Does Compression Imply Generalization? A Minimum Description Length Perspective

Lincen Yang[1] **Jiayang Shi**[2,1] **Nan Pu**[1,3] **Daniël M. Pelt**[1]
Matthijs van Leeuwen[1] **Zhong Li**[4,1,*] **Shujian Yu**[5,6,*]

[1]Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
[2]Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands
[3]Hefei University of Technology, Hefei, China
[4]Great Bay University, Dongguan, China
[5]Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands
[6]Department of Physics and Technology, UiT The Arctic University of Norway

{l.yang, j.shi, n.pu, d.m.pelt, m.van.leeuwen, z.li}@liacs.leidenuniv.nl
yusj9011@gmail.com

Abstract

A central question in modern learning theory is what notion of compression, if any, explains why over-parameterized neural networks generalize. Existing answers measure complexity at different levels, ranging from weights to representations; however, estimating the associated quantities either requires held-out data in principle or relies on the challenging (ultra) high-dimensional mutual information estimation. To overcome these challenges, we approach the question through the minimum description length (MDL) principle, and specifically, we propose the prequential regret as a complementary signal that is computable from the training trajectory alone. We establish that, in the (nearly) interpolating regime, the expected prequential regret upper-bounds the description length of the trained parameter under the algorithm-induced distribution, and that this description length translates into a high-probability generalization bound in the PAC-MDL setting. Along the way, we show that the encoder-complexity correction recently added to the algorithm-level information bottleneck (IB) is upper-bounded by an MDL parameter description length, bringing together the two views of compression from MDL and IB. Empirically, the prequential regret tracks the generalization error on CIFAR-10 and CIFAR-100, with Pearson correlations exceeding 0.9.

1 Introduction

A fundamental question in statistical learning theory is why over-parameterized models generalize to unseen data, despite having enough capacity to memorize their training sets [45, 6]. Classical theory measures complexity at the level of the model class, through quantities such as VC dimension [41], Rademacher complexity [3], or norm-based capacity [4]. By contrast, a modern line of work shifts attention to what the learned model actually retains from the training data [10, 33, 42]. The complexity of the learned model is then characterized through information-theoretic quantities, and the central question becomes how to formalize and measure this complexity.

Three mainstream approaches make this complexity precise as a *coding cost*. First, PAC-Bayes [28, 15] bounds generalization by $D_{\mathrm{KL}}(Q\|\pi)$ between a posterior Q over learned models and a prior π, which by the bits-back argument [23, 24] is the code length of transmitting a sample from Q induced

*Corresponding authors: Zhong Li (z.li@liacs.leidenuniv.nl) and Shujian Yu (yusj9011@gmail.com).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

by π [16]. Second, the information-theoretic line [33, 42] replaces the prior with the data-marginal and bounds generalization by $I(S;W)$, measuring how much information the weights W retain about the training set S. Since $I(S;W)$ can be infinite for deterministic algorithms, the supersample/CMI variants [38, 19, 20] introduce a pooled sample $\tilde{Z}$ of $2n$ points together with a uniform selector U that assigns half to training and half to a 'ghost' test set, and bound generalization by $I(W;U \mid \tilde{Z})$, capturing how much the weights leak about the train/test assignment given the pool. Third, the information bottleneck (IB) [40, 39, 37] asks how much the internal representations Z retain about X, with $I(X;Z)$ or $I(X;Z \mid Y)$ [40] as the coding cost of the representation. Recent work observes that vanilla IB implicitly assumes a fixed encoder and can fail when the encoder, i.e., the map from an input to its internal representation, is itself learned from data [18]; Kawaguchi et al. [25] hence addresses this by extending IB to the algorithm level, adding an encoder-complexity term that penalizes how much the trained encoder depends on the training set.

While these three lines capture different aspects of compression, they have the following shortcomings when we want a compression signal directly tied to fixed algorithms for training networks. On the conceptual side, what is being compressed is different across the literature: PAC-Bayes and the $I(S;W)$ line compress at the level of weights, while IB compresses at the level of representations, and the two notions have evolved separately, leaving it unclear whether they are facets of one underlying property or distinct phenomena that share a name. Further, even granting a notion of compression, whether Occam-style reasoning survives for over-parameterized neural networks is itself debated [45, 36, 30].

On the practical side, the existing complexity measures are hard to estimate or evaluate in deep learning regimes. Specifically, IB and $I(S;W)$ require estimating mutual information for ultra-high-dimensional feature vectors X or weight parameters W, the supersample/CMI variant requires a held-out twin sample, and PAC-Bayes requires a prior that captures the complex dependency among parameters. Thus, a complementary signal computable from the training set, with no extra data and no high-dimensional MI estimation, would shed light on generalization from the inside of training.

This motivates us to consider quantifying the model complexity from the alternative perspective of the minimum description length (MDL) [32, 16, 17], which provides the tools to link the complexity of a model via the description length (in bits) needed to encode the model itself in a lossless manner. Specifically, to resolve the aforementioned shortcomings, *we propose to use the prequential coding regret (defined in Section 2) as an operational generalization bound, together with both theoretical justifications and empirical validations*.

Contributions. We summarize our main contributions as follows.

- We connect MDL parameter compression to the algorithm-level information bottleneck, showing that the encoder-complexity correction added to obtain the first rigorous IB-based generalization bound is upper-bounded by an MDL parameter description length, which relates the two views of compression from IB and MDL (Section 3).
- We prove that, in the (nearly) interpolating regime, the expected prequential coding regret upper-bounds the description length of the trained parameter under the algorithm-induced distribution over parameters given the data (Section 4).
- We further show that this description length translates into a generalization bound in the PAC-MDL setting, with the algorithm-dependent content carried entirely by the prequential regret. We also empirically demonstrate that the regret tracks the generalization gap on CIFAR-10 and CIFAR-100 with Pearson correlations of 0.9 or higher (Sections 5 and 6).

Related work. Although MDL was incorporated into neural networks as early as the 1990s [23], the MDL-based model complexity of neural networks and its connection to generalization remain understudied, with a few exceptions as follows. First, the PAC-MDL bound [8] connects the description length of a model to its generalization, yet, as in the PAC-Bayesian framework, leaves open how to quantify this description length. Second, Blier and Ollivier [7] empirically demonstrated that the prequential code can be used to effectively compress the data and the neural network together, and argued that the description length of the trained neural network was 'contained' in the total code length; however, quantifying the description length of the model itself remains unclear. Third, recent theoretical work connects MDL with generalization bounds [34, 35], but the "MDL" there is defined as an expected KL divergence between the posterior of the latent representation and a symmetric conditional prior; while conceptually related, this does not directly measure the description length

of the model. Fourth, the excess description length [14], defined as the prequential code length minus the negative log-likelihood on a *test* set, is conceptually close to the prequential regret but, by depending on the test set, does not measure model complexity from the training side alone.

Organization. The remainder of the paper is organized as follows. Section 2 introduces the MDL principle and the prequential code, from which the prequential regret is defined. Section 3 reviews the recent correction to the information bottleneck and connects it to MDL, linking the two notions of compression. Section 4 makes precise what the complexity of a trained network means and proves that it is bounded by the prequential regret, a quantity that can be computed from the training run alone. Section 5 shows that this complexity leads to generalization guarantees, and Section 6 tests empirically whether the regret tracks the generalization gap.

2 Preliminary: The MDL Principle

The minimum description length (MDL) principle, rooted in information theory, formulates the task of learning a model from data as a compression problem. That is, given a model class, the "quality" of a model is evaluated as how much it can compress the data, measured by the cost of encoding the model itself, plus the cost of encoding the data given the model [32, 16]; the best model is hence the one that achieves the highest compression in this sense. It can be viewed as a practical approximation of Kolmogorov complexity [27] and a formalization of Occam's razor [9].

We next briefly review the elements needed for our analysis. By Kraft's inequality [16], prefix-free codes correspond to sub-probability distributions: any code with lengths $\ell(x)$ satisfies $\sum_x 2^{-\ell(x)} \leq 1$, and conversely any distribution Q defines a code of length $-\log Q(x)$, optimal up to one bit [12]. A single distribution thus determines a code.

In the MDL literature, however, a *model* refers not to a single distribution but to a family of probability distributions indexed by a parameter $\theta \in \Theta$, i.e., $M = \{Q_\theta\}_{\theta\in\Theta}$ [16]. Encoding data under such a model is more delicate than the single-distribution case: the Kraft inequality applies to individual distributions, not to families, so the family alone does not determine a code. A *universal code* [17] resolves this by inducing a single distribution over outcomes from the entire family, in such a way that its code length is nearly as short as $-\log Q_{\hat{\theta}}(y^m \mid x^m)$, the code length of the best-fitting parameter $\hat{\theta}$ that corresponds to the shortest code length in hindsight (i.e., after seeing the data).

Two universal codes are central to our analysis: the *two-part code* and the *prequential code*, which we describe next. In both cases, we frame the encoding scheme as a protocol in which Alice transmits the labels y^m to Bob, who has access to the inputs x^m.

2.1 The two-part encoding scheme

The *two-part code* [16] is the most direct realization of the MDL principle: it splits the description of the data into one part for the model and one for the data given the model. Alice first transmits a parameter $\theta \in \Theta$ using some code with length $L(\theta)$, then transmits the labels using Q_θ at cost $-\log Q_\theta(y^m \mid x^m)$ bits, choosing θ to minimize the total.

Definition 2.1 (Two-part code length)**.** Given a code $L(\theta)$ over the parameter space Θ, the *two-part code length* is

$$L_{2\mathrm{p}}(y^m \mid x^m) \;=\; \min_{\theta\in\Theta} \Big\{\; L(\theta) \;-\; \log Q_\theta(y^m \mid x^m) \;\Big\}. \tag{1}$$

The first term acts as a complexity penalty: parameters that require more bits to describe must yield correspondingly larger reductions in the data term to be selected. Despite its conceptual clarity, the two-part code is generally suboptimal among universal codes, and designing $L(\theta)$ for high-dimensional models such as neural networks requires discretizing Θ to some precision, with the choice itself affecting the trade-off [16]. The prequential code, described next, sidesteps both issues by avoiding any explicit encoding of θ.

2.2 The prequential encoding scheme

The prequential code encodes the data sequentially [13, 16], using at each step a predictive distribution derived from the data seen so far; Alice and Bob together transmit the labels of the training set while implicitly transmitting the learned model.

Alice and Bob share, prior to any communication, the network architecture, the initialization θ_0, the optimizer and its hyperparameters, and the unlabeled data $\{x_i\}_{i=1}^m$. Here $\theta \in \Theta$ are the network weights and $Q_\theta(y \mid x)$ is the network's predictive (softmax) distribution, with $Q_\theta(y^m \mid x^m) = \prod_i Q_\theta(y_i \mid x_i)$. Let $B_1, \ldots, B_K$ be a partition of $\{1, \ldots, m\}$ into K consecutive blocks; the case $K = m$ recovers the classical one-by-one prequential code, while coarser partitions are typically used for computational efficiency [7].

Definition 2.2 (Prequential code length)**.** Let $\hat{\theta}_t$ be the model obtained by running the shared training procedure on $\bigcup_{k=1}^t B_k$, with $\hat{\theta}_0 = \theta_0$. The *prequential code length* is

$$L_{\mathrm{preq}}(y^m \mid x^m) \;=\; \sum_{t=0}^{K-1} \sum_{i \in B_{t+1}} -\log Q_{\hat{\theta}_t}(y_i \mid x_i), \tag{2}$$

i.e., the labels in block B_{t+1} are encoded under the model trained on blocks $B_1, \ldots, B_t$. We write $Q_{\mathrm{preq}}(y^m \mid x^m) := \prod_{t=0}^{K-1} \prod_{i \in B_{t+1}} Q_{\hat{\theta}_t}(y_i \mid x_i)$ for the prequential predictive distribution, so that $L_{\mathrm{preq}} = -\log Q_{\mathrm{preq}}(y^m \mid x^m)$.

No parameters are ever transmitted: Bob can replicate each $\hat{\theta}_t$ from the shared protocol and the previously decoded labels, so the model is communicated implicitly through the sequence of predictive distributions. Once Bob has received y^m, he can reconstruct the parameters θ^* obtained by training on the full set (x^m, y^m), achieving the in-hindsight code length $\hat{L}(y^m \mid x^m) = -\log Q_{\theta^*}(y^m \mid x^m)$. The gap between the prequential code length and this code length is central to our analysis.

Definition 2.3 (Regret)**.** The *regret* of the prequential code is $R := L_{\mathrm{preq}}(y^m \mid x^m) - \hat{L}(y^m \mid x^m)$.

Intuitively, R counts the extra bits incurred by encoding the parameters jointly with the data, and plays the role of a model-complexity penalty analogous to $L(\theta)$ in the two-part code.

Notably, in classical universal-coding settings with the maximum likelihood estimator (MLE), $R \geq 0$ holds pointwise [16] (to avoid 0 probability and hence infinite log, a small pseudo-count can be added to each class before any data is seen). In our setting where $\hat{\theta}_t$ is produced by a training procedure rather than an MLE predictor, $R \geq 0$ should hold with any reasonable training procedure, since the final model is trained on the full dataset and should fit the labels at least as well as the intermediate models trained on prefixes. Thus, we assume $R \geq 0$ throughout the theoretical analysis.

3 Compression of Parameters and the Modern Information Bottleneck

Compression and generalization of deep neural networks have been studied rarely through MDL, yet widely through the information bottleneck (IB). Before developing our MDL-based analysis, we therefore clarify that MDL does not use "compression" in an unrelated sense, and why it is a natural perspective closely related to IB, even though the concepts of compression in MDL and IB seem to be different.

Specifically, the IB places compression at the level of representations, calling $Z = \phi_l(X)$ compressive when $I(X; Z)$ or $I(X; Z \mid Y)$ is small, while MDL (Section 2) places it at the level of parameters, calling a model compressive when its weights admit a short code. We show that the two notions meet at a single quantity once the encoder is learned from the same sample whose representation it encodes. The in-sample $I(X^m; Z^m)$ decomposes into a fixed-encoder IB term plus the encoder-complexity correction that the modern IB requires [25], and the correction is upper-bounded by an MDL parameter description length, the quantity we operationalize in Section 4.

The in-sample IB and its decomposition. Let $(X^m, Y^m) \sim P$ be a training sample, and let a (possibly randomized) learning algorithm produce an encoder ϕ_l, the map from an input to its layer-l representation, via $T(\phi_l \mid X^m, Y^m)$; for a deterministic algorithm this is a point mass. Write T for the induced joint on (X^m, Y^m, ϕ_l), and $Z^m = \phi_l(X^m)$ for the representations of the training inputs under the trained encoder.

The classical IB studies $I(X; Z)$ for an encoder fixed in advance. The quantity actually measured in practice, e.g., when IB-style mutual information is used as a regularizer in deep networks [2, 1], is $I(X^m; Z^m)$ under the joint law T, in which ϕ_l depends on X^m. The classical criterion is known to fail in this regime: Hafez-Kolahi et al. [18] exhibit an encoder that memorizes how X maps to Y on

the training set yet still appears compressive by the classical criterion. The criterion ignores how ϕ_l depends on the sample, and the chain rule on $(X^m; Z^m, \phi_l)$ makes the missing term explicit.

Proposition 3.1. *Let $(X^m, Y^m, \phi_l) \sim T$ as above, and Z^m a measurable function of (X^m, ϕ_l). Then*

$$I(X^m; Z^m) \;=\; I(X^m; Z^m \mid \phi_l) \;+\; I(X^m; \phi_l) \;-\; I(X^m; \phi_l \mid Z^m), \tag{3}$$

and consequently

$$I(X^m; Z^m) \;\le\; I(X^m; Z^m \mid \phi_l) \;+\; I(X^m; \phi_l). \tag{4}$$

All proofs are deferred to Appendix A.

We interpret the two terms as follows. The first term, $I(X^m; Z^m \mid \phi_l)$, is the fixed-encoder IB averaged over the algorithm-induced encoder law. It reduces to the classical IB exactly when ϕ_l is constant. The second term, $I(X^m; \phi_l)$, is an encoder-complexity correction. It measures how much the trained encoder varies with the training data, and vanishes whenever the algorithm returns the same ϕ_l on every sample.

From encoder complexity to parameter description length. We next show that the encoder-complexity correction is bounded by an MDL parameter description length through a chain of elementary inequalities. Suppose ϕ_l is realized by a parameter vector θ, and assume θ takes values in a discrete set; i.e., in practice, the quantization implicit in any finite-precision implementation. Then

$$I(X^m; \phi_l) \;\le\; H(\phi_l) \;\le\; H(\theta) \;\le\; \mathbb{E}[L(\theta)], \tag{5}$$

where L is a prefix-free code on the parameter space. The first inequality is the discrete-MI bound, the second is data processing as ϕ_l is a function of θ, and the third is Shannon's source coding theorem.

A related encoder-complexity term $I(\phi_l; S)$, in which S is the training set and ϕ_l is the encoder learned from it, appears in Kawaguchi et al. [25], who add it to the classical IB bound to prove the first rigorous IB-based generalization guarantee. Our $I(X^m; \phi_l)$ is bounded by $I(\phi_l; S)$, and the parameter description length in (5) in turn bounds $I(\phi_l; S)$. The IB correction needed for learned encoders is bounded by a parameter description length, so a model with parameters admitting a short code in the MDL sense is also compressive in the modern IB sense. The remainder of the paper takes up this parameter-level quantity directly, showing how it can be measured from the training trajectory alone (Section 4) and how it translates into a generalization bound (Section 5).

4 Prequential Regret Bounds the Description Length of Neural Networks

This section formalizes the intuition that the prequential regret contains the description length of the trained network. We begin with the conceptual intuition, then introduce the two probability spaces needed to make it precise, and state the formal connection in Theorem 4.1.

The conceptual gap. The intuition starts from how the prequential encoding scheme can be interpreted. Consider the message transmission between Alice and Bob again: the total cost L_{preq} is the number of bits Alice spends, and by the time Bob has decoded the labels he can replay the training procedure and recover the same parameters Alice produced; he ends up holding both data and model.

Thus, informally, one could argue that the description length of the model is intuitively "contained" in the prequential code, from which one can decode both the model and the data. Splitting this total into "bits for the data" and "bits for the model", the data part is well-approximated by $\hat{L}$, the in-hindsight code length under the trained model, and the model part is what remains, namely the regret $L_{\mathrm{preq}} - \hat{L}$. Specifically, this intuition was articulated by Blier and Ollivier [7] as motivation for using prequential codes to argue Occam's razor for neural networks, in which they argue that the description length of the model is 'contained' in L_{preq}.

However, this conceptual intuition is heuristic, as the prequential code is literally a code on the data, not on the parameters: every bit of L_{preq} is allocated to encoding a label, not to encoding θ. Yet Bob recovers θ at the end, so the parameter information must be in those bits somewhere, just not in a designated parameter section of the code. Whether the heuristic holds rigorously, and what description-length quantity it formally upper-bounds, has been left open.

Setup. Making the conceptual intuition precise requires two probability spaces: 1) the model's predictive distribution Q_θ, which assigns likelihoods to data given parameters, and 2) the joint

distribution T over data and trained parameters, induced by the data-generating process and the algorithm $\mathcal{A}$ together.

We consider a training set (X^m, Y^m) drawn from P and write $P(Y^m \mid X^m)$ for the conditional data-generating distribution. A learning algorithm $\mathcal{A}$ run on (X^m, Y^m) produces a parameter $\theta(X^m, Y^m)$, random under the algorithm's internal randomness U; we write $T(\theta \mid X^m, Y^m)$ for the resulting conditional distribution. Marginalizing over labels gives $T(\theta \mid X^m) = \sum_{y^m} T(\theta \mid X^m, y^m)\, P(y^m \mid X^m)$, the distribution over trained parameters when the inputs are fixed and the labels (together with the algorithm's randomness) vary; $T(Y^m \mid X^m, \theta)$ denotes the corresponding reverse conditional, the posterior over label sequences given the trained parameter. We further consider the parametric family $\{Q_\theta\}_{\theta \in \Theta}$ defining the model and write $Q_{\mathrm{preq}}(\cdot \mid X^m)$ for the prequential predictive distribution induced by $\mathcal{A}$.

Notably, X^m is treated as shared side information in the coding protocol throughout this section, as is common practice in supervised MDL [43, 31, 44], in which the goal is to learn $P(Y \mid X)$ rather than $P(X, Y)$. Thus, the cost of encoding X^m does not contribute to the code length attributable to learning, and we condition on X^m throughout. We keep this conditioning implicit when convenient. Section 5 relaxes this in the PAC-MDL setting.

We next present our main theoretical results that the prequential regret can upper-bound the parameter description length.

Theorem 4.1 (Prequential regret upper-bounds parameter description length)**.** *Let $\theta = \theta(X^m, Y^m)$ denote the random parameter produced by training, $\hat{L} = -\log Q_\theta(Y^m \mid X^m)$, and $L_{\mathrm{preq}} = -\log Q_{\mathrm{preq}}(Y^m \mid X^m)$. Suppose that, pointwise,*

$$Q_\theta(Y^m \mid X^m) \;\geq\; T(Y^m \mid X^m, \theta). \tag{6}$$

Then the following two statements hold.

(i) (Pointwise.) For every (X^m, Y^m),

$$L_{\mathrm{preq}} - \hat{L} \;\geq\; \underbrace{-\log T(\theta \mid X^m) + \log T(\theta \mid X^m, Y^m)}_{\text{Description length of } \theta \text{ for fixed } X^m, Y^m} \;+\; \log \frac{P(Y^m \mid X^m)}{Q_{\mathrm{preq}}(Y^m \mid X^m)}. \tag{7}$$

(ii) (In expectation.) Taking expectations under the joint distribution over (X^m, Y^m, θ),

$$\mathbb{E}[L_{\mathrm{preq}} - \hat{L}] \;\geq\; \underbrace{H_T(\theta \mid X^m) - H_T(\theta \mid X^m, Y^m)}_{\text{Expected description length of } \theta} \;+\; \mathbb{E}_{X^m}\big[\mathrm{KL}\big(P(\cdot \mid X^m) \,\|\, Q_{\mathrm{preq}}(\cdot \mid X^m)\big)\big], \tag{8}$$

where both terms on the right are non-negative.

The interpretation of the bound depends on whether training is deterministic or random given the data, as we describe in detail below.

Deterministic training. If the algorithm is fully deterministic given (X^m, Y^m), e.g., with a fixed seed controlling batch order, dropout, etc., then θ is a function of the training set, $T(\theta \mid X^m, Y^m)$ is a point mass, and $H_T(\theta \mid X^m, Y^m) = 0$. Inequality (8) simplifies to

$$\mathbb{E}[L_{\mathrm{preq}} - \hat{L}] \;\geq\; H_T(\theta \mid X^m) + \mathbb{E}_{X^m}[\mathrm{KL}(P \| Q_{\mathrm{preq}})] \;\geq\; \mathbb{E}[-\log T(\theta \mid X^m)], \tag{9}$$

i.e., the expected model complexity $\mathbb{E}[R] = \mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$ is at least the conditional self-information of the trained parameters under their marginal $T(\theta \mid X^m)$. Thus, this interpretation reveals our key finding that *prequential regret "behaves like" a parameter code length*, with description length measured against the algorithm's induced distribution over trained parameters at the given inputs.

Stochastic training and the bits-back interpretation. When training retains randomness (random seeds, stochastic batching, dropout, etc.), $H_T(\theta \mid X^m, Y^m) > 0$ and the residual $H_T(\theta \mid X^m) - H_T(\theta \mid X^m, Y^m) = I_T(Y^m; \theta \mid X^m)$ takes its place on the right-hand side.

Notably, $I_T(Y^m; \theta \mid X^m)$ can be interpreted with the *bits-back* coding argument [23, 24]: when Alice transmits a stochastic parameter $\theta \sim T(\theta \mid X^m, Y^m)$, she nominally pays $H_T(\theta \mid X^m)$ bits, but Bob, given θ and the protocol, can recover the random bits used to sample θ from $T(\theta \mid X^m, Y^m)$

and reclaim $H_T(\theta \mid X^m, Y^m)$ of them. The net description length of θ under this stochastic protocol is therefore $I_T(Y^m; \theta \mid X^m)$, exactly the term appearing in (8). Inequality (8) thus extends the deterministic statement to the stochastic setting.

Remark 4.2. The condition $Q_\theta(Y^m|X^m) \geq T(Y^m|X^m, \theta)$ can be interpreted as follows. The LHS is the likelihood of the training labels under the trained model, which is large in the nearly interpolating regime; the RHS is the posterior probability of these labels among all label sequences that the algorithm could have mapped to θ. The condition holds whenever the latter is the smaller of the two. This is the case in typical many-to-one training, where many label sequences are mapped to the same parameter, so that θ leaves substantial uncertainty about Y^m; it fails when the model underfits, so that the LHS is small, or under strong memorization, where θ all but uniquely identifies Y^m and pushes $T(Y^m \mid X^m, \theta)$ close to one. We hence refer to this condition as the *interpolation-genericity* condition. Direct verification is unavailable since $T(\theta \mid X^m, Y^m)$ has no closed form for nontrivial deep learning algorithms, so we test the theorem's operational prediction in Section 6, i.e., $L_{\text{preq}} - \hat{L}$ tracks generalization.

In summary, the intuition that the prequential code length contains a description of the trained model, used informally by Blier and Ollivier [7], becomes precise once two probability spaces are distinguished. The predictive distribution Q_θ governs how the trained model assigns likelihoods to data, while the algorithm-induced distribution $T(\theta \mid X^m)$ governs how the trained parameters themselves vary under the data-generating process and the training algorithm. The heuristic interpretation cannot say which description-length quantity is bounded by the regret, because it does not separate these two spaces. Theorem 4.1 shows that the relevant quantity is the description length of θ under $T(\theta \mid X^m)$, the algorithm's induced distribution over trained parameters at the given inputs.

5 From Prequential Regret to Generalization

The previous section established that the expected prequential regret $\mathbb{E}[L_{\text{preq}} - \hat{L}]$ operationally upper-bounds the description length of the trained parameter under the algorithm's induced distribution $T(\theta \mid X^m)$. We now leverage this description length to prove a generalization guarantee. For deterministic algorithms we base our theoretical results on the PAC-MDL bound of Blum and Langford [8], which converts any sub-probability prior over description methods into a high-probability bound on test error; for stochastic algorithms we extend the argument to the transductive PAC-Bayes setting of Bégin et al. [5], which accommodates the bits-back argument as we identified in Section 4. We first recall the PAC-MDL bound in its standard form.

Theorem 5.1 (PAC-MDL bound; Blum and Langford [8], Theorem 6)**.** *Let D be a distribution over $\mathcal{X} \times \mathcal{Y}$. Draw $m+n$ fixed labeled examples $(x, y)^{m+n} \sim D^{m+n}$ and split them uniformly at random into a training set S of size m and a test set S' of size n. A* description method *σ is a binary string that, together with the unlabeled data x^{m+n}, determines a labeling y_σ of all $m+n$ examples; write $\hat{e}(\sigma, S)$ and $\hat{e}(\sigma, S')$ for the error rates of y_σ on the training and test sets, respectively. Let $P(\sigma) \geq 0$ be any* prior *over description methods satisfying $\sum_\sigma P(\sigma) \leq 1$. Then with probability at least $1-\delta$ over the draw of $(x, y)^{m+n}$ and the random partition into S, S':*

$$\forall \sigma \quad n\hat{e}(\sigma, S') \;\leq\; b_{\max}\big(n,\; \hat{e}(\sigma, S),\; P(\sigma)\delta\big),$$

where $b_{\max}(n, a, \beta)$ is the largest integer b such that $\text{Bucket}(m, n, a, b) \geq \beta$, with $\text{Bucket}(m, n, a, b) = \sum_{t=b}^{a+b} \binom{n}{t}\binom{m}{a+b-t} / \binom{m+n}{a+b}$.

Remark 5.2. Theorem 5.1 controls how much the test error $\hat{e}(\sigma, S')$ can exceed the training error $\hat{e}(\sigma, S)$ as a function of the description length $-\log P(\sigma)$. Shorter descriptions yield tighter bounds, since $P(\sigma)\delta$ is larger and the right-hand side is smaller.

The key feature of Theorem 5.1 is that the prior $P(\sigma)$ is unconstrained beyond being a sub-probability, and can depend on objects fixed in advance of the random partition into S, S', including the unlabeled sample x^{m+n} shared between Alice and Bob.

However, the algorithm-induced distribution $T(\theta \mid X^m)$ is partition-dependent, since X^m is determined by which subset of x^{m+n} ends up in the training partition. We next show that we resolve this challenge by treating the description method as a pair (π, θ) specifying both the partition and the trained parameter, with the partition encoded uniformly. As a result, the resulting joint prior is partition-invariant by construction.

5.1 Deterministic algorithms

We next apply the joint prior to the PAC-MDL bound and present the generalization bound for deterministic algorithms.

Theorem 5.3 (Generalization bound via prequential regret, deterministic algorithms)**.** *Let $\mathcal{A}$ be a deterministic algorithm, and consider the PAC-MDL setup of Theorem 5.1: draw $m+n$ labeled examples $(x,y)^{m+n} \sim P^{\otimes(m+n)}$ and split them uniformly at random into a training set S of size m and a test set S' of size n. Let $\theta = \mathcal{A}(S)$ denote the parameter produced by training on S, with induced conditional distribution $T(\theta \mid X^m)$ over θ given the training inputs. Suppose the interpolation-genericity condition* (6) *holds and θ takes values in a discrete (finite-precision) parameter space. For any $\delta \in (0,1)$, with probability at least $1-\delta$ over the joint draw of the sample and the partition,*

$$n\hat{e}(\theta, S') \;\le\; b_{\max}\left(n,\; \hat{e}(\theta, S),\; \frac{\delta}{2\binom{m+n}{m}} \cdot 2^{-2\,\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]/\delta}\right), \tag{10}$$

where L_{preq} and $\hat{L}$ are the prequential and in-hindsight code lengths of Section 2 evaluated on the training partition. $\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$ is taken over the joint draw of $(x,y)^{m+n}$ and the random partition.

Remark 5.4. The proof (Appendix A) combines a PAC-MDL bound at confidence $1-\delta/2$, using the partition-invariant prior $P(\pi,\theta) = T(\theta|X^m_\pi)/\binom{m+n}{m}$, together with a Markov inequality on the description length at confidence $1-\delta/2$. The key step is the expectation chain $\mathbb{E}[-\log T(\theta|X^m_{\pi^*})] = H_T(\theta|X^m) \le \mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$, used in the Markov step: the algorithm-dependent factor in the bound is exactly the prequential regret, which can be computed directly from the training set.

5.2 Stochastic algorithms

Theorem 5.3 assumes $\mathcal{A}$ is deterministic, so $T(\theta \mid X^m, Y^m)$ is a Dirac mass and the description length $-\log T(\theta^* \mid X^m_{\pi^*})$ of the realized parameter is bounded by the prequential regret in expectation. For stochastic $\mathcal{A}$ this bound no longer holds, as the posterior $T(\theta \mid X^m, Y^m)$ is non-degenerate, and

$$\mathbb{E}\big[-\log T(\theta^* \mid X^m)\big] \;=\; H_T(\theta \mid X^m) \;=\; I_T(\theta; Y^m \mid X^m) + H_T(\theta \mid X^m, Y^m),$$

which can exceed $\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$ by up to the algorithm's intrinsic conditional entropy $H_T(\theta \mid X^m, Y^m)$.

Therefore, applying PAC-MDL via Markov as in Theorem 5.3 yields a bound with irreducible slack: the framework charges $-\log P(\sigma)$ at the realized σ pointwise, and does not account for the bits-back term that prequential coding extracts in Section 4.

We close the gap by replacing PAC-MDL with its PAC-Bayes counterpart [5] in the same transductive setup. With $T(\theta \mid X^m, Y^m)$ playing the role of the data-dependent posterior and $T(\theta \mid X^m)$ the data-independent prior, the complexity term becomes $\mathrm{KL}\big(T(\theta \mid X^m, Y^m) \,\|\, T(\theta \mid X^m)\big)$, whose expectation is then $I_T(\theta; Y^m \mid X^m)$, i.e., the bits-back-tight quantity.

The cost is that the bound now controls the Gibbs test error $\mathbb{E}_{\theta\sim T(\cdot|X^m,Y^m)}[\hat{e}(\theta, S\cup S')]$ rather than the error of a single sampled θ^*, which is the natural object for a stochastic predictor and reduces to the point error in the deterministic limit.

Theorem 5.5 (Generalization bound via prequential regret, stochastic algorithms)**.** *Let $\mathcal{A}$ be a (possibly stochastic) learning algorithm with induced conditional distribution $T(\theta \mid X^m, Y^m)$. Consider the PAC-MDL setup of Theorem 5.1: draw $m+n$ labeled examples $(x,y)^{m+n} \sim P^{\otimes(m+n)}$ and split uniformly at random into a training set S of size m and a test set S' of size n. Suppose the interpolation-genericity condition* (6) *holds and θ takes values in a discrete (finite-precision) parameter space. For any $\delta \in (0,1)$, with probability at least $1-\delta$ over the joint draw of sample and partition,*

$$\mathrm{kl}\Big(\hat{e}(\rho^*, S) \,\Big\|\, \hat{e}(\rho^*, S\cup S')\Big) \;\le\; \frac{1}{m}\left[\log\binom{m+n}{m} + \frac{2\;\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]}{\delta} + \log\frac{2\,\xi(m,n)}{\delta}\right], \tag{11}$$

where $\rho^* = T(\cdot \mid X^m, Y^m)$, $\hat{e}(\rho^*, D) = \mathbb{E}_{\theta\sim\rho^*}[\hat{e}(\theta, D)]$ for $D \in \{S, S\cup S'\}$, and $\xi(m,n)$ is the transductive PAC-Bayes concentration constant of Bégin et al. [5]. As in Theorem 5.3, L_{preq} and $\hat{L}$

are evaluated on the training partition S, and the expectation $\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$ is taken over the joint draw of $(x, y)^{m+n}$ and the random partition.

Remark 5.6. The key step is again an expectation chain, now in mutual-information form: $\mathbb{E}\big[\mathrm{KL}\big(T(\theta \mid X^m, Y^m) \,\|\, T(\theta \mid X^m)\big)\big] = I_T(\theta; Y^m \mid X^m) \le \mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$. This sharpens the deterministic chain $H_T(\theta \mid X^m) \le \mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$ by exactly the algorithm's intrinsic entropy $H_T(\theta \mid X^m, Y^m)$, i.e., the bits-back term. Two technical points: (11) is stated on $S \cup S'$ following the convention of Bégin et al. [5], and is related to test-error-only forms via $\hat{e}(\rho^*, S \cup S') = \big(m\,\hat{e}(\rho^*, S) + n\,\hat{e}(\rho^*, S')\big)/(m+n)$; and the discrete-parameter assumption inherited from Theorem 5.3 ensures that all KL divergences are well-defined as sums, with the continuous case requiring only $T(\cdot \mid X^m, Y^m) \ll T(\cdot \mid X^m)$ for the relative entropy to be finite.

6 Experiments

We empirically test whether the prequential regret $L_{\mathrm{preq}} - \hat{L}$ tracks the generalization gap on CIFAR-10 and CIFAR-100 [26], sweeping over a range of network configurations and recording the regret alongside the held-out generalization error.

Setup. We compute L_{preq} following the empirical prequential coding scheme of Blier and Ollivier [7], partitioning the training trajectory into $K = 40$ blocks of geometrically growing size, and obtain $\hat{L}$ from the final trained network. We trained 540 networks on CIFAR-10 and 99 networks on CIFAR-100 with ResNet architectures [22, 21] of varying depth, sweeping over batch size, weight decay, dropout rate, and seed. An independent data subsample is drawn per configuration to enable comparison with the IB-based baselines [25]; our quantity itself does not require subsampling. Hyperparameters, schedules, and prequential blocking details are deferred to Appendix B. Our code is available at `https://github.com/minimumdescriptionlength/mdl_gen`.

Computing the regret. Let $0 = m_0 < m_1 < \cdots < m_K = m$ be the block boundaries, so that block B_{t+1} consists of examples $m_t + 1, \ldots, m_{t+1}$. We next describe the procedure of computing the regret, in the following four steps. (1) The first block is encoded with the uniform distribution over classes, at $m_1 \log |\mathcal{Y}|$ bits. (2) For $t = 1, \ldots, K-1$, $\hat{\theta}_t$ is trained on the first m_t examples, warm-started from $\hat{\theta}_{t-1}$, and the cross-entropy of block B_{t+1} under $Q_{\hat{\theta}_t}$ is added to L_{preq}. (3) $\hat{L}$ is the cross-entropy of the final model θ^*, trained on all m examples, on the *training* set. (4) $R = L_{\mathrm{preq}} - \hat{L}$.

Table 1: Correlations between generalization-bound quantities and the empirical generalization gap on CIFAR-10, measured both in loss and in classification error. Notation follows the previous work on IB generalization bounds [25]. We write S for the training sample, D for the network depth, $\check{I}$ for a finite-sample mutual-information estimator, Z_l^s for the activations at layer l on data subsample s, θ_l^S for the parameters at layer l trained on S, and θ_{D+1}^S for the full network parameter trained on S. The baseline $\check{I}(S; \theta_{D+1}^S)$ is the parameter-information quantity $I(S; W)$ of the information-theoretic line [33, 42] and is the only pure model-complexity baseline besides ours. Best results are bolded.

	Spearman corr.		Pearson corr.		Kendall corr.	
Generalization gap:	Loss	Error	Loss	Error	Loss	Error
$\frac{1}{D}\sum_{l=1}^{D} \check{I}(X; Z_l^s)$	0.8481	0.7410	0.2116	0.1831	0.6425	0.5436
$\min_{l \in [D]} \check{I}(X; Z_l^s)$	0.7145	0.5602	0.7203	0.5719	0.4461	0.3404
$\frac{1}{D}\sum_{l=1}^{D} \check{I}(X; Z_l^s \mid Y)$	0.8481	0.7406	0.2140	0.1853	0.6427	0.5435
$\min_{l \in [D]} \check{I}(X; Z_l^s \mid Y)$	0.7004	0.5434	0.7062	0.5560	0.4386	0.3305
$\check{I}(S; \theta_{D+1}^S)$	0.4688	0.3112	0.2512	0.0775	0.2121	0.1208
$\min_{l \in [D]} \check{I}(S; \theta_l^S) + \check{I}(X; Z_l^s \mid Y)$	0.8434	0.7313	0.8437	0.7195	0.6270	0.5332
$L_{\mathrm{preq}} - \hat{L}$ **(Ours)**	**0.8784**	**0.8846**	**0.9211**	**0.9172**	**0.6800**	**0.6874**

Results on CIFAR-10. Our hyperparameter and seed grid matches that of the IB-based baselines [25] on CIFAR-10, so we take their reported numbers directly into Table 1. In their pipeline, the mutual information terms $\check{I}(X; Z_l^s)$ and $\check{I}(X; Z_l^s \mid Y)$ are estimated by adaptive kernel density estimation [11] on layer-l activations, while the parameter-information term $\check{I}(S; \theta_l^S)$ is estimated by

fitting a SWAG [29] posterior over parameters across data subsamples and applying a Jensen upper bound. The 'Loss' and 'Error' columns report the correlation between each complexity quantity and the generalization gap measured in loss and in classification error, respectively.

The prequential regret outperforms all baselines across all six rank and linear correlation columns. The most informative comparison is with $\check{I}(S;\theta^S_{D+1})$, the only other pure model-complexity quantity in the table (no representation-level term). The gap there is substantial: Pearson 0.92 versus 0.25 on the loss gap, and 0.92 versus 0.08 on the error gap. We attribute the gap to the estimation pipeline: $\check{I}(S;\theta^S_{D+1})$ requires fitting a high-dimensional Gaussian posterior over parameters and is sensitive to the SWAG approximation, whereas the prequential regret is read directly off the training trajectory and avoids high-dimensional density estimation altogether.

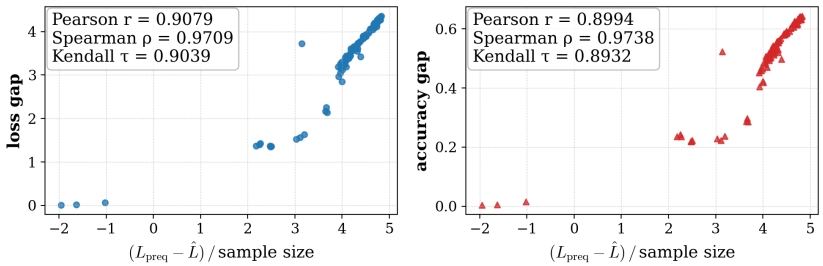


Figure 1: Correlation between the prequential regret and the generalization gaps on CIFAR-100.

Results on CIFAR-100. Figure 1 plots the empirical prequential regret against the loss and accuracy gaps, and shows similarly strong correlations. The three outliers in the bottom-left quadrant have negative empirical regret and therefore cannot upper-bound any description length. They correspond to severely underfit configurations (training accuracy below 0.25) where small networks are paired with overly aggressive weight decay, which violates the interpolation-genericity condition (6) and places these cases outside the regime where Theorems 5.3 and 5.5 apply. On the remaining configurations, which cover the regime of practical interest, the bound is well-defined and tracks the gap closely.

Due to the higher per-configuration cost on CIFAR-100, we restrict this evaluation to 99 network configurations. We do not report the IB baselines of Kawaguchi et al. [25] on CIFAR-100, as their post-training estimation did not finish within our 48-hour compute budget.

7 Conclusion

We approached the question of why over-parameterized neural networks generalize through the lens of MDL, and proposed the prequential regret $L_{\text{preq}} - \hat{L}$ as an operational measure of model complexity computable from the training trajectory alone. The expected regret upper-bounds the description length of the trained parameter under the algorithm-induced distribution, yields a generalization bound in the PAC-MDL setting, and tracks the empirical generalization gap on CIFAR-10 and CIFAR-100 with Pearson correlations exceeding 0.9. Along the way, we connected parameter compression in the MDL sense to the algorithm-level information bottleneck, showing that the encoder-complexity correction recently added to IB is upper-bounded by an MDL parameter description length.

Limitations and future work. Our generalization bound holds in the transductive PAC-MDL setting, where the test set is finite and fixed. Extending it to a population-risk statement is a concrete open problem. Further, the prequential regret itself is architecture- and modality-agnostic, and tracking how it behaves on transformers and on language or multimodal data is an interesting future direction, particularly in regimes where the nearly interpolating assumption is stressed, such as large-scale pretraining.

Acknowledgments and Disclosure of Funding

This work is partly funded by the Research Council of Norway (RCN) under grant 309439 and Dutch Research Council (NWO) under grant no. ENWSS.2018.003, NWA.1160.18.316.

References

[1] A. Achille and S. Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2897–2905, 2018.

[2] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[3] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of machine learning research*, 3(Nov):463–482, 2002.

[4] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[5] L. Bégin, P. Germain, F. Laviolette, and J.-F. Roy. Pac-bayesian theory for transductive learning. In *Artificial Intelligence and Statistics*, pages 105–113. PMLR, 2014.

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

[7] L. Blier and Y. Ollivier. The description length of deep learning models. *Advances in Neural Information Processing Systems*, 31, 2018.

[8] A. Blum and J. Langford. Pac-mdl bounds. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pages 344–357. Springer, 2003.

[9] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[10] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[11] I. Chelombiev, C. Houghton, and C. O'Donnell. Adaptive estimators show information compression in deep neural networks. *arXiv preprint arXiv:1902.09037*, 2019.

[12] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[13] A. P. Dawid. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290, 1984.

[14] E. Donoway, H. Joren, F. Roger, and J. Leike. Excess description length of learning generalizable predictors. *arXiv preprint arXiv:2601.04728*, 2026.

[15] G. K. Dziugaite, K. Hsu, W. Gharbieh, G. Arpino, and D. Roy. On the role of data in pac-bayes bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 604–612. PMLR, 2021.

[16] P. Grünwald. *The minimum description length principle*. MIT press, 2007.

[17] P. Grünwald and T. Roos. Minimum description length revisited. *arXiv preprint arXiv:1908.08484*, 2019.

[18] H. Hafez-Kolahi, S. Kasaei, and M. Soleymani-Baghshah. Sample complexity of classification with compressed input. *Neurocomputing*, 415:286–294, 2020.

[19] M. Haghifam, J. Negrea, A. Khisti, D. M. Roy, and G. K. Dziugaite. Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. *Advances in Neural Information Processing Systems*, 33:9925–9935, 2020.

[20] H. Harutyunyan, M. Raginsky, G. Ver Steeg, and A. Galstyan. Information-theoretic generalization bounds for black-box learning algorithms. *Advances in Neural Information Processing Systems*, 34:24670–24682, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[23] G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

[24] A. Honkela and H. Valpola. Variational learning and bits-back coding: an information-theoretic view to bayesian learning. *IEEE transactions on Neural Networks*, 15(4):800–810, 2004.

[25] K. Kawaguchi, Z. Deng, X. Ji, and J. Huang. How does information bottleneck help deep learning? In *International conference on machine learning*, pages 16049–16096. PMLR, 2023.

[26] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[27] M. Li, P. Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

[28] S. Lotfi, M. Finzi, S. Kapoor, A. Potapczynski, M. Goldblum, and A. G. Wilson. Pac-bayes compression bounds so tight that they can explain generalization. *Advances in Neural Information Processing Systems*, 35:31459–31473, 2022.

[29] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019.

[30] C. Mingard, H. Rees, G. Valle-Pérez, and A. A. Louis. Deep neural networks have an inbuilt occam's razor. *Nature Communications*, 16(1):220, 2025.

[31] H. M. Proença and M. van Leeuwen. Interpretable multiclass classification by mdl-based rule lists. *Information Sciences*, 512:1372–1393, 2020.

[32] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[33] D. Russo and J. Zou. Controlling bias in adaptive data analysis using information theory. In *Artificial Intelligence and Statistics*, pages 1232–1240. PMLR, 2016.

[34] M. Sefidgaran, A. Zaidi, and P. Krasnowski. Minimum description length and generalization guarantees for representation learning. *Advances in Neural Information Processing Systems*, 36: 1489–1525, 2023.

[35] M. Sefidgaran, A. Zaidi, and P. Krasnowski. Generalization guarantees for representation learning via data-dependent gaussian mixture priors. *arXiv preprint arXiv:2502.15540*, 2025.

[36] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[37] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[38] T. Steinke and L. Zakynthinou. Reasoning about generalization via conditional mutual information. In *Conference on Learning Theory*, pages 3437–3452. PMLR, 2020.

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

[40] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[41] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[42] A. Xu and M. Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in neural information processing systems*, 30, 2017.

[43] L. Yang and M. van Leeuwen. Truly unordered probabilistic rule sets for multi-class classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 87–103. Springer, 2022.

[44] L. Yang and M. van Leeuwen. Conditional density estimation with histogram trees. In *Advances in Neural Information Processing Systems*, volume 37, pages 117315–117339. Curran Associates, Inc., 2024. doi: 10.52202/079017-3724.

[45] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

A Proofs

A.1 Proof of Proposition 3.1

Proof. Apply the chain rule to $(X^m; Z^m, \phi_l)$ in two ways:

$$I(X^m; Z^m, \phi_l) \;=\; I(X^m; \phi_l) + I(X^m; Z^m \mid \phi_l) \;=\; I(X^m; Z^m) + I(X^m; \phi_l \mid Z^m).$$

Equating and rearranging gives (3); dropping $I(X^m; \phi_l \mid Z^m) \geq 0$ gives (4). □

A.2 Proof of Theorem 4.1

Proof. (i) Pointwise. Taking $\log$ on both sides of (6) and subtracting $\log Q_{\mathrm{preq}}(Y^m \mid X^m)$,

$$\log \frac{Q_\theta(Y^m \mid X^m)}{Q_{\mathrm{preq}}(Y^m \mid X^m)} \;\geq\; \log \frac{T(Y^m \mid X^m, \theta)}{Q_{\mathrm{preq}}(Y^m \mid X^m)}. \tag{12}$$

The LHS equals $L_{\mathrm{preq}} - \hat{L}$. For the RHS, multiply and divide by $P(Y^m \mid X^m)$:

$$\log \frac{T(Y^m \mid X^m, \theta)}{Q_{\mathrm{preq}}(Y^m \mid X^m)} \;=\; \log \frac{T(Y^m \mid X^m, \theta)}{P(Y^m \mid X^m)} \;+\; \log \frac{P(Y^m \mid X^m)}{Q_{\mathrm{preq}}(Y^m \mid X^m)}.$$

Since the algorithm-induced joint $T(\theta, Y^m \mid X^m)$ factors two ways,

$$T(\theta \mid X^m)\, T(Y^m \mid X^m, \theta) \;=\; P(Y^m \mid X^m)\, T(\theta \mid X^m, Y^m),$$

so

$$\log \frac{T(Y^m \mid X^m, \theta)}{P(Y^m \mid X^m)} \;=\; \log T(\theta \mid X^m, Y^m) - \log T(\theta \mid X^m).$$

Substituting back into (12) yields (7).

(ii) In expectation. Taking expectations of (7) over the joint distribution and using $\mathbb{E}[-\log T(\theta \mid X^m)] = H_T(\theta \mid X^m)$, $\mathbb{E}[-\log T(\theta \mid X^m, Y^m)] = H_T(\theta \mid X^m, Y^m)$, and $\mathbb{E}[\log P(Y^m \mid X^m)/Q_{\mathrm{preq}}(Y^m \mid X^m)] = \mathbb{E}_{X^m}[\mathrm{KL}(P(\cdot \mid X^m) \| Q_{\mathrm{preq}}(\cdot \mid X^m))] \geq 0$, we obtain (8). □

A.3 Proof of Theorem 5.3

The proof has two steps: a PAC-MDL bound at confidence $1 - \delta/2$ with a partition-invariant prior, and a Markov bound at confidence $1 - \delta/2$ converting the expected-description-length result of Theorem 4.1 into a high-probability lower bound on $T(\theta \mid X^m)$. A union bound combines them.

Step 1: A partition-invariant prior and PAC-MDL. A description method $\sigma = (\pi, \theta)$ specifies a training partition $\pi \subset [m+n]$ with $|\pi| = m$ together with a parameter θ; given x^{m+n}, the labeling is determined by $y_\sigma(x_i) = \theta(x_i)$. Define

$$P(\pi, \theta) \;=\; \frac{T(\theta \mid X_\pi^m)}{\binom{m+n}{m}},$$

where X_π^m are the inputs at indices in π. The prior is determined by $\mathcal{A}$ and x^{m+n}, both fixed in advance of the random partition, and sums to one since $\sum_\theta T(\theta \mid X_\pi^m) = 1$ for each π and there are $\binom{m+n}{m}$ partitions. Applying Theorem 5.1 with confidence $1-\delta/2$ gives that, with probability at least $1-\delta/2$ over the joint draw, the realized $\sigma^* = (\pi^*, \theta^*)$ with $\theta^* = \mathcal{A}(S)$ satisfies

$$n\hat{e}(\theta^*, S') \;\le\; b_{\max}\!\left(n,\, \hat{e}(\theta^*, S),\, \frac{T(\theta^* \mid X_{\pi^*}^m)}{\binom{m+n}{m}} \cdot \frac{\delta}{2}\right), \tag{13}$$

where we used that the labeling produced by σ^* on S and S' coincides with that of θ^*.

Step 2: Markov bound on the description length. The random variable $-\log T(\theta \mid X^m)$ is non-negative, and Theorem 4.1(ii) under the deterministic assumption ($H_T(\theta \mid X^m, Y^m) = 0$) gives $\mathbb{E}[-\log T(\theta \mid X^m)] = H_T(\theta \mid X^m) \le \mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$. By Markov's inequality, with probability at least $1-\delta/2$,

$$-\log T(\theta \mid X^m) \;\le\; \frac{2\,\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]}{\delta}, \quad \text{i.e.,} \quad T(\theta \mid X^m) \;\ge\; 2^{-2\,\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]/\delta}. \tag{14}$$

Step 3: Union bound. The events in (13) and (14) each hold with probability at least $1-\delta/2$ over the joint draw of sample and partition. By the union bound, both hold simultaneously with probability at least $1-\delta$. On this event, since $b_{\max}(n, a, \cdot)$ is non-increasing in its third argument [8], plugging the lower bound on $T(\theta^* \mid X_{\pi^*}^m)$ from (14) into (13) gives

$$n\hat{e}(\theta^*, S') \;\le\; b_{\max}\!\left(n,\, \hat{e}(\theta^*, S),\, \frac{\delta}{2\binom{m+n}{m}} \cdot 2^{-2\,\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]/\delta}\right). \quad \square$$

A.4 Proof of Theorem 5.5

The proof parallels that of Theorem 5.3, with transductive PAC-Bayes replacing PAC-MDL and Markov applied to a KL divergence rather than to a description length.

Step 1: Lifted prior and posterior; transductive PAC-Bayes. Lift the hypothesis to a pair (π, θ) as in the proof of Theorem 5.3, and define

$$\pi_0(\pi, \theta) \;=\; \frac{T(\theta \mid X_\pi^m)}{\binom{m+n}{m}}, \qquad \rho(\pi, \theta) \;=\; \mathbf{1}[\pi = \pi^*]\, T(\theta \mid X_{\pi^*}^m, Y_{\pi^*}^m),$$

where π^* is the realized partition. The prior π_0 is determined by $\mathcal{A}$ and x^{m+n}, both fixed in advance of the partition, and is a probability distribution by the same calculation as in Theorem 5.3. Applying the transductive PAC-Bayes inequality (5, Theorem 1) on the lifted hypothesis space at confidence $1-\delta/2$ gives, with probability at least $1-\delta/2$ over the joint draw of sample and partition,

$$\mathrm{kl}\big(\hat{e}(\rho, S) \,\|\, \hat{e}(\rho, S \cup S')\big) \;\le\; \frac{\mathrm{KL}(\rho \,\|\, \pi_0) + \log(2\xi(m,n)/\delta)}{m}. \tag{15}$$

Since ρ concentrates on $\pi = \pi^*$, the lifted Gibbs errors reduce to $\hat{e}(\rho, D) = \hat{e}(\rho^*, D)$ for $D \in \{S, S \cup S'\}$, and the KL decomposes as

$$\begin{aligned}\mathrm{KL}(\rho \,\|\, \pi_0) \;&=\; \sum_\theta T(\theta \mid X_{\pi^*}^m, Y_{\pi^*}^m) \log \frac{T(\theta \mid X_{\pi^*}^m, Y_{\pi^*}^m)}{T(\theta \mid X_{\pi^*}^m)/\binom{m+n}{m}} \\ &=\; \log\binom{m+n}{m} + \mathrm{KL}\big(T(\theta \mid X_{\pi^*}^m, Y_{\pi^*}^m) \,\|\, T(\theta \mid X_{\pi^*}^m)\big).\end{aligned}$$

Step 2: Markov on the KL. The random variable $\mathrm{KL}\big(T(\theta \mid X^m, Y^m) \,\|\, T(\theta \mid X^m)\big)$ is non-negative, and by the identity $\mathbb{E}_{(B,C)}[\mathrm{KL}(P_{A|B,C} \,\|\, P_{A|C})] = I(A; B \mid C)$ applied with $A = \theta$, $B = Y^m$, $C = X^m$,

$$\mathbb{E}\big[\mathrm{KL}\big(T(\theta \mid X^m, Y^m) \,\|\, T(\theta \mid X^m)\big)\big] \;=\; I_T(\theta; Y^m \mid X^m).$$

The bits-back extension of Theorem 4.1(ii) gives $I_T(\theta; Y^m \mid X^m) \le \mathbb{E}[L_{\mathrm{preq}} - \hat{L}]$, so by Markov's inequality, with probability at least $1-\delta/2$,

$$\mathrm{KL}\big(T(\theta \mid X^m, Y^m) \,\|\, T(\theta \mid X^m)\big) \;\le\; \frac{2\,\mathbb{E}[L_{\mathrm{preq}} - \hat{L}]}{\delta}. \tag{16}$$

Step 3: Union bound. The events in (15) and (16) each hold with probability at least $1 - \delta/2$, so both hold simultaneously with probability at least $1 - \delta$. On this event, substituting (16) into (15) via the KL decomposition yields (11). □

B Additional Experiment Details

B.1 CIFAR-10 Experiment Details

We replicate the 30%-subsample CIFAR-10 protocol of Kawaguchi et al. [25]. For each (seed, instance) pair we draw a 15,000-image subset of the CIFAR-10 training split using the original RNG order from their released codebase (a sequential `np.random.choice` over five instances after seeding), so our subsamples match theirs bit-for-bit. Inputs are normalized with the standard CIFAR-10 statistics; training uses random crops (`padding`$= 4$) and horizontal flips, while bit-coded evaluation passes use no augmentation. We use the three architectures from the original work, PreResNet-$\{56, 83, 110\}$, imported directly as a submodule of their released code.

Each block is trained with SGD (momentum $= 0.9$, initial learning rate $= 0.1$, mean-reduction cross-entropy, no mixed precision, no label smoothing). The learning rate follows the piecewise schedule of Kawaguchi et al. [25] with $\mathrm{swa_start} = 0.805 \times E$ epochs: constant at the initial rate for the first half of pre-SWA training, linearly decayed to $\mathrm{swa_lr} = 0.05$ over the next 40%, and constant at swa_lr thereafter. We sweep weight decay $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$ and batch size $\in \{64, 128, 1024\}$, identical to their Table 2 grid.

Prequential MDL layer. On top of this recipe we run the prequential coding procedure: each 15k-sample instance is partitioned into $K \leq 40$ blocks whose cumulative prefix sizes grow geometrically from $\max(\lceil 0.005\, m \rceil, 100)$ to m. The first block B_1 is coded with the uniform class prior ($\log_2 10$ bits/sample), and for each subsequent block we train on the prefix and evaluate cross-entropy on the held-out block. Per-block epoch budgets are tiered by prefix fraction r: $40/80/120/200$ epochs for $r < 0.05/ < 0.20/ < 0.50/ \geq 0.50$, so that the final block matches the original 200-epoch straight-training budget. We use warm-start (the model is carried across blocks) and report $R/m = (L_{\mathrm{preq}} - \hat{L})/m$ as the predictor; targets are $\mathrm{GL} = \mathrm{test_loss} - \mathrm{train_loss}$ (in bits) and $\mathrm{GE} = \mathrm{train_acc} - \mathrm{test_acc}$.

Setup grid. The full grid is $3\,(\mathrm{arch}) \times 3\,(\mathrm{wd}) \times 3\,(\mathrm{bs}) \times 4\,(\mathrm{seed}) \times 5\,(\mathrm{instance}) = 540$ cells, matching the CIFAR-10 grid of Kawaguchi et al. [25]. Following their filter, correlations are reported on cells with training accuracy above 0.85.

B.2 CIFAR-100 Experiment Details

We use the standard CIFAR-100 training split ($m = 50{,}000$, no subsampling) and the held-out test split ($n = 10{,}000$). Inputs are normalized with the standard CIFAR-100 statistics ($\mathrm{mean} = (0.5071, 0.4865, 0.4409)$, $\mathrm{std} = (0.2673, 0.2564, 0.2762)$); training uses random crops (`padding`$=$ 4) and horizontal flips, while bit-coded evaluation passes use no augmentation. Architectures are the torchvision ResNet family at five depths, ResNet-$\{18, 34, 50, 101, 152\}$, sweeping model size as a clean log-of-parameter-count axis without changing the residual block design.

Each block is trained with SGD (Nesterov momentum $= 0.9$, initial learning rate $= 0.1$, label smoothing $= 0.1$, batch size $= 256$, mean-reduction cross-entropy, no mixed precision). The learning-rate schedule is per-batch warmup-then-cosine: a linear warmup over the first 5% of the block's optimizer steps starting at $0.1\times$ the initial rate, followed by cosine annealing to $\eta_{\min} = 10^{-4}$ over the remaining steps; the schedule is rebuilt at each block over its own step budget. We run three independent repeats per cell, with warm-start across blocks (model and optimizer state are carried over from the previous prefix; no cold restarts).

Prequential MDL layer. We run the same prequential coding procedure as on CIFAR-10: the full 50k-sample training set is partitioned into $K \leq 40$ blocks whose cumulative prefix sizes grow geometrically from $\max(\lceil 0.005\, m \rceil, 100) = 250$ to $m = 50{,}000$. The first block B_1 is coded with the uniform class prior ($\log_2 100 \approx 6.644$ bits/sample), and for each subsequent block we train on the prefix and evaluate cross-entropy on the held-out block. Per-block epoch budgets follow the same prefix-fraction tiers as on CIFAR-10 ($40/80/120/200$ epochs for $r < 0.05/ < 0.20/ < 0.50/ \geq 0.50$), so the final block again receives a 200-epoch budget. We report $R/m = (L_{\mathrm{preq}} - \hat{L})/m$ as

the predictor; targets are $\mathrm{GL} = \mathrm{test_loss} - \mathrm{train_loss}$ in bits/sample (computed post hoc as test bits-per-image minus final-train bits-per-image) and $\mathrm{GE} = \mathrm{train_acc} - \mathrm{test_acc}$.

Setup grid. The CIFAR-100 grid is built from three orthogonal axes that share a common anchor (ResNet-34, $\mathrm{wd} = 0$, $\mathrm{dropout} = 0$, run for three repeats):

- **Architecture sweep.** $\mathrm{arch} \in \{\text{ResNet-}18, 50, 101, 152\}$ at $\mathrm{wd} = 0$, $\mathrm{dropout} = 0$; $4 \times 3\,(\text{repeat}) = 12$ cells. Together with the anchor's three ResNet-34 repeats, this covers all five depths.
- **Weight-decay sweep.** ResNet-34 at $\mathrm{dropout} = 0$, $\mathrm{wd} \in \{10^{-6}, 3{\cdot}10^{-6}, 10^{-5}, 3{\cdot}10^{-5}, 10^{-4}, 3{\cdot}10^{-4}, 10^{-3}, 3{\cdot}10^{-3}, 10^{-2}\}$; $9 \times 3 = 27$ cells. Combined with the anchor at $\mathrm{wd} = 0$, this gives 10 weight-decay values in total.
- **Dropout sweep.** ResNet-34 at $\mathrm{wd} = 0$, $\mathrm{dropout}$ on a 19-point uniform grid in $(0, 0.5]$; $19 \times 3 = 57$ cells. Combined with the anchor at $\mathrm{dropout} = 0$, this gives a 20-point uniform grid on $[0, 0.5]$.

The three sweeps together with the shared anchor yield $12 + 27 + 57 + 3 = 99$ cells.

B.3 Computation Resources

All experiments were executed on NVIDIA A100 GPUs via SLURM array jobs on a shared HPC cluster. Each individual run was allocated:

- **GPU**: 1 × NVIDIA A100 (80 GB GPU memory)
- **CPU**: 4 CPU cores
- **Host memory**: 16 GB
- **Wall-clock time limits**: 24 hours per task for weight-decay and dropout sweeps; 48 hours per task for the architecture sweep (ResNet-152 training takes longer).

The CIFAR-10 grid (540 cells, including all subsampling instances) was executed similarly: on the same hardware, with 24–48 hour allocations depending on the run complexity. For CIFAR-100, the three sweeps (architecture, weight decay, dropout) were orchestrated as separate array jobs: the architecture sweep ran up to 5 tasks in parallel (`-array=0-4`), and the weight-decay and dropout sweeps each ran up to 20 tasks in parallel (`-array=0-19`). Most individual runs completed well within their wall-clock allocations; the 48-hour limit for the architecture sweep served as a safety margin for the largest models (ResNet-152) under variable cluster load. Typical single-run wall time was 8–16 hours for CIFAR-100 and 6–12 hours for CIFAR-10 (30%-subsample), depending on architecture size and weight-decay value.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) state three contributions that map directly to the paper's body: connecting MDL parameter compression to the algorithm-level information bottleneck (Section 3), proving that the prequential regret upper-bounds the description length of the trained parameter under the algorithm-induced distribution in the nearly-interpolating regime (Section 4), turning this into a PAC-MDL generalization bound (Section 5), and empirically tracking the generalization gap on CIFAR-10/100 with Pearson correlations exceeding 0.9 (Section 6). Limitations on the regime of validity and on the transductive setting are flagged in Section 7.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.

- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 ("Limitations and future work") notes that our generalization bound lives in the transductive PAC-MDL setting and that lifting it to a population-risk statement is left open, and that the empirical validation is restricted to ResNets on CIFAR-10/100 in the nearly-interpolating regime. The interpolation-genericity assumption (Equation 7) and the discrete (finite-precision) parameter space requirement are stated explicitly in Theorems 4.1, 5.3, and 5.5, and Remark 4.2 discusses when these conditions hold or fail. Section 6 also reports outliers outside the nearly-interpolating regime where the bound is not expected to apply.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results (Proposition 3.1, Theorem 4.1, Theorem 5.1, Theorem 5.3, Theorem 5.5) are numbered, cross-referenced, and stated with their assumptions in the main text—e.g., the interpolation-genericity condition in Equation (7), the discrete/finite-precision parameter space, and the transductive PAC-MDL setup. Complete proofs are given in Appendix A (Sections A.1–A.4), with the corresponding intuition and proof sketches provided for the main theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 6 describes the empirical prequential coding scheme, the partition into $T = 40$ blocks of geometrically growing size, the architectures (ResNet family), and the datasets (CIFAR-10/100). Appendix B gives the full reproduction recipe: subsampling protocol, normalization and augmentation, optimizer (SGD with momentum, learning-rate schedule, weight decay grid, batch sizes, label smoothing), the prequential block schedule and per-block epoch budgets, the warm-start protocol, the full $3 \times 3 \times 3 \times 4 \times 5 = 540$-cell CIFAR-10 grid, and the $15 + 30 + 60 - 3 = 99$-cell CIFAR-100 grid (architecture, weight-decay, dropout sweeps). An anonymized code release is also provided (see next item).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized code repository is linked in Section 6 (`https://anonymous.4open.science/r/mdl_gen-E317`), containing the prequential coding implementation and training scripts. The datasets used (CIFAR-10 and CIFAR-100) are publicly available standard benchmarks. Hyperparameters, schedules, and prequential blocking details are documented in Appendix B in sufficient detail to reproduce the experimental grids.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 6 specifies the high-level setup (datasets, architecture families, prequential blocking, sweep over depth, batch size, weight decay, dropout, seed). Appendix B.1 (CIFAR-10) and B.2 (CIFAR-100) give the data splits, normalization, augmentation, optimizer settings (SGD momentum, learning-rate schedules), weight-decay grids, batch sizes, dropout grids, prequential block schedule, per-block epoch budgets, and the rationale for each grid choice (matching the IB baseline of Kawaguchi et al. [25] on CIFAR-10).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Our central empirical claim is that the prequential regret tracks the generalization gap across many configurations, and we substantiate it with three rank/linear correlation statistics (Pearson, Spearman, Kendall) computed across the full 540-cell (CIFAR-10) and 99-cell (CIFAR-100) hyperparameter grids, together with scatter plots that display all per-configuration points (Table 1 and Figure 1). We do not report confidence intervals on the correlation values themselves; the main source of variability—across architecture, weight decay, dropout, batch size, and seed—is reflected in the spread of points entering the correlation, not in error bars on a single number.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 6 reports that 540 networks were trained on CIFAR-10 and 99 on CIFAR-100, and that the CIFAR-100 evaluation was constrained by a 48-hour compute budget—which itself motivated the smaller grid. Appendix B specifies the full per-cell training cost (200-epoch budget per cell, prequential block schedule, batch sizes, optimizer settings), and also reports the compute resources (GPU, CPU, host memory, and wall-clock time limits).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The work is theoretical and uses only standard public image-classification benchmarks (CIFAR-10/100); it involves no human subjects, no personally identifiable data, and no model release with foreseeable misuse risk. We have reviewed the NeurIPS Code of Ethics and the research conforms to it; anonymity is preserved in the submission (anonymized code URL).

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: This is foundational theoretical work on understanding generalization in over-parameterized neural networks via the MDL principle. It does not propose a deployable system, a generative model, or any application tied to a specific user-facing technology, and we are not aware of a direct path from the prequential regret as a generalization diagnostic to a specific positive or negative societal impact beyond the generic effects of progress in learning theory.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper releases only research code for computing the prequential regret on standard public image-classification benchmarks; no pretrained generative models, language models, or scraped datasets with misuse risk are released.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets used are cited: CIFAR-10/100 (Krizhevsky [26]), the ResNet/PreResNet architectures (He et al. [21, 22]), the IB baseline pipeline of Kawaguchi et al. [25] whose released codebase we reuse for the CIFAR-10 subsampling protocol (Appendix B.1), the SWAG posterior approximation (Maddox et al. [29]), the prequential coding scheme of Blier and Ollivier [7], and the PAC-MDL and transductive PAC-Bayes frameworks (Blum and Langford [8]; Bégin et al. [5]). These are standard research assets used under their public terms and within their intended research use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release research code implementing the prequential coding procedure and the experimental pipeline at the anonymized URL given in Section 6 (`https://anonymous.4open.science/r/mdl_gen-E317`). Documentation of the algorithm and the configurations it reproduces is provided in Section 6 and Appendix B; no new datasets or pretrained models are released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects; all experiments are on standard public image-classification benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the core methodology of this paper; the contributions are an MDL-based generalization analysis and image-classification experiments on CIFAR-10/100. Any use of LLMs was limited to writing/editing assistance and does not affect scientific content.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.