was parameterized by k. Precisely, the first layer had 3 input and $\lfloor 6k \rceil$ output channels; the second layer $\lfloor 6k \rceil$ input and $\lfloor 16k \rceil$ output channels; and the last layer $\lfloor 16k \rceil$ input and $\lfloor 120k \rceil$ output channels. The set of models are created ranging k from 0.2 to 4.9 every 0.1; raising models from $7k$ parameters to models with $1.2M$ parameters. Each of this models is then trained until the train loss is lower than 0.01 or until the train loss has not lowered in two epochs (this only happens in the smallest models).

Figure 7. The rate function of a subset of all the models in Figure 6 is computed here.

Figure 9. The batch size is fixed to 250 and images are standardize (this was necessary to improve learning in the MLP model). The precise MLP has 3 hidden layers with 512 units, with a total of 1.735.178 parameters. All models are trained until the interpolation regime, that is, until the train loss is under 0.015, which, in the worst case where 20.000 iterations for the MLP. Inception models are trained using a learning rate of 0.001 whereas MLP models use 0.1, both with 0.9 momentum and 0.95 exponential decay. Regarding the data, D_0 is CIFAR10's test set, D_1 is the result of performing random translations of 5% and D_2 considers random translations of 5% and rotations of up-to 20%. Both transformations are computed using `RandomAffine` function of `torchvision`.

Figure 10. The model's specification and training setup is the same as in Figure 9. Regarding the data, the random shuffling of the pixels was performed using a random permutation using `Numpy`; the dataset was fully permuted and stores as a new dataset.

Figure 11. The model's specification and training setup is the same as in Figure 9. D_1 considers random translations of 5% and rotations of up-to 20% (the same as D_2 in Figure 9). Data-augmentation was produced using the same transformations as those that define D_1.

A.2 Estimating the Cumulant Function $J_{\boldsymbol{\theta}}(\lambda)$ and the Rate Function $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$

From the definition of the cummulant function $J_{\boldsymbol{\theta}}(\lambda)$,

$$J_{\boldsymbol{\theta}}(\lambda) = \ln \mathbb{E}_\nu \left[e^{\lambda(L(\boldsymbol{\theta}) - \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))} \right] = \ln \mathbb{E}_\nu \left[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \right] - \mathbb{E}_\nu [\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda] \,,$$

it is clear that computing its true value requires access to the true data generation distribution ν. However, in real-world problems, this distribution is unknown and innacesible.

The Machine Learning community is used to approximate this kind of quantities (such as the expected loss $L(\boldsymbol{\theta})$) using separate validation datasets D^{val}. In fact, due to the large amount of data available in nowadays problems, using this approach is perfectly doable, leading to

$$J_{\boldsymbol{\theta}}(\lambda) \approx \ln \left(\frac{1}{M} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{val}} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \right) - \frac{1}{M} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{val}} \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \,.$$

It is important to notice that the above estimator is biased due to the first term and Jensen's Inequality. In fact,

$$\begin{aligned} \mathbb{E}_{D^{val}} \left[\ln \left(\frac{1}{M} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{val}} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \right) \right] &\leq \ln \left(\mathbb{E}_{D^{val}} \left[\frac{1}{M} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{val}} p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \right] \right) \\ &= \ln \mathbb{E}_\nu \left[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \right] \,. \end{aligned}$$

As a result, if the size of D^{val} is not large enough, we might end understimating the cummulant function.

In regard of computational stability, computing the estimation in Equation (A.2) can be computationally unstable due to the use of probabilities. We encourage the use of log-probabilities and log-sum-exp operations as,

$$J_{\boldsymbol{\theta}}(\lambda) \approx \ln \left(\sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{val}} \exp(\lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})) \right) - \ln M - \frac{1}{M} \sum_{(\boldsymbol{x},\boldsymbol{y}) \in D^{val}} \lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) \,.$$

From this, it is straight-forward to compute the log-probabilities of the model (for example skipping the softmax layer of a NN), multiply them by λ and compute the mean and log-sum-exp of these quantities.

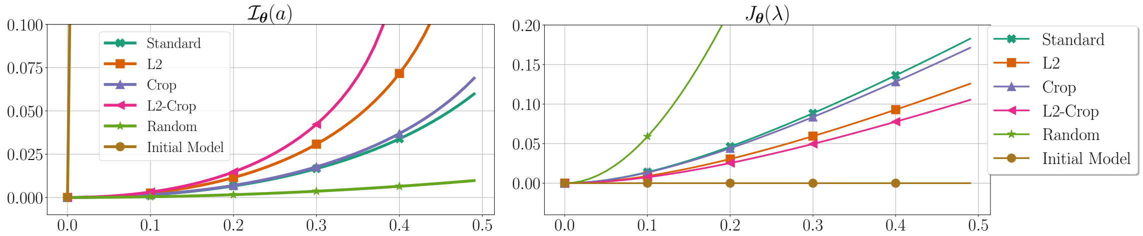

Figure 12: Illustration on the relationship between the rate function $\mathcal{I}_{\boldsymbol{\theta}}(a)$ (left) and the cummulant function $J_{\boldsymbol{\theta}}(\lambda)$ (right). They display the rate and the cummulant functions for the same set of models (same experiment than Figure 2).

Once the cummulant function has being approximated, computing the rate function relies on computing the optimal value of λ,

$$\mathcal{I}_{\boldsymbol{\theta}}(a) = \sup_{\lambda>0} \lambda a - J_{\boldsymbol{\theta}}(\lambda)\,.$$

In this regard, trying to optimize the value of λ doing automatic optimization resulted in a very unstable method in our experiments. Thus, the method we are recommending and the one we have employed is using a *binary search* algorithm. Fixed a range in which to optimize lambda $[\lambda_{min}, \lambda_{max}]$, a binary search algorithm has complexity $\mathcal{O}(\log_2(\lambda_{max} - \lambda_{min}))$. In fact, if (due to the nature of the problem) the needed value of λ_{max} is too large, one might perform the binary search in $[\ln(\lambda_{min}), \ln(\lambda_{max})]$, which has the same complexity but makes it easier to consider larger values of λ.

It is clear that computing the rate function is more complex than computing only the cummulant function (as the former requires the latter). In fact, the next result shows that it might not be necessary to compute the rate function, as the cummulant might be enough to characterize smoothness.

Proposition. *If $\forall\lambda \geq 0$, $J_{\boldsymbol{\theta}}(\lambda) \leq J_{\boldsymbol{\theta}'}(\lambda)$, then $\forall a \geq 0$, we have $\mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a)$.*

Proof. Direct consequence of the $\mathcal{I}_{\boldsymbol{\theta}}(a)$ being the Legendre transform of $J_{\boldsymbol{\theta}}(\lambda)$. □

From this result, if a model $\boldsymbol{\theta}$ has a higher cummulant function than another model $\boldsymbol{\theta}'$, then $\boldsymbol{\theta}$ is smoother than $\boldsymbol{\theta}'$ and many results apply. In fact, this is the method we recommend to analyze whether a model is smoother than another. Figure 12 clearly illustrates this case. *Just plotting the cummulants is enough to understand which models are smoother.*

Appendix B. Theorems and Proofs

B.1 The Rate Function (go back to Section 3)

Proposition 3.1 (Return)**.** *Under Assumption 1, $\forall\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\mathcal{I}_{\boldsymbol{\theta}}(\cdot)$ and $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(\cdot)$, are well defined. That is, $\forall a \in [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$, $\mathcal{I}_{\boldsymbol{\theta}}(a) < \infty$ and $\forall s \in \mathbb{R}^+_0$, $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(s) < \infty$.*

Proof. First, from Assumption 1, we have that $m_{\boldsymbol{\theta}} \geq 0$, then $\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \geq 0$ $\forall(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$. Then, it verifies that $\lambda(L(\boldsymbol{\theta}) - \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})) \leq \lambda L(\boldsymbol{\theta})$. Taking exponential and expectations,

$$\mathbb{E}_\nu\left[e^{\lambda(L(\boldsymbol{\theta}) - \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}))}\right] \leq \mathbb{E}_\nu\left[e^{\lambda L(\boldsymbol{\theta})}\right]\,.$$

Lastly, the expectation on the r.h.s. is constant, leading to

$$J_{\boldsymbol{\theta}}(\lambda) = \ln \mathbb{E}_\nu \left[e^{\lambda(L(\boldsymbol{\theta})-\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))}\right] \leq \ln \mathbb{E}_\nu \left[e^{\lambda L(\boldsymbol{\theta})}\right] = \lambda L(\boldsymbol{\theta})\,.$$

As, by Assumption 1, $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$, $L(\boldsymbol{\theta}) < \infty$ and the function $J_{\boldsymbol{\theta}}(\lambda)$ is well-defined for $\lambda > 0$. Next we need to show that the supremum over λ is reached in the definition of the rate function. For that, we are showing that it is actually a maximum. First, we have that,

$$\frac{\partial}{\partial\lambda}(\lambda a - J_{\boldsymbol{\theta}}(\lambda)) = a - \frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda)\,,$$

Where the second derivative is negative as $\frac{\partial^2}{\partial\lambda^2} J_{\boldsymbol{\theta}}(\lambda) \geq 0$ (the cumulant is convex). As a result, when the previous derivative is zero, the maximum is reached. In fact the optimum of λ is a $\lambda^\star$ such that $a = \frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda^\star)$. Then, $\forall a \in (0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$, we need to show that $\exists \lambda^\star \in \mathbb{R}^+$. We have that $\frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda)$ is a continuous function, as it is combination of continuous functions:

$$\frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda) = \frac{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]}{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda]} - \mathbb{E}_\nu[\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})]\,.$$

By standard properties of the cumulant generating function, we have that $\frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(0) = 0$. On the other had, also by standard properties of the cummulant (Herdegen, 2008, Lemma 1), we have that

$$\lim_{\lambda\to\infty} \frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda) = \operatorname*{ess\,sup}_{(\boldsymbol{x},\boldsymbol{y})} \; L(\boldsymbol{\theta}) - \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}\,. \tag{12}$$

Thus, if $\frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda)$ is continuous, $\frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(0) = 0$ and $\lim_{\lambda\to\infty} \frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda) = L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$, then $\forall a \in [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$ there always exist a $\lambda^\star \in \mathbb{R}^+$ such that $a = \frac{\partial}{\partial\lambda} J_{\boldsymbol{\theta}}(\lambda^\star)$. Thus, for such values of a the rate function is finite and well defined.

For $a \geq L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$, we have that the supremum is reached when $\lambda \to \infty$, because $J_{\boldsymbol{\theta}}(\lambda)$ is monotonically increasing. Every $a \geq L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$, can be written as $a = L(\boldsymbol{\theta}) - b$, where $b \leq m_{\boldsymbol{\theta}}$. Then the limit when $\lambda \to \infty$ for any $a \geq L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$ can be written as follows,

$$\begin{aligned}\lim_{\lambda\to\infty} \lambda(L(\boldsymbol{\theta}) - b) - J_{\boldsymbol{\theta}}(\lambda) &= \lim_{\lambda\to\infty} -\lambda b - \ln \mathbb{E}_\nu \left[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\right] = \lim_{\lambda\to\infty} -\ln \mathbb{E}_\nu \left[\left(\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{e^{-b}}\right)^\lambda\right] \\ &= -\ln \mathbb{E}_\nu \left[\lim_{\lambda\to\infty} \left(\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{e^{-b}}\right)^\lambda\right] = -\ln \mathbb{E}_\nu \left[\mathbb{1}(p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = e^{-b}\right] \\ &= -\ln \mathbb{E}_\nu[\mathbb{1}(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}) = b)] = -\ln \mathbb{P}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}) = b)\,.\end{aligned}$$

If $b < m_{\boldsymbol{\theta}}$ or, equivalently, $a > L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}$, we then have $\mathcal{I}_{\boldsymbol{\theta}}(a) = \infty$. And if $a = L - m_{\boldsymbol{\theta}}$, we then have $\mathcal{I}_{\boldsymbol{\theta}}(a) = -\ln \mathbb{P}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}) = m_{\boldsymbol{\theta}})$, which, may be well defined or equal to ∞.

We now proceed to show that inverse rate is well defined. By definition of the inverse rate function,

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s) = \inf_{\lambda\geq 0} \frac{s + J_{\boldsymbol{\theta}}(\lambda)}{\lambda} \leq \lim_{\lambda\to\infty} \frac{s + J_{\boldsymbol{\theta}}(\lambda)}{\lambda} = \lim_{\lambda\to\infty} \frac{J_{\boldsymbol{\theta}}(\lambda)}{\lambda}$$

Then, by L'Hôpital's rule and Equation (12), we have

$$\lim_{\lambda\to\infty}\frac{J_{\boldsymbol\theta}(\lambda)}{\lambda} = \lim_{\lambda\to\infty}\nabla_\lambda J_{\boldsymbol\theta}(\lambda) = L(\boldsymbol\theta) - m_{\boldsymbol\theta}$$

From Assumption 1 we have that $L(\boldsymbol\theta) < \infty$ and $m_{\boldsymbol\theta} \geq 0$, then we deduce that $\forall s \geq 0$ we have that $\mathcal{I}_{\boldsymbol\theta}^{-1}(s) < \infty$. □

Proposition 3.4 (Return)**.** *For any fixed $\boldsymbol\theta \in \boldsymbol\Theta$ and $n > 0$, it satisfies*

$$\lim_{a\to L(\boldsymbol\theta)-m_{\boldsymbol\theta}} \mathbb{P}_{D\sim\nu^n}\Big(L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq a\Big) = \lim_{a\to L(\boldsymbol\theta)-m_{\boldsymbol\theta}} e^{-n\mathcal{I}_{\boldsymbol\theta}(a)}\,.$$

Proof. It is clear from the definition of $m_{\boldsymbol\theta} := \operatorname{ess\,inf}_{(\boldsymbol y,\boldsymbol x)\sim\nu}\ell(\boldsymbol y,\boldsymbol x,\boldsymbol\theta)$ and the rate function $\mathcal{I}_{\boldsymbol\theta}(a)$, that the limit at both sides is 0. □

B.2 The Rate Function Characterizes the Generalization of Interpolators (go back to Section 4)

Theorem 4.1 (Return)**.** *With h.p. $1-\delta$ over $D\sim\nu^n$, for all $\boldsymbol\theta\in\boldsymbol\Theta$, simultaneously,*

$$L(\boldsymbol\theta) \leq \hat{L}(D,\boldsymbol\theta) + \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\,.$$

Proof. By Chernoff's Theorem 3.3, for a given $\boldsymbol\theta$, we have, $\mathbb{P}\Big(L(\boldsymbol\theta)-\hat{L}(D,\boldsymbol\theta)\geq a\Big) \leq e^{-n\mathcal{I}_{\boldsymbol\theta}(a)}$. Naming $\delta' = e^{-n\mathcal{I}_{\boldsymbol\theta}(a)}$ and re-arranging terms, $a = \mathcal{I}_{\boldsymbol\theta}^{-1}\left(-\tfrac{1}{n}\ln\delta'\right)$. This allows us to rewrite the first equation as

$$\mathbb{P}\Big(L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{1}{\delta'}\right)\Big) \leq \delta'\,.$$

Reversing and using an union bound over the set of models,

$$\mathbb{P}\Big(\bigcup_{\boldsymbol\theta\in\boldsymbol\Theta} L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{1}{\delta'}\right)\Big) \leq \sum_{\boldsymbol\theta\in\boldsymbol\Theta}\mathbb{P}\Big(L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{1}{\delta'}\right)\Big)\,.$$

As we have k^p different models, the r.h.s. can be rewritten as

$$\mathbb{P}\Big(\bigcup_{\boldsymbol\theta\in\boldsymbol\Theta} L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{1}{\delta'}\right)\Big) \leq k^p\delta'\,.$$

By reparameterizing the above inequality with $\delta' = \delta k^{-p}$ we have

$$\mathbb{P}\Big(\bigcup_{\boldsymbol\theta\in\boldsymbol\Theta} L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\Big) \leq \delta\,.$$

Which verifies,

$$1-\mathbb{P}\Big(\bigcup_{\boldsymbol\theta\in\boldsymbol\Theta} L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \geq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\Big) \geq 1-\delta\,.$$

Which is equivalent to,

$$\mathbb{P}\Big(\bigcap_{\boldsymbol\theta\in\boldsymbol\Theta} L(\boldsymbol\theta) - \hat{L}(D,\boldsymbol\theta) \leq \mathcal{I}_{\boldsymbol\theta}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\Big) \geq 1-\delta\,.$$

□

Corollary 4.2 (Return). *With h.p.* $1-\delta$ *over* $D\sim\nu^n$*, for all* $\boldsymbol{\theta}\in\boldsymbol{\Theta}$*, simultaneously,*

$$L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{|\bar{\boldsymbol{\Theta}}|}{\delta}\right).$$

Proof. Thet proof follows the same approach as Theorem 4.1, with the union bound applied to the models in $\bar{\boldsymbol{\Theta}}$ rather than directly to the models in $\boldsymbol{\Theta}$. This adjustment is valid because, when applying the union bound, we only need to consider the total number of distinct random variables; in this case, $\hat{L}(D,\boldsymbol{\theta})$ with $D\sim\nu^n$. Models in $\boldsymbol{\Theta}$ that are not in $\bar{\boldsymbol{\Theta}}$ define random variables that are duplicated by definition and therefore do not need to be accounted for in the application of the union bound. □

Theorem 4.3 (Return). *For any* $\delta\in(0,1)$ *and any* $\boldsymbol{\theta}\in\boldsymbol{\Theta}$*, it verifies that*

$$\lim_{n\to\infty}\ \sqrt{n}\,\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) = \sqrt{2\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))\ln\tfrac{k^p}{\delta}}\,.$$

Proof. Let $\bar{Z}_n := \sqrt{n}(L(\boldsymbol{\theta})-\hat{L}(D,\boldsymbol{\theta}))$ and $J_{\bar{Z}_n}, \mathcal{I}_{\bar{Z}_n}$ and $\mathcal{I}^{-1}_{\bar{Z}_n}$ denote its cumulant, rate and inverse rate function. Then, by properties of the cummulant, it verifies that $J_{\bar{Z}_n}(\lambda) = nJ_{\boldsymbol{\theta}}\big(\frac{\lambda}{\sqrt{n}}\big)$. Using the definition of the rate function, we can deduce that

$$\mathcal{I}_{\bar{Z}_n}(a) = \sup_\lambda\ a\lambda + nJ_{\boldsymbol{\theta}}\big(\tfrac{\lambda}{\sqrt{n}}\big) = n\sup_\lambda\ \tfrac{\lambda}{\sqrt{n}}\tfrac{a}{\sqrt{n}} + J_{\boldsymbol{\theta}}\big(\tfrac{\lambda}{\sqrt{n}}\big) = n\mathcal{I}_{\boldsymbol{\theta}}\big(\tfrac{a}{\sqrt{n}}\big)\,.$$

Using a second order Taylor expansion over $\mathcal{I}_{\boldsymbol{\theta}}(a)$ around $a=0$, where $\mathcal{I}_{\boldsymbol{\theta}}(0)=0$ and $\frac{\partial}{\partial a}\mathcal{I}_{\boldsymbol{\theta}}(0)=0$, we got that

$$\mathcal{I}_{\boldsymbol{\theta}}\big(\tfrac{a}{\sqrt{n}}\big) = \tfrac{1}{2}\tfrac{\partial^2}{\partial a^2}\mathcal{I}_{\boldsymbol{\theta}}(0)\big(\tfrac{a}{\sqrt{n}}\big)^2 + \big(\tfrac{a}{\sqrt{n}}\big)^2 h\big(\tfrac{a}{\sqrt{n}}\big)\,,$$

where $\lim_{t\to 0}h(t)=0$. Furthermore, it verifies that (Ellis, 2012)

$$\tfrac{\partial^2}{\partial a^2}\mathcal{I}_{\boldsymbol{\theta}}(0) = \left(\tfrac{\partial^2}{\partial\lambda^2}J_{\boldsymbol{\theta}}(0)\right)^{-1}.$$

From there, we have that

$$\begin{aligned}\frac{\partial^2}{\partial\lambda^2}J_{\boldsymbol{\theta}}(\lambda) &= -\frac{\partial}{\partial\lambda}\frac{\mathbb{E}_\nu[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})]}{\mathbb{E}_\nu[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}]}\\ &= \frac{\mathbb{E}_\nu[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))^2]}{\mathbb{E}_\nu[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}]} - \frac{\mathbb{E}_\nu[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})]^2}{\mathbb{E}_\nu[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}]^2}.\end{aligned}$$

Which evaluated at $\lambda=0$, gives

$$\tfrac{\partial^2}{\partial\lambda^2}J_{\boldsymbol{\theta}}(0) = \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) =: \sigma^2$$

Then,

$$\lim_{n\to\infty}\mathcal{I}_{\bar{Z}_n}(a) = \lim_{n\to\infty}n\mathcal{I}_{\boldsymbol{\theta}}\big(\tfrac{a}{\sqrt{n}}\big) = \lim_{n\to\infty}n\frac{1}{2}\sigma^{-2}\big(\tfrac{a}{\sqrt{n}}\big)^2 + n\big(\tfrac{a}{\sqrt{n}}\big)^2h\big(\tfrac{a}{\sqrt{n}}\big) = \frac{1}{2}\sigma^{-2}a^2\,.$$

Then, as both the rate and its inverse are continuous functions, it verifies that

$$\lim_{n\to\infty} \mathcal{I}^{-1}_{\bar{Z}_n}(s) = \left(\lim_{n\to\infty} \mathcal{I}_{\bar{Z}_n}\right)^{-1}(s) = \sqrt{2\sigma^2 s}$$

By definition of the inverse rate, we got that

$$\mathcal{I}^{-1}_{\bar{Z}_n}(s) = \inf_\lambda \frac{s + J_{\bar{Z}_n}(\lambda)}{\lambda} = \inf_\lambda \frac{s + nJ_{\boldsymbol{\theta}}(\frac{\lambda}{\sqrt{n}})}{\lambda} = \sqrt{n}\inf_\lambda \frac{\frac{s}{n} + nJ_{\boldsymbol{\theta}}(\frac{\lambda}{\sqrt{n}})}{\frac{\lambda}{\sqrt{n}}} = \sqrt{n}\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{s}{n}\right) .$$

As a result,

$$\lim_{n\to\infty} \sqrt{n}\mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{s}{n}\right) = \lim_{n\to\infty} \mathcal{I}^{-1}_{\bar{Z}_n}(s) = \sqrt{2\sigma^2 s}\,.$$

□

Proposition 4.4 (Return)**.** *With h.p. $1-\delta$ over $D\sim\nu^n$, for all $\boldsymbol{\theta}\in\boldsymbol{\Theta}$, simultaneously,*

$$\text{if} \quad \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \quad \text{then} \quad 0 \leq L(\boldsymbol{\theta}) - \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \epsilon\,. \tag{3}$$

Proof. By Theorem 4.1, $L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$, and, by Proposition 3.1, $\mathcal{I}_{\boldsymbol{\theta}}(a)$ is well defined $\forall a\in[0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$. Then, by Proposition 3.2, $\forall b>0$, $\mathcal{I}^{-1}_{\boldsymbol{\theta}}(b) \in [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$. In consequence, we have,

$$L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}^{-1}_{\boldsymbol{\theta}}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \hat{L}(D,\boldsymbol{\theta}) + L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}}\,.$$

The result follows from $m_{\boldsymbol{\theta}} \geq 0$, $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$ and rearranging terms.. □

Proposition 4.5 (Return)**.** *For any $\boldsymbol{\theta}\in\boldsymbol{\Theta}$ and $\boldsymbol{\theta}'\in\boldsymbol{\Theta}'$,*

$$\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) \leq \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}')) \iff \exists\beta>0 \text{ s.t. } \boldsymbol{\theta} \text{ is } \beta\text{-smoother than } \boldsymbol{\theta}'\,.$$

Proof. Let $\bar{Z}_n := \sqrt{n}(L(\boldsymbol{\theta}) - \hat{L}(D,\boldsymbol{\theta}))$ and $J_{\bar{Z}_n}$, $\mathcal{I}_{\bar{Z}_n}$ and $\mathcal{I}^{-1}_{\bar{Z}_n}$ denote its cumulant, rate and inverse rate function. Then, by properties of the cummulant, it verifies that $J_{\bar{Z}_n}(\lambda) = nJ_{\boldsymbol{\theta}}\big(\frac{\lambda}{\sqrt{n}}\big)$. Using the definition of the rate function, we can deduce that

$$\mathcal{I}_{\bar{Z}_n}(a) = \sup_\lambda\ a\lambda + nJ_{\boldsymbol{\theta}}\big(\tfrac{\lambda}{\sqrt{n}}\big) = n\sup_\lambda\ \tfrac{\lambda}{\sqrt{n}}\tfrac{a}{\sqrt{n}} + J_{\boldsymbol{\theta}}\big(\tfrac{\lambda}{\sqrt{n}}\big) = n\mathcal{I}_{\boldsymbol{\theta}}\big(\tfrac{a}{\sqrt{n}}\big)\,.$$

Using a second order Taylor expansion over $\mathcal{I}_{\boldsymbol{\theta}}(a)$ around $a=0$, where $\mathcal{I}_{\boldsymbol{\theta}}(0)=0$ and $\frac{\partial}{\partial a}\mathcal{I}_{\boldsymbol{\theta}}(0) = 0$, we got that

$$\mathcal{I}_{\boldsymbol{\theta}}\big(\tfrac{a}{\sqrt{n}}\big) = \tfrac{1}{2}\tfrac{\partial^2}{\partial a^2}\mathcal{I}_{\boldsymbol{\theta}}(0)\big(\tfrac{a}{\sqrt{n}}\big)^2 + \big(\tfrac{a}{\sqrt{n}}\big)^2 h\big(\tfrac{a}{\sqrt{n}}\big)\,,$$

where $\lim_{t\to 0} h(t) = 0$. Furthermore, it verifies that

$$\tfrac{\partial^2}{\partial a^2}\mathcal{I}_{\boldsymbol{\theta}}(0) = \left(\tfrac{\partial^2}{\partial\lambda^2}J_{\boldsymbol{\theta}}(0)\right)^{-1} = \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))^{-1} =: \sigma^{-2}\,.$$

Then,

$$\lim_{n\to\infty} \mathcal{I}_{\bar{Z}_n}(a) = \lim_{n\to\infty} n\mathcal{I}_{\boldsymbol{\theta}}(\tfrac{a}{\sqrt{n}}) = \lim_{n\to\infty} n\frac{1}{2}\sigma^{-2}\big(\tfrac{a}{\sqrt{n}}\big)^2 + n\big(\tfrac{a}{\sqrt{n}}\big)^2 h\big(\tfrac{a}{\sqrt{n}}\big) = \frac{1}{2}\sigma^{-2}a^2\,. \tag{13}$$

Let us assume that $\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) \leq \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}'))$. Then, we aim to show that there exists $\beta > 0$ such that $\mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a) \quad \forall a \leq \beta$. For a fixed value of a, by the limit in Equation 13, we got that for any $\epsilon > 0$, there exists $n_0(a), n_0'(a) > 0$ such that

$$\left|n\mathcal{I}_{\boldsymbol{\theta}}(\tfrac{a}{\sqrt{n}}) - \tfrac{1}{2}\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))^{-2}a^2\right| < \epsilon \quad \forall n > n_0(a)\,,$$

and

$$\left|n\mathcal{I}_{\boldsymbol{\theta}'}(\tfrac{a}{\sqrt{n}}) - \tfrac{1}{2}\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}'))^{-2}a^2\right| < \epsilon \quad \forall n > n_0'(a)\,.$$

Let $\epsilon = \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))^{-2}a^2 - \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}'))^{-2}a^2 \geq 0$ and $n_0^\star(a) = \max\{n_0(a), n_0'(a)\}$. Then,

$$\mathcal{I}_{\boldsymbol{\theta}}(\tfrac{a}{\sqrt{n}}) \geq \mathcal{I}_{\boldsymbol{\theta}'}(\tfrac{a}{\sqrt{n}}) \quad \forall n > n_0^\star(a)\,.$$

This implies that $\mathcal{I}_{\boldsymbol{\theta}}(c) \geq \mathcal{I}_{\boldsymbol{\theta}'}(c)$, for any $c \leq a/\sqrt{n_0^\star(a)}$, as there exists $n = (a/c)^2$ verifying $c = a/\sqrt{n}$. We may consider then $\beta := \sup_{a>0}\{a/\sqrt{n_0^\star(a)}\} > 0$, which might be infinite. It verifies that

$$\mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a) \quad \forall a \leq \beta\,.$$

Let us now assume that there exists $\beta > 0$ such that $\boldsymbol{\theta}$ is β-smoother than $\boldsymbol{\theta}'$, or, equivalently $\mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a) \quad \forall a \leq \beta$. Then,

$$\lim_{n\to\infty} n\mathcal{I}_{\boldsymbol{\theta}}(\tfrac{a}{\sqrt{n}}) \geq \lim_{n\to\infty} n\mathcal{I}_{\boldsymbol{\theta}'}(\tfrac{a}{\sqrt{n}})\,.$$

Using the limit in Equation 13, we got

$$\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))^{-2}a^2 \geq \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}'))^{-2}a^2\,,$$

leading to

$$\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) \leq \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}'))\,.$$

□

Theorem 4.6 (Return)**.** *For any $\epsilon \geq 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^p$ and $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$, simultaneously,*

$$\textit{if } \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \textit{ and } \boldsymbol{\theta} \textit{ is } \mathcal{I}_{\boldsymbol{\theta}}^{-1}\big(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\big)\textit{-smoother than } \boldsymbol{\theta}', \textit{ then, } L(\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}') + \epsilon\,.$$

Proof. If $\boldsymbol{\theta}$ is β-smoother than $\boldsymbol{\theta}'$, by Definition 3, $\forall a \in (0,\beta] \quad \mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a)$, where $\beta = \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$. Then, we have that

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\,.$$

As the rate function $\mathcal{I}_{\boldsymbol{\theta}'}(a)$ is invertible and its image lies in $[0, L(\boldsymbol{\theta}') - m_{\boldsymbol{\theta}'})$, where $m_{\boldsymbol{\theta}'} \geq 0$, due to Assumption 1, we have that $\mathcal{I}_{\boldsymbol{\theta}'}^{-1}(s) \in [0, L(\boldsymbol{\theta}') - m_{\boldsymbol{\theta}'})$. In consequence, we also have

$$\mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq L(\boldsymbol{\theta}')\,.$$

As $\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) \leq \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$, it verifies $\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) \leq L(\boldsymbol{\theta}')$. By the PAC-Chernoff bound of Theorem 4.1 and because $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$, we have with h.p. $1-\delta$ over $D \sim \nu^n$,

$$L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \,.$$

Combining the last two inequalities, we have

$$L(\boldsymbol{\theta}) \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \epsilon + L(\boldsymbol{\theta}')\,.$$

The statement of the theorem directly derives from the above inequality. □

B.3 Explicit Regularization (go back to Section 6)

Theorem 6.1 (Return). *For any $\epsilon > 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, $|L(\boldsymbol{\theta}_\epsilon^\star) - L(\boldsymbol{\theta}_\epsilon^\times)| \leq \epsilon$.*

Proof. From the definitions of $\boldsymbol{\theta}_\epsilon^\times$ and $\boldsymbol{\theta}_\epsilon^\star$, given by

$$\boldsymbol{\theta}_\epsilon^\star = \mathop{\arg\min}_{\boldsymbol{\theta}\,:\,\hat{L}(D,\boldsymbol{\theta})\,\leq\,\epsilon} L(\boldsymbol{\theta})\,, \qquad \boldsymbol{\theta}_\epsilon^\times = \mathop{\arg\min}_{\boldsymbol{\theta}\,:\,\hat{L}(D,\boldsymbol{\theta})\,\leq\,\epsilon} \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)\,,$$

it is clear that $L(\boldsymbol{\theta}_\epsilon^\star) \leq L(\boldsymbol{\theta}_\epsilon^\times)$. On the other hand, because for any $s \geq 0$, $\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s) \in [0, L(\boldsymbol{\theta}) - m_{\boldsymbol{\theta}})$ (Proposition 3.2), we have that $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$:

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) + \hat{L}(D,\boldsymbol{\theta}) \leq L(\boldsymbol{\theta}) + \hat{L}(D,\boldsymbol{\theta})\,.$$

By definition of $\boldsymbol{\theta}_\epsilon^\times$ and $\boldsymbol{\theta}_\epsilon^\star$,

$$\mathcal{I}_{\boldsymbol{\theta}_\epsilon^\times}^{-1}(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}) + \hat{L}(D,\boldsymbol{\theta}_\epsilon^\times) \leq \mathcal{I}_{\boldsymbol{\theta}_\epsilon^\star}^{-1}(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}) + \hat{L}(D,\boldsymbol{\theta}_\epsilon^\star)\,,$$

which gives

$$\mathcal{I}_{\boldsymbol{\theta}_\epsilon^\times}^{-1}(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}) + \hat{L}(D,\boldsymbol{\theta}_\epsilon^\times) \leq L(\boldsymbol{\theta}_\epsilon^\star) + \hat{L}(D,\boldsymbol{\theta}_\epsilon^\star) \leq L(\boldsymbol{\theta}_\epsilon^\star) + \epsilon\,.$$

This, in combination with the PAC-Chernoff bound of Theorem 4.1 gives

$$L(\boldsymbol{\theta}_\epsilon^\times) \leq \mathcal{I}_{\boldsymbol{\theta}_\epsilon^\times}^{-1}(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}) + \hat{L}(D,\boldsymbol{\theta}_\epsilon^\times) \leq L(\boldsymbol{\theta}_\epsilon^\star) + \epsilon\,.$$

From this, we deduce that, $L(\boldsymbol{\theta}_\epsilon^\times) \leq L(\boldsymbol{\theta}_\epsilon^\star) + \epsilon$. Thus, we got that $L(\boldsymbol{\theta}_\epsilon^\star) \leq L(\boldsymbol{\theta}_\epsilon^\times)$ and $L(\boldsymbol{\theta}_\epsilon^\times) \leq L(\boldsymbol{\theta}_\epsilon^\star) + \epsilon$, finishing the proof. □

Proposition 6.2 (Return). *If the loss function $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{\theta}$ with constant $M > 0$, then, for any $\boldsymbol{\theta}_0 \in \boldsymbol{\Theta}_0 = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid \mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})) = 0\}$, it verifies that*

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \sqrt{2Ma}\;\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2\,,$$

where $a = min\big(1, \frac{1}{n}\ln\frac{k^p}{\delta}\big)$.

Proof. If the loss is Lipschitz continuous with constant M, $\forall y, x, \boldsymbol{\theta} \quad \|\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\|_2^2 \leq M$. Then, $J_{\boldsymbol{\theta}}(\lambda)$ verifies

$$\begin{aligned}
\|\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\lambda)\|_2^2 &= || -\lambda \mathbb{E}_{\nu p^\lambda}\left[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\right] + \lambda \mathbb{E}_{\nu}\left[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\right] ||_2^2 \\
&\leq \lambda^2 \mathbb{E}_{\nu p^\lambda}\left[\|\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\|_2^2\right] + \lambda^2 \mathbb{E}_{\nu}\left[\|\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\|_2^2\right] \\
&\leq 2M\lambda^2\,,
\end{aligned}$$

where $\mathbb{E}_{\nu p^\lambda}\left[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\right] = \frac{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})]}{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda]}$. With this, we have

$$|J_{\boldsymbol{\theta}}(\lambda) - J_{\boldsymbol{\theta}_0}(\lambda)| \leq 2M\lambda^2 \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2 \implies J_{\boldsymbol{\theta}}(\lambda) \leq 2M\lambda^2 \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2\,.$$

Then, for any $a \geq 0$, we have $\frac{a + J_{\boldsymbol{\theta}}(\lambda)}{\lambda} \leq \frac{a + 2M\lambda^2\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2}{\lambda}$; where by definition of the inverse rate we got

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}(a) \leq \frac{a + J_{\boldsymbol{\theta}}(\lambda)}{\lambda} \leq \frac{a + 2M\lambda^2\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2}{\lambda}\,.$$

As the inequality holds for any $\lambda \geq 0$, we take the one that minimizes the r.h.s, leading to $\mathcal{I}_{\boldsymbol{\theta}}^{-1}(a) \leq \sqrt{2Ma}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2$. On the other hand, $\mathcal{I}_{\boldsymbol{\theta}}^{-1}(a)$ is Lipschitz with constant M as

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{I}_{\boldsymbol{\theta}}^{-1}(a)\|_2^2 = \left\|\frac{\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\lambda_a^\star)}{\lambda_a^\star}\right\|_2^2 \leq \frac{\|\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\lambda_a^\star)\|_2^2}{\lambda_a^{\star,2}} \leq 2M\,.$$

In consequence, $(\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) - \mathcal{I}_{\boldsymbol{\theta}_0}^{-1}(\frac{1}{n}\ln\frac{k^p}{\delta}))^2 \leq 2M||\boldsymbol{\theta} - \boldsymbol{\theta}_0||^2$. Which implies that $\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) \leq \sqrt{2M}||\boldsymbol{\theta} - \boldsymbol{\theta}_0||_2$. Thus, it simultaneously holds that

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \sqrt{2M}||\boldsymbol{\theta} - \boldsymbol{\theta}_0||_2 \quad \text{and} \quad \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \sqrt{2M\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2\,.$$

□

Corollary 6.3 (Return)**.** *Considering the log-loss, if it belongs to the exponential family with a constant base measure, that is, there exist $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^p$, $a : \boldsymbol{\Theta} \to \mathbb{R}$ and $k \in \mathbb{R}$ such that*

$$\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) := -\ln p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T s(\boldsymbol{x}, \boldsymbol{y}) - a(\boldsymbol{\theta}) + k \quad \forall (\boldsymbol{x}, \boldsymbol{y}) \sim \nu\,;$$

Then, $\forall \epsilon > 0$, exists $n_0 > 0$ such that $\forall n > n_0$:

$$\left|\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) - \sqrt{2\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\sqrt{\boldsymbol{\theta}^T Cov_\nu(s(\boldsymbol{y}, \boldsymbol{x}))\boldsymbol{\theta}}\right| \leq \epsilon\,,$$

where $Cov_\nu(\cdot)$ is the covariance w.r.t. ν of the sufficient statistics of each $(\boldsymbol{x}, \boldsymbol{y})$ sample.

Proof. Using Theorem 4.3, we got that $\lim_{n\to\infty} \sqrt{n}\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right) = \sqrt{2\mathbb{V}_\nu(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}))\ln\frac{k^p}{\delta}}$. The proof concludes from the fact that using the exponential family,

$$\mathbb{V}_\nu\big(\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\big) = \mathbb{V}_\nu\big(\boldsymbol{\theta}^T s(\boldsymbol{y}, \boldsymbol{x}) - a(\boldsymbol{\theta}) + k\big) = \boldsymbol{\theta}^T \mathrm{Cov}_\nu\big(s(\boldsymbol{y}, \boldsymbol{x})\big)\boldsymbol{\theta}\,.$$

□

Proposition 6.4 (Return). *For any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, if the equality of Equation (5) holds, then*

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) = \sqrt{2\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\sqrt{\left(\boldsymbol{\theta}-\boldsymbol{\theta}_0\right)^T Cov_\nu\left(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0)\right)\left(\boldsymbol{\theta}-\boldsymbol{\theta}_0\right)}\,.$$

Proof. A second-order Taylor expansion of $J_{\boldsymbol{\theta}}(\lambda)$ w.r.t. to $\boldsymbol{\theta}$ centered around $\boldsymbol{\theta}_0$ is

$$J_{\boldsymbol{\theta}_0}(\lambda) + \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}_0}(\lambda)(\boldsymbol{\theta}-\boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}}J_{\boldsymbol{\theta}_0}(\lambda)(\boldsymbol{\theta}-\boldsymbol{\theta}_0)\,.$$

By standard properties of the cumulant generating function over centered random variables, we have $J_{\boldsymbol{\theta}_0}(\lambda) = 0$. While the $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\lambda)$ can be expressed as,

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\lambda) = \lambda\mathbb{E}_{\nu p^\lambda}\left[\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\right] - \lambda\mathbb{E}_{\nu}\left[\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\right]\,,$$

where νp^λ denotes $\nu p^\lambda(\boldsymbol{y},\boldsymbol{x}) = \frac{\nu(\boldsymbol{y},\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda}{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda]}$. At $\boldsymbol{\theta}_0$, we have that $\nu p^\lambda = \nu$, because, by definition, $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0)$ is constant for any $(\boldsymbol{y},\boldsymbol{x})$. In consequence, the gradient at $\boldsymbol{\theta}_0$ simplifies as,

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}_0}(\lambda) = \lambda\mathbb{E}_{\nu}\left[\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\right] - \lambda\mathbb{E}_{\nu}\left[\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\right] = 0\,.$$

The Hessian of the cumulant $J_{\boldsymbol{\theta}}(\lambda)$ w.r.t. to $\boldsymbol{\theta}$ can be written as follows:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\lambda) = {} & \lambda^2 Cov_{\nu p^\lambda}(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})) + \lambda\mathbb{E}_{\nu p^\lambda}\left[\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\right] \\ & - \lambda\mathbb{E}_{\nu}\left[\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})\right]\,.\end{aligned}$$

Again, at $\boldsymbol{\theta}_0$, we have that $\nu p^\lambda = \nu$, so the Hessian of the cumulant $J_{\boldsymbol{\theta}}(\lambda)$ at $\boldsymbol{\theta}_0$ simplifies as, $\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}} J_{\boldsymbol{\theta}_0}(\lambda) = \lambda^2 Cov_\nu(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0))$. With this, the second order Taylor expansion of $J_{\boldsymbol{\theta}}(\lambda)$ evaluated on $\boldsymbol{\theta}_0$ is $\frac{\lambda^2}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\mathrm{Cov}_\nu(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0))(\boldsymbol{\theta}-\boldsymbol{\theta}_0)$.

On the other hand, if, at the definition of $\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s)$ given in Equation (2), we replace $J_{\boldsymbol{\theta}}(\lambda)$ by the above Taylor expansion, we have that,

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s) = \inf_{\lambda>0}\ \frac{s}{\lambda} + \frac{\lambda}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\mathrm{Cov}_\nu(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0))(\boldsymbol{\theta}-\boldsymbol{\theta}_0)\,.$$

The optimal value of λ is acquired derivating the above expression w.r.t. λ, giving,

$$\frac{-s}{\lambda^2} + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\mathrm{Cov}_\nu(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0))(\boldsymbol{\theta}-\boldsymbol{\theta}_0) = 0\,.$$

One may show that it is a minimum by taking the second derivative. From this, the optimal value of λ is $\lambda^\star = \left(\frac{2s}{(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\mathrm{Cov}_\nu(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0))(\boldsymbol{\theta}-\boldsymbol{\theta}_0)}\right)^{\frac{1}{2}}$. Using this expression,

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s) = 2s\sqrt{\frac{1}{2s}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\mathrm{Cov}_\nu(\nabla_{\boldsymbol{\theta}}\ln p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}_0))(\boldsymbol{\theta}-\boldsymbol{\theta}_0)}\,.$$

Evaluating the expression on $s = \frac{1}{n}\ln\frac{k^p}{\delta}$ and $\boldsymbol{\theta} = \boldsymbol{0}$ concludes the proof. □

Proposition 6.5 (Return). *If $\nu(\boldsymbol{x},\boldsymbol{y})$ is an uniformly strictly log-concave density and $\nu(\boldsymbol{y}|\boldsymbol{x})$ is deterministic, then $\exists M > 0$, such that, for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,*

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) \leq \sqrt{\tfrac{1}{n}\ln\tfrac{k^p}{\delta}}\sqrt{M\mathbb{E}_\nu\left[\left\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\right\|_2^2\right]}\,.$$

Proof. Using Chafaï (2004)'s Corollary 2.1 on $\phi(f) = -\ln f$ and $f_{\boldsymbol{\theta}}(y, \boldsymbol{x}) = e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}$; we get that

$$J_{\boldsymbol{\theta}}(\lambda) \leq M\lambda^2\, \mathbb{E}_\nu\left[\left\|\nabla_x \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})\right\|_2^2\right].$$

Then,

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s) = \inf_{\lambda>0} \frac{s + J_{\boldsymbol{\theta}}(\lambda)}{\lambda} \leq \inf_{\lambda>0} \frac{s + M\lambda^2\mathbb{E}_\nu[|\nabla_x \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})|^2]}{\lambda}.$$

Deriving w.r.t. to λ to compute the optimal value, gives $\lambda_{inf} = \sqrt{\frac{s}{M\mathbb{E}_\nu[|\nabla_x \ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})|^2]}}$. Using this,

$$\mathcal{I}_{\boldsymbol{\theta}}^{-1}(s) \leq \sqrt{s}\sqrt{M\mathbb{E}_\nu[|\nabla_x \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})|^2]}.$$

□

B.4 Invariances (go back to Section 7)

Proposition 7.1 (Return). *Under Assumption 2, let $\boldsymbol{x}_t$ denote the random variable denoting a transformation of an input $\boldsymbol{x}_0$ using a composition of t transformations, then*

$$MI(\boldsymbol{y}\,; \boldsymbol{x}_t) \leq MI(\boldsymbol{y}\,; \boldsymbol{x}_{t-1}) \quad \forall t \in \mathbb{N}^+.$$

As a consequence, $MI(\boldsymbol{y}; \boldsymbol{x}) = MI(\boldsymbol{y}; \boldsymbol{x}_T) \leq MI(\boldsymbol{y}; \boldsymbol{x}_0)$.

Proof. According to Assumption 2, the targets $\boldsymbol{y}$ are independent of the input $\boldsymbol{x}_t$ given $\boldsymbol{x}_{t-1}$, that is, $\boldsymbol{y} \perp \boldsymbol{x}_t | \boldsymbol{x}_{t-1}$. This is a result of the fact that $\boldsymbol{x}_{t-1}$ d-connects $\boldsymbol{x}_t$ and $\boldsymbol{y}$ in the graph representation of Assumption 2. The result follows then by the data processing inequality. □

Proposition 7.2 (Return). *Under Assumption 2, if $\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is convex under $\boldsymbol{x}$ and $\mathbb{E}_{g_t}[g_t(\boldsymbol{x})] = \boldsymbol{x}$, then, $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$, $L^{\nu_{t+1}}(\boldsymbol{\theta}) \geq L^{\nu_t}(\boldsymbol{\theta})$.*

Proof. By definition we have $L^{\nu_{t+1}}(\boldsymbol{\theta}) = \mathbb{E}_{\nu_{t+1}}[\ell(\boldsymbol{y}, \boldsymbol{x}_{t+1}), \boldsymbol{\theta})] = \mathbb{E}_{\nu_t}\mathbb{E}_{g_{t+1}}[\ell(\boldsymbol{y}, g_{t+1}(\boldsymbol{x}_t), \boldsymbol{\theta})]$. As $\ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is convex in $\boldsymbol{x}$, the expectation under g_{t+1} can be moved inside ℓ:

$$L^{\nu_{t+1}}(\boldsymbol{\theta}) \geq \mathbb{E}_{\nu_t}[\ell(\boldsymbol{y}, \mathbb{E}_{g_{t+1}}[g_{t+1}(\boldsymbol{x}_t)], \boldsymbol{\theta})].$$

Given that $\mathbb{E}_{g_{t+1}}[g_{t+1}(\boldsymbol{x})] = \boldsymbol{x}$, we got $L^{\nu_{t+1}}(\boldsymbol{\theta}) \geq \mathbb{E}_{\nu_t}[\ell(\boldsymbol{y}, \boldsymbol{x}_t, \boldsymbol{\theta})] = L^{\nu_t}(\boldsymbol{\theta})$. □

Proposition 7.3 (Return). *Under Assumption 2, if the change observed in the loss after a transformation $\Delta(\boldsymbol{y}, \boldsymbol{x}, g, \boldsymbol{\theta}) = \ell(\boldsymbol{y}, g(\boldsymbol{x}), \boldsymbol{\theta}) - \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$ is statistically independent of the loss itself, $\Delta(\boldsymbol{y}, \boldsymbol{x}, g, \boldsymbol{\theta}) \perp \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$, then*

$$\mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a) \leq \mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}(a) \quad \forall a > 0.$$

Proof. First of all we will show that $J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) \geq J_{\boldsymbol{\theta}}^{\nu_t}(\lambda)$; from there, the result is immediate due to definition of the inverse rate. By definition, we got $J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) = \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_{t+1}}[e^{-\ell(\boldsymbol{y},\boldsymbol{x}_{t+1},\boldsymbol{\theta})}]$, where ν_{t+1} can be expanded leading to

$$J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) = \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_t}\mathbb{E}_{g_{t+1}}\left[e^{-\lambda\ell(\boldsymbol{y},g_{t+1}(\boldsymbol{x}_t),\boldsymbol{\theta})}\right].$$

Using the decomposition of the loss, $J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) = \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_t}\mathbb{E}_{g_{t+1}}\left[e^{-\lambda(\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})+\Delta(\boldsymbol{y},\boldsymbol{x}_t,g_{t+1},\boldsymbol{\theta}))}\right]$, which can be written as

$$J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) = \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_t}\left[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})}\right] + \ln \mathbb{E}_{\nu_t}\mathbb{E}_{g_{t+1}}\left[e^{-\lambda\Delta(\boldsymbol{y},\boldsymbol{x}_t,g_{t+1},\boldsymbol{\theta})}\right].$$

Using Jensen's inequality in the last term,

$$J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) \geq \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_t}\left[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})}\right] - \lambda\mathbb{E}_{\nu_t}\mathbb{E}_{g_{t+1}}[\Delta(\boldsymbol{y},\boldsymbol{x}_t,g_{t+1},\boldsymbol{\theta})],$$

and using the definition of $\Delta(\boldsymbol{y},\boldsymbol{x}_t,g_{t+1},\boldsymbol{\theta})$,

$$J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) \geq \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_t}\left[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})}\right] - \lambda\mathbb{E}_{\nu_t}\mathbb{E}_{g_{t+1}}[\ell(\boldsymbol{y},g_{t+1}(\boldsymbol{x}_t),\boldsymbol{\theta})] + \lambda\mathbb{E}_{\nu_t}[\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})]].$$

By definition of the expected loss,

$$\begin{aligned} J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) &\geq \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \ln \mathbb{E}_{\nu_t}[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})}] - \lambda L^{\nu_{t+1}}(\boldsymbol{\theta}) + \lambda L^{\nu_t}(\boldsymbol{\theta}) \\ &= \ln \mathbb{E}_{\nu_t}[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x}_t,\boldsymbol{\theta})}] + \lambda L^{\nu_t}(\boldsymbol{\theta}) = J_{\boldsymbol{\theta}}^{\nu_t}(\lambda). \end{aligned}$$

From this inequality and by considering standard properties of the Legendre transform, we have $J_{\boldsymbol{\theta}}^{\nu_{t+1}}(\lambda) \geq J_{\boldsymbol{\theta}}^{\nu_t}(\lambda) \implies \mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a) \leq \mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}(a)$. □

Proposition 7.4 (Return)**.** *Under Assumption 2, if a model $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is G_t-invariant then,*

$$L^{\nu_{t+1}}(\boldsymbol{\theta}) = L^{\nu_t}(\boldsymbol{\theta}) \quad \textit{and} \quad \mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a) = \mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}(a) \quad \forall a > 0.$$

Proof. It immediately follows from the definition of model invariant (Definition 4) and the definitions of $L^{\nu_{t+1}}(\boldsymbol{\theta})$, $L^{\nu_t}(\boldsymbol{\theta})$, $\mathcal{I}_{\boldsymbol{\theta}}^{\nu_{t+1}}(a)$ and $\mathcal{I}_{\boldsymbol{\theta}}^{\nu_t}$. □

Theorem 7.5 (Return)**.** *Under Assumption 2, it verifies that*

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}, \quad L^{\nu_T,\ell}(\boldsymbol{\theta}) = L^{\nu_0,\ell_G}(\boldsymbol{\theta}) \quad \textit{and} \quad \mathcal{I}_{\boldsymbol{\theta}}^{\nu_T,\ell}(a) \leq \mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G}(a)\ \forall a > 0.$$

Proof. Under Assumption 2, it is clear that

$$L^{v_0,\ell_G}(\boldsymbol{\theta}) = \mathbb{E}_{\nu_0}[\ell_G(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))] = \mathbb{E}_{\nu_0}\mathbb{E}_{g\sim h}[\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta}))] = \mathbb{E}_{\nu}[\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))] = L^{\nu_T,\ell}(\boldsymbol{\theta}).$$

On the other hand, from the definition of the cummulant function, $J_{\boldsymbol{\theta}}(\lambda) = \ln \mathbb{E}_{\nu_T}[e^{-\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})}] + \lambda\mathbb{E}_{\nu_T}[\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})]$. Where by definition

$$J_{\boldsymbol{\theta}}(\lambda) = \ln \mathbb{E}_{\nu_0}\mathbb{E}_{g\sim h}[e^{-\lambda\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})}] + \lambda\mathbb{E}_{\nu_0}\mathbb{E}_{g\sim h}[\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})],$$

where expectations can be exchanged as

$$J_{\boldsymbol{\theta}}(\lambda) = \ln \mathbb{E}_{\nu_0(\boldsymbol{x})}\mathbb{E}_{\nu(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_g[e^{-\lambda\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})}] + \lambda\mathbb{E}_{\nu_0(\boldsymbol{x})}\mathbb{E}_{\nu(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_g[\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})]\,.$$

Applying Jensen's inequality to the exponential,

$$J_{\boldsymbol{\theta}}(\lambda) \geq \ln \mathbb{E}_{\nu_0(\boldsymbol{x})}\mathbb{E}_{\nu(\boldsymbol{y}|\boldsymbol{x})}\left[e^{-\lambda\mathbb{E}_g[\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})]}\right] + \lambda\mathbb{E}_{\nu_0(\boldsymbol{x})}\mathbb{E}_{\nu(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_g[\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta})]\,.$$

Where by definition of ℓ_G,

$$J_{\boldsymbol{\theta}}(\lambda) \geq \ln \mathbb{E}_{\nu_0}[e^{-\lambda[\ell_G(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})]}] + \lambda\mathbb{E}_{\nu_0}[\ell_G(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})] = J_{\boldsymbol{\theta}}^{\nu_0,\ell_G}(\lambda)\,.$$

From this point, the inequality regarding the inverse rate is clear from the definition. □

Corollary 7.6 (Return)**.** *With h.p. $1-\delta$ over $D_0 \sim \nu_0^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$\text{if } \hat{L}^{\ell_G}(D_0,\boldsymbol{\theta}) \leq \epsilon \quad \text{then} \quad \left(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G}\right)^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\right) \leq L^{\nu_T,\ell}(\boldsymbol{\theta}) \leq \left(\mathcal{I}_{\boldsymbol{\theta}}^{\nu_0,\ell_G}\right)^{-1}\left(\tfrac{1}{n}\ln\tfrac{kp}{\delta}\right) + \epsilon\,.$$

Proof. Under Assumption 2, it is clear that

$$L^{v_0,\ell_G}(\boldsymbol{\theta}) = \mathbb{E}_{\nu_0}[\ell_G(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))] = \mathbb{E}_{\nu_0}\mathbb{E}_{g\sim h}[\ell(\boldsymbol{y},g(\boldsymbol{x}),\boldsymbol{\theta}))] = \mathbb{E}_{\nu}[\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))] = L^{\nu_T,\ell}(\boldsymbol{\theta})\,.$$

Where it verifies that $L(\boldsymbol{\theta}) := L^{\nu_T,\ell}(\boldsymbol{\theta})$. Thus, the result comes from the application of Theorem 4.1.

□

Proposition 7.7 (Return)**.** *Under Assumption 2, if the transformations G define a group and its probability distribution h is uniform, then*

$$\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta}) = \hat{L}^{\ell_G}(D_0,\boldsymbol{\theta})\,, \tag{10}$$

where D_0 is any un-transformed dataset and D_T is the corresponding transformed dataset obtained by applying a transformation $g \sim h$ to each of the samples in D_0.

Proof. On one hand, by definition of ℓ_G it verifies that

$$\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta}) = \frac{1}{n}\sum_i \mathbb{E}_{g'\sim h}[\ell(\boldsymbol{y}_i,g'(\boldsymbol{x}_i),\boldsymbol{\theta})] = \frac{1}{n}\sum_i \int h(g')\ell(\boldsymbol{y}_i,g'(g_i(x_{0,i})),\boldsymbol{\theta})\;dg'\,.$$

As the set of transformations define a group, for each transformation g_i that generated each input, there exists g_i^{-1}. Furthermore, there exists $l := g' \circ g_i \in G_1$ such that $g' = l \circ g_i^{-1}$. Thus,

$$\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta}) = \frac{1}{n}\sum_i \int_{g'=l\circ g_i^{-1}} h(l\circ g_i^{-1})\ell(\boldsymbol{y}_i,l\circ g_i^{-1}\circ g_i(x_{0,i}),\boldsymbol{\theta})\;dg'\,.$$

Then, using a change of variables

$$\hat{L}^{\ell_G}(D_T,\boldsymbol{\theta}) = \frac{1}{n}\sum_i \int_l h(l\circ g_i^{-1})\ell(\boldsymbol{y}_i,l(x_{0,i}),\boldsymbol{\theta})\;dl\,.$$

As h is uniform $h(l \circ g_i^{-1}) = h(l)$,

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = \frac{1}{n} \sum_i \int_l h(l) \ell(\boldsymbol{y}_i, l(x_{0,i}), \boldsymbol{\theta}) dl = \hat{L}^{\ell_G}(D_0, \boldsymbol{\theta}) .$$

□

Proposition 7.8 (Return). *Given an un-transformed dataset D_0, for any transformed dataset D_T derived from D_0 under Assumption 2, if $m_{\boldsymbol{\theta}} := \operatorname{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, g(\boldsymbol{x}), \boldsymbol{\theta}) \quad \forall g \in G$, and $\forall g \in G$, $\exists g^{-1} \in G$. Then, it verifies that*

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \quad \hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \hat{L}^{\ell}(D_0, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} .$$

Proof. First of all, by definition we got that $m_{\boldsymbol{\theta}} = \operatorname{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})$. Then, it verifies that

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \ell_G(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) = \operatorname*{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \quad \forall (\boldsymbol{x}, \boldsymbol{y}) \in D_T .$$

Thus, by definition of ℓ_G:

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \mathbb{E}_g[\ell(\boldsymbol{y}, g(\boldsymbol{x}), \boldsymbol{\theta})] = \operatorname*{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \quad \forall (\boldsymbol{x}, \boldsymbol{y}) \in D_T .$$

Now, using that $\operatorname{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, g(\boldsymbol{x}), \boldsymbol{\theta}) = m_{\boldsymbol{\theta}}$ $\forall g \in G$, reaching the essential infimum in expectation means all the losses inside the expectation reach such infimum:

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \ell(\boldsymbol{y}, g(\boldsymbol{x}), \boldsymbol{\theta}) = \operatorname*{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \quad \forall g \in G, \forall (\boldsymbol{x}, \boldsymbol{y}) \in D_T .$$

Using that for every $(\boldsymbol{x}_0, \boldsymbol{y}) \in D_0$ exists $g' \in G$ associated to that input such that $\boldsymbol{x} = g'(\boldsymbol{x}_0)$ and that

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \ell(\boldsymbol{y}, g \circ g'(\boldsymbol{x}_0), \boldsymbol{\theta}) = \operatorname*{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \quad \forall g \in G, \forall (\boldsymbol{x}_0, \boldsymbol{y}) \in D_0 .$$

Then, using that there exists $g'^{-1} \in G$, we got that

$$\hat{L}^{\ell_G}(D_T, \boldsymbol{\theta}) = m_{\boldsymbol{\theta}} \iff \ell(\boldsymbol{y}, \boldsymbol{x}_0, \boldsymbol{\theta}) = \operatorname*{ess\,inf}_{(\boldsymbol{x},\boldsymbol{y})\sim\nu_T} \ell(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}) \quad \forall (\boldsymbol{x}_0, \boldsymbol{y}) \in D_0 .$$

As a result, the empirical loss $\hat{L}^{\ell}(D_0, \boldsymbol{\theta})$ is equal to $m_{\boldsymbol{\theta}}$ too. □

B.5 Over-Parameterization (go back to Section 8)

Theorem 8.1 (Return). *For any $\epsilon \in (0, L^\star)$ and any $\delta \in (0,1)$, with high probability $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$\textit{if} \quad \hat{L}(D, \boldsymbol{\theta}) \leq \epsilon \quad \textit{then} \quad p \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L^\star - \epsilon) + \ln \delta}{\ln k} .$$

Proof. Due to fact that $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$ and to Theorem 4.1, we have that with high probability $1-\delta$ over $D \sim \nu^n(y,x)$,

$$L(\boldsymbol{\theta}) \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) .$$

Then, just rearranging terms as follows, we get $L(\boldsymbol{\theta}) - \epsilon \leq \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)$. Applying the rate function at both sides,

$$\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon) \leq \tfrac{1}{n}\ln\tfrac{k^p}{\delta} = \frac{1}{n}\left(\ln k^p - \ln\delta\right) .$$

Then, we got $\ln k^p \geq n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon) + \ln\delta$ and

$$p \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta}{\ln k} .$$

As $L^\star \leq L(\boldsymbol{\theta})$ and the rate function is monotonically increasing, then

$$p \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta}{\ln k} \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L^\star-\epsilon)+\ln\delta}{\ln k} .$$

□

Corollary 8.2 (Return)**.** *If $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{x}$ with constant $Lip(\boldsymbol{\theta})$ and satisifies a c-isoperimetry assumption, then for any $\epsilon \in (0, L^\star)$ and any $\delta \in (0,1)$, with high probability $1-\delta$ over $D \sim \nu^n$, for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$\text{if} \quad \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \quad \text{then} \quad Lip(\boldsymbol{\theta}) \geq \sqrt{\tfrac{nd}{2c(p\ln k - \ln\delta)}}(L^\star - \epsilon) .$$

Proof. Under the isoperimetry assumption, we have $\mathcal{I}_{\boldsymbol{\theta}}(a) \leq \frac{da^2}{2cLip(\boldsymbol{\theta})^2}$. Due to fact that $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$ and to Theorem 4.1, we have that with high probability $1-\delta$ over $D \sim \nu^n(y,x)$,

$$L(\boldsymbol{\theta}) \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right) .$$

Then, just rearranging terms as follows, we get $L(\boldsymbol{\theta}) - \epsilon \leq \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^p}{\delta}\right)$. Applying the rate function at both sides,

$$\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon) \leq \tfrac{1}{n}\ln\tfrac{k^p}{\delta} = \frac{1}{n}\left(\ln k^p - \ln\delta\right) .$$

Then, we got $\ln k^p \geq n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon) + \ln\delta$ and

$$p \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta}{\ln k} .$$

As $L^\star \leq L(\boldsymbol{\theta})$ and the rate function is monotonically increasing, then

$$p \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta}{\ln k} \geq \frac{n\mathcal{I}_{\boldsymbol{\theta}}(L^\star-\epsilon)+\ln\delta}{\ln k} .$$

Using the upper bound over the rate function we obtained before, we got that

$$p \geq \frac{n\frac{d(L^\star-\epsilon)^2}{2cLip(\boldsymbol{\theta})^2}+\ln\delta}{\ln k} .$$

Re-arranging terms,

$$\sqrt{\frac{nd(L^\star-\epsilon)^2}{2c(p\ln k-\ln\delta)}}\leq Lip(\boldsymbol{\theta})\,.$$

□

Corollary 8.3 (Return)**.** *If the loss function $\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})$ is Lipschitz w.r.t. $\boldsymbol{\theta}$ with constant $M>0$, then, for any $\boldsymbol{\theta}_0\in\boldsymbol{\Theta}_0=\{\boldsymbol{\theta}\in\boldsymbol{\Theta}\mid\mathbb{V}_\nu(\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}))=0\}\subset\boldsymbol{\Theta}$, any $\epsilon\in(0,L^\star)$ and any $\delta\in(0,1)$, with high probability $1-\delta$ over $D\sim\nu^n$, for all $\boldsymbol{\theta}\in\boldsymbol{\Theta}$, simultaneously,*

$$\textit{if}\quad \hat{L}(D,\boldsymbol{\theta})\leq\epsilon\quad\textit{then}\quad \|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2\geq\sqrt{\tfrac{n}{8M(p\ln k-\ln\delta)}}(L^\star-\epsilon)\,.$$

Proof. If the loss is Lipschitz continuous with constant M, $\forall y,x,\boldsymbol{\theta}\quad\|\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\|_2^2\leq M$. Then, $J_{\boldsymbol{\theta}}(\lambda)$ verifies

$$\|\nabla_{\boldsymbol{\theta}}J_{\boldsymbol{\theta}}(\lambda)\|_2^2=||-\lambda\mathbb{E}_{\nu p^\lambda}\left[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\right]+\lambda\mathbb{E}_\nu\left[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\right]||_2^2\leq 2M\lambda^2\,.\tag{14}$$

where $\mathbb{E}_{\nu p^\lambda}\left[\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})\right]=\frac{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda\ell(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})]}{\mathbb{E}_\nu[p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})^\lambda]}$. With this, we have

$$|J_{\boldsymbol{\theta}}(\lambda)-J_{\boldsymbol{\theta}_0}(\lambda)|\leq 2M\lambda^2\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2\implies J_{\boldsymbol{\theta}}(\lambda)\leq 2M\lambda^2\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2\,.$$

Then, for any $a\geq 0$, we have by definition of the rate function that

$$\mathcal{I}_{\boldsymbol{\theta}}(a)\geq a\lambda-J_{\boldsymbol{\theta}}(\lambda)\geq a\lambda-2M\lambda^2\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2\,.$$

As the inequality holds for any $\lambda>0$, maximizing it raises $\lambda^\star=\frac{a}{4M\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2}$. Thus,

$$\mathcal{I}_{\boldsymbol{\theta}}(a)\geq a\lambda-J_{\boldsymbol{\theta}}(\lambda)\geq a\frac{a}{4M\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2}-2M\frac{a^2}{(4M\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2)^2}\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2=\frac{a^2}{8M\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2}\,.$$

Due to fact that $\hat{L}(D,\boldsymbol{\theta})\leq\epsilon$ and to Theorem 4.1, we have that with high probability $1-\delta$ over $D\sim\nu^n(\boldsymbol{x},\boldsymbol{y})$, $L(\boldsymbol{\theta})\leq\epsilon+\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$. Then, just rearranging terms as follows, we get $L(\boldsymbol{\theta})-\epsilon\leq\mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^p}{\delta}\right)$. Applying the rate function at both sides,

$$\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)\leq\tfrac{1}{n}\ln\tfrac{k^p}{\delta}=\frac{1}{n}\left(\ln k^p-\ln\delta\right)\,.$$

Then, we got $\ln k^p\geq n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta$ and $p\geq\frac{n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta}{\ln k}$. As $L^\star\leq L(\boldsymbol{\theta})$ and the rate function is monotonically increasing, then

$$p\geq\frac{n\mathcal{I}_{\boldsymbol{\theta}}(L(\boldsymbol{\theta})-\epsilon)+\ln\delta}{\ln k}\geq\frac{n\mathcal{I}_{\boldsymbol{\theta}}(L^\star-\epsilon)+\ln\delta}{\ln k}\,.$$

Using the upper bound over the rate function we obtained before, we got that

$$p\geq\frac{n\frac{(L^\star-\epsilon)^2}{8M\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2^2}+\ln\delta}{\ln k}\implies(L^\star-\epsilon)\sqrt{\frac{n}{8M(p\ln k-\ln\delta)}}\leq\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|_2\,.$$

□

Corollary 5.2 (Return). *Let $\boldsymbol{\Theta} \subset \boldsymbol{\Theta}'$ be two nested model classes with $p < p'$ parameters respectively. For any $\epsilon > 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, for any $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously,*

$$\hat{L}(D,\boldsymbol{\theta}') \leq \epsilon,\ \ \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \text{ and } L(\boldsymbol{\theta}') + \epsilon < L(\boldsymbol{\theta})$$
$$\Downarrow$$
$$\boldsymbol{\theta} \text{ is not } \mathcal{I}_{\boldsymbol{\theta}}^{-1}\big(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\big)\text{-smoother than } \boldsymbol{\theta}'\,.$$

Proof. We can apply Theorem 4.1 on $\boldsymbol{\theta}$ assuming that $\boldsymbol{\theta} \in \boldsymbol{\Theta}'$, because we have that $\boldsymbol{\Theta} \subset \boldsymbol{\Theta}'$. In consequence, with h.p., we have

$$L(\boldsymbol{\theta}) \leq \hat{L}(D,\boldsymbol{\theta}) + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\right)$$

Using the fact that $\hat{L}(D,\boldsymbol{\theta}) \leq \epsilon$, and that the inverse rate is bounded by the expected loss, we arrive to the following inequality

$$L(\boldsymbol{\theta}) \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\right) \leq \epsilon + L(\boldsymbol{\theta})$$

The same reasoning applies for $\boldsymbol{\theta}'$:

$$L(\boldsymbol{\theta}') \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\right) \leq \epsilon + L(\boldsymbol{\theta}')$$

Using the theorem's premise, $L(\boldsymbol{\theta}') + \epsilon < L(\boldsymbol{\theta})$, we can chain the last two h.p. upper bounds. And note that we can do that *with no extra cost*, as both apply on the same bigger model class $\boldsymbol{\Theta}'$. That is, it is the same upper bound, which holds simultaneously for all models within $\boldsymbol{\Theta}'$, used on two different models $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Then, we have,

$$\mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\right) < \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\right)\,.$$

Naming $a = \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\right)$, such that $\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta} = \mathcal{I}_{\boldsymbol{\theta}}(a)$, we got that

$$\mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\mathcal{I}_{\boldsymbol{\theta}}(a)\right) < \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\mathcal{I}_{\boldsymbol{\theta}}(a)\right) = a\,.$$

Applying $\mathcal{I}_{\boldsymbol{\theta}'}(\cdot)$ at both sides:

$$\mathcal{I}_{\boldsymbol{\theta}'}(\mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\mathcal{I}_{\boldsymbol{\theta}}(a)\right)) = \mathcal{I}_{\boldsymbol{\theta}}(a) < \mathcal{I}_{\boldsymbol{\theta}'}(a)\,.$$

Then, we can negate the inverse statement using $\neg$ notation as $\neg\Big[\mathcal{I}_{\boldsymbol{\theta}}(a) \geq \mathcal{I}_{\boldsymbol{\theta}'}(a)\Big]$. Then we have that $\boldsymbol{\theta}$ is not β-smoother than $\boldsymbol{\theta}'$ for any $\beta \geq a$. Which is equivalent to say that, $\boldsymbol{\theta}$ is not β-smoother than $\boldsymbol{\theta}'$ for any β such that $\mathcal{I}_{\boldsymbol{\theta}}(\beta) \geq \tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}$. □

Theorem 5.1 (Return). *Let $\boldsymbol{\Theta} \subset \boldsymbol{\Theta}'$ be two nested model classes with $p < p'$ parameters respectively. For any $\epsilon > 0$, with h.p. $1-\delta$ over $D \sim \nu^n$, for any $\boldsymbol{\theta}' \in \boldsymbol{\Theta}'$, $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, simultaneously*

$$\text{if } \hat{L}(D,\boldsymbol{\theta}') \leq \epsilon\,,\ \hat{L}(D,\boldsymbol{\theta}) \leq \epsilon \ \text{ and } \ \boldsymbol{\theta}' \ \text{ is } \ \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\big(\tfrac{1}{n}\ln\tfrac{k^{p'}}{\delta}\big)\text{-smoother than } \ \boldsymbol{\theta}$$
$$\Downarrow$$
$$L(\boldsymbol{\theta}') \leq L(\boldsymbol{\theta}) + \epsilon\,.$$

Proof. If $\boldsymbol{\theta}'$ is β-smoother than $\boldsymbol{\theta}$, by Definition 3, $\forall a \in (0, \beta] \quad \mathcal{I}_{\boldsymbol{\theta}'}(a) \geq \mathcal{I}_{\boldsymbol{\theta}}(a)$ where $\beta = \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right)$. Then, we have that $\mathcal{I}_{\boldsymbol{\theta}'}^{-1}(s) \leq \mathcal{I}_{\boldsymbol{\theta}}^{-1}(s)$ for $s = \mathcal{I}_{\boldsymbol{\theta}'}(\beta) = \mathcal{I}_{\boldsymbol{\theta}'}\left(\mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right)\right) = \frac{1}{n}\ln\frac{k^{p'}}{\delta}$. Thus,

$$\mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right) \leq \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right) \leq L(\boldsymbol{\theta})\,.$$

Where we used that the inverse rate is upper bounded by $L(\boldsymbol{\theta})$. By Theorem 4.1, because $\hat{L}(D, \boldsymbol{\theta}) \leq \epsilon$, we have with h.p. $1-\delta$ over $D \sim \nu^n$,

$$L(\boldsymbol{\theta}') \leq \hat{L}(D, \boldsymbol{\theta}') + \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right) \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right)\,.$$

By combining the last two inequalities, we have

$$L(\boldsymbol{\theta}') \leq \epsilon + \mathcal{I}_{\boldsymbol{\theta}'}^{-1}\left(\frac{1}{n}\ln\frac{k^{p'}}{\delta}\right) \leq \epsilon + L(\boldsymbol{\theta})\,.$$

□

References

Arora, S., Cohen, N., & Hazan, E. (2018a). On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR.

Arora, S., Ge, R., Neyshabur, B., & Zhang, Y. (2018b). Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pages 254–263. PMLR.

Arora, S., Liang, Y., & Ma, T. (2016). Why are deep nets reversible: A simple theory, with implications for training.

Bartlett, P. L., Foster, D. J., & Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30.

Bartlett, P. L., Harvey, N., Liaw, C., & Mehrabian, A. (2019). Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research*, 20(1):2285–2301.

Bartlett, P. L., Long, P. M., Lugosi, G., & Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070.

Behboodi, A., Cesa, G., & Cohen, T. S. (2022). A PAC-Bayesian generalization bound for equivariant networks. *Advances in Neural Information Processing Systems*, 35:5654–5668.

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

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

Bubeck, S., Li, Y., & Nagaraj, D. M. (2021). A law of robustness for two-layers neural networks. In *Conference on Learning Theory*, pages 804–820. PMLR.

Bubeck, S. & Sellke, M. (2023). A universal law of robustness via isoperimetry. *Journal of the ACM*, 70(2):1–18.

Casado, I., Ortega, L. A., Masegosa, A. R., & Pérez, A. (2024). PAC-Bayes-Chernoff bounds for unbounded losses. *Proceedings of the 38th Conference on Neural Information Processing Systems.*

Catoni, O. (2007). PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *Lecture Notes-Monograph Series*, 56:i–163.

Chafaï, D. (2004). Entropies, convexity, and functional inequalities, on Φ-entropies and Φ-sobolev inequalities. *Journal of Mathematics of Kyoto University*, 44(2):325–363.

Chen, S., Dobriban, E., & Lee, J. H. (2020). A group-theoretic framework for data augmentation. *The Journal of Machine Learning Research*, 21(1):9885–9955.

Chernoff, H. (1952). A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507.

Cramér, H. (1938). Sur un nouveau théoreme-limite de la théorie des probabilités. *Actual. Sci. Ind.*, 736:5–23.

Elesedy, B. (2022). Group symmetry in PAC learning. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning.*

Ellis, R. S. (2012). *Entropy, large deviations, and statistical mechanics*, volume 271. Springer Science & Business Media.

Feldman, V. & Zhang, C. (2020). What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891.

Gastpar, M., Nachum, I., Shafer, J., & Weinberger, T. (2024). Fantastic generalization measures are nowhere to be found. *International Conference on Learning Representations.*

Gilbert, A. C., Zhang, Y., Lee, K., Zhang, Y., & Lee, H. (2017). Towards understanding the invertibility of convolutional neural networks. pages 1703–1710.

Golowich, N., Rakhlin, A., & Shamir, O. (2018). Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*, volume 1. MIT press.

Goodfellow, I., Lee, H., Le, Q., Saxe, A., & Ng, A. (2009). Measuring invariances in deep networks. *Advances in neural information processing systems*, 22.

Gouk, H., Frank, E., Pfahringer, B., & Cree, M. J. (2021). Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416.

Gunasekar, S., Lee, J. D., Soudry, D., & Srebro, N. (2018). Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31.

Hardt, M. & Ma, T. (2017). Identity matters in deep learning. *International Conference on Learning Representations.*

Herdegen, M. (2008). The theorem of bahadur and rao and large portfolio losses. *Journal of Applied Mathematics*, 2011.

Hu, W., Li, Z., & Yu, D. (2020). Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *International Conference on Learning Representations.*

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., & Bengio, S. (2020). Fantastic generalization measures and where to find them. *International Conference on Learning Representations.*

Kawaguchi, K., Kaelbling, L. P., & Bengio, Y. (2022). Generalization in deep learning. In *Mathematical Aspects of Deep Learning.* Cambridge University Press.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2017a). On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations.*

Keskar, N. S., Nocedal, J., Tang, P., Mudigere, D., & Smelyanskiy, M. (2017b). On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of ICLR 2017.*

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, X. & Orabona, F. (2019). On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd international conference on artificial intelligence and statistics*, pages 983–992. PMLR.

Liang, T., Poggio, T., Rakhlin, A., & Stokes, J. (2019). Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics*, pages 888–896. PMLR.

Livni, R., Shalev-Shwartz, S., & Shamir, O. (2014). On the computational efficiency of training neural networks. *Advances in neural information processing systems*, 27.

Lyle, C., van der Wilk, M., Kwiatkowska, M., Gal, Y., & Bloem-Reddy, B. (2020). On the benefits of invariance in neural networks. *arXiv preprint arXiv:2005.00178*.

McAllester, D. A. (1999). PAC-Bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Nagarajan, V. (2021). Explaining generalization in deep learning: progress and fundamental limits. *arXiv preprint arXiv:2110.08922*.

Nagarajan, V. & Kolter, J. Z. (2017). Generalization in deep networks: The role of distance from initialization. *NeurIPS Workshop on Deep Learning: Bridging Theory and Practice*.

Nagarajan, V. & Kolter, J. Z. (2019a). Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience. *International Conference on Learning Representations*.

Nagarajan, V. & Kolter, J. Z. (2019b). Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32.

Negrea, J., Dziugaite, G. K., & Roy, D. (2020). In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors. In *International Conference on Machine Learning*, pages 7263–7272. PMLR.

Nesterov, Y. (2003). *Introductory Lectures on Convex Optimization: A Basic Course*. Springer.

Neyshabur, B., Bhojanapalli, S., McAllester, D., & Srebro, N. (2017a). Exploring generalization in deep learning. *Advances in neural information processing systems*, 30.

Neyshabur, B., Bhojanapalli, S., & Srebro, N. (2017b). A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations*.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., & Srebro, N. (2019). The role of over-parametrization in generalization of neural networks.

Neyshabur, B., Salakhutdinov, R. R., & Srebro, N. (2015). Path-sgd: Path-normalized optimization in deep neural networks. *Advances in neural information processing systems*, 28.

Patel, A. B., Nguyen, M. T., & Baraniuk, R. (2016). A probabilistic framework for deep learning. *Advances in neural information processing systems*, 29.

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

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press, Princeton.

Schoenholz, S. S., Gilmer, J., Ganguli, S., & Sohl-Dickstein, J. (2017). Deep information propagation. In *International Conference on Learning Representations.*

Shalev-Shwartz, S. & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

Shawe-Taylor, J., Bartlett, P. L., Williamson, R. C., & Anthony, M. (1998). Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, 44(5):1926–1940.

Shorten, C. & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Sokolic, J., Giryes, R., Sapiro, G., & Rodrigues, M. (2017). Generalization error of invariant classifiers. In *Artificial Intelligence and Statistics*, pages 1094–1103. PMLR.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., & Srebro, N. (2018). The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Tishby, N. & Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE.

Vapnik, V. N. & Chervonenkis, A. Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity: festschrift for alexey chervonenkis*, pages 11–30. Springer.

Vidal, R., Bruna, J., Giryes, R., & Soatto, S. (2020). Mathematics of deep learning. *Mexican Conference on Pattern Recognition.*

Wang, Y., Sonthalia, R., & Hu, W. (2024). Near-interpolators: Rapid norm growth and the trade-off between interpolation and generalization. *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 238:4483–4491.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations.*

Zhang, T. (2006). Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., & Orbanz, P. (2019). Non-vacuous generalization bounds at the imagenet scale: a PAC-Bayesian compression approach. In *International Conference on Learning Representations.*