A Compression Perspective on Simplicity Bias

Tom Marty[*1,2] **Eric Elmoznino**[1,2] **Leo Gagnon**[1,2] **Tejas Kasetty**[1,2] **Mizu Nishikawa-Toomey**[1,2]
Sarthak Mittal[1,2] **Guillaume Lajoie**[1,2] **Dhanya Sridhar**[†1,2]

[1]Mila – Quebec AI Institute
[2]Université de Montréal

Abstract

Deep neural networks exhibit a simplicity bias, a well-documented tendency to favor simple functions over complex ones. In this work, we cast new light on this phenomenon through the lens of the Minimum Description Length principle, formalizing supervised learning as a problem of optimal two-part lossless compression. Our theory explains how simplicity bias governs feature selection in neural networks through a fundamental trade-off between model complexity (the cost of describing the hypothesis) and predictive power (the cost of describing the data). Our framework predicts that as the amount of available training data increases, learners transition through qualitatively different features – from simple spurious shortcuts to complex features – only when the reduction in data encoding cost justifies the increased model complexity. Consequently, we identify distinct data regimes where increasing data promotes robustness by ruling out trivial shortcuts, and conversely, regimes where limiting data can act as a form of complexity-based regularization, preventing the learning of unreliable complex environmental cues. We validate our theory on a semi-synthetic benchmark showing that the feature selection of neural networks follows the same trajectory of solutions as optimal two-part compressors.

1 INTRODUCTION

Several works point to the simplicity bias exhibited in the training of deep neural networks [Arpit et al., 2017, Valle-Pérez et al., 2018, Shah et al., 2020, Mingard et al., 2025]. Put simply, simplicity bias refers to the tendency of learning algorithms such as stochastic gradient descent (SGD) to find solutions that encode "simple functions." In this paper, we take the view that the notion of simplicity bias is captured by the Minimum Description Length (MDL) principle [Grünwald, 2004, 2007]. The MDL principle formalizes simplicity from the lens of lossless compression: we can describe a dataset with as few bits as possible if we have a model that accurately predicts the data (i.e., minimizes the negative log-likelihood) while requiring minimal bits to encode the model itself (i.e., by leveraging the structure in the data). In this paper, we revisit simplicity bias in neural networks by studying whether optimal compression serves as a predictive theory of neural network behavior. We then characterize the resulting implications of this theory for out-of-distribution (OOD) generalization.

A large body of work on simplicity bias in neural networks has focused on supervised learning in the presence of *spurious features*: latent properties of inputs that are easy to extract and use for prediction, but are unreliable. For example, consider the problem of classifying images of birds as "water-based" or "land-based" as introduced in Sagawa et al. [2020]. The background of the image (the presence or lack of water) often co-occurs with its corresponding label, offering an appealing shortcut that a machine learning classifier can easily exploit to achieve good performance in-distribution, but which leads to poor predictions if the spurious co-occurrence disappears in a shifted test distribution. In contrast, a human labeller might rely exclusively on a bird's phenotypical characteristics, classifying images accurately even when a water bird is on land. Studies point to the fact that when learning predictive models from a static dataset, a preference for simple functions can be a curse, leading to models with poor generalization under distribution shift due to their reliance on spurious features [Geirhos et al., 2020, Teney et al., 2022, Vasudeva et al., 2023].

This paper contributes to our understanding of the interactions between learning under simplicity bias and OOD generalization. Concretely, we focus our MDL predictive theory on heterogeneous data sampled from a mixture of underlying distributions that we refer to as environments. We vary relevant properties of heterogeneous data – the amount

of training data available, the complexity and predictiveness of different features, etc. – and use our theory to predict which features a learned model will use in each data regime.

We summarize the following contributions:

- We formalize supervised learning under simplicity bias as two-part lossless compression, following the MDL principle. We operationalize our predictive theory using prequential coding: a tractable method for approximating the complexity of a function.
- We show how simplicity bias gives rise to a dynamic feature preference that **depends on the amount of training data available**. Under this compression paradigm, learners behave as MDL-optimal compressors, shifting between distinct solutions when the reduction in data encoding cost outweighs the increase in model complexity.
- We provide empirical evidence on heterogeneous semi-synthetic datasets demonstrating that our MDL-based framework accurately predicts the OOD behavior of learners, validating it as a quantitative theory for predicting generalization failure modes in limited data regimes.

Code to reproduce our experiments is available at https://github.com/3rdCore/complicity.

2 THEORY

In this section, we formalize the connection between simplicity bias and data compression. We first cast supervised learning as a two-part lossless compression problem in Section 2.1, where the learner needs to jointly minimize training error and model complexity. We then analyze in Section 2.2 how the MDL-optimal solution shifts across data regimes, showing that the dominant hypothesis transitions from simple, low-cost solutions to more predictive but complex ones as the amount of training data grows. Finally, in Section 2.3, we focus this analysis on the robust learning setting, where we identify two antagonistic scenarios that delimit a *robustness window* as a function of dataset size.

2.1 LEARNING AS OPTIMAL TWO-PART COMPRESSION

Consider a dataset of i.i.d. samples $\mathcal{D}_N = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ sampled from an unknown distribution $p^*(x, y)$. We further assume that the distribution $p^*(x, y)$ factorizes as a mixture $\sum_e p^*(e)\, p^*(x, y \mid e)$ over discrete environments $e \sim p^*(e)$. Each environment-specific distribution $p(x, y \mid e)$ reflects potentially different associations between spurious features in x and the label y. Since this paper aims to study the interactions between compression and OOD generalization, we focus on distribution shifts arising from changing $p^*(e)$ at test time.

We adopt the MDL perspective, where learning is equivalent to finding the shortest description of the training dataset $\mathcal{D}_N$. In our supervised setting, the corresponding compression task consists of encoding the labels y given the input x. In the two-part version of the MDL principle, we compress the labels sequentially by entropy-coding each y using a candidate conditional probabilistic model $p(y \mid x)$ selected from a hypothesis class $\mathcal{M}$. Because we don't know a priori which model in $\mathcal{M}$ was used to encode the data, the model itself must be encoded as well. The two-part MDL procedure consists then of two steps:

1. **Model Encoding:** We first encode the model p itself. The shortest description length of p is given by its Kolmogorov complexity, denoted $K(p)$. Kolmogorov complexity implies optimal compression, but if some other coding strategy c is used to encode the model, then we can denote the description length of the model under c as $L_c(p)$. We will see in Section 3.2 how we can compute $L_c(p)$, using a method called prequential coding.
2. **Data Encoding:** Using the model p, we then encode each label at a cost of $-\log_2 p(y \mid x)$ bits, using methods categorized as entropy coding.

Consequently, the total cost to encode the dataset $\mathcal{D}_N$ using a candidate model p is:

$$\mathcal{J}(p, \mathcal{D}_N) = \underbrace{L_c(p)}_{\text{model cost}} + \underbrace{\sum_{(x,y)\in\mathcal{D}_N} -\log p(y \mid x)}_{\text{data cost given model}} \quad (1)$$

To characterize the learning dynamics in a way that is independent of a specific dataset realization, we model neural networks as MDL-optimal compressors in expectation. In this idealized view, we denote by $\mathcal{L}$ the learner that selects the model $p \in \mathcal{M}$ which minimizes the *expected* two-part description length over datasets of size N drawn from the true distribution p^*.

By taking the expectation over the data-generating process $\mathcal{D}_N \overset{\text{iid}}{\sim} p^*$, the objective function decomposes into interpretable information-theoretic quantities:

$$\mathbb{E}_{\mathcal{D}_N\sim p^*}\left[L_{\mathcal{L}}(p) + \sum_{(x,y)\in\mathcal{D}_N} -\log p(y \mid x)\right] \quad (2)$$

$$= L_{\mathcal{L}}(p) + N \cdot \mathbb{E}_{(x,y)\sim p^*}[-\log p(y \mid x)] \quad (3)$$

$$= L_{\mathcal{L}}(p) + N \cdot \mathbb{E}_{x\sim p^*(x)}\, H(p^*_x, p_x) \quad (4)$$

$$= L_{\mathcal{L}}(p) + N \cdot \mathbb{E}_{x\sim p^*(x)}\big[H(p^*_x) + D_{KL}\big(p^*_x \| p_x\big)\big] \quad (5)$$

where $p_x = p(\cdot \mid x)$ and $p^*_x = p^*(\cdot \mid x)$. $H(p^*_x)$ is the entropy of the labels under the true conditional distribution – the irreducible epistemic uncertainty in the labels –

and $D_{KL}(p_x^* \| p_x)$ is the average amount of excess bits required to encode a label $y \sim p_x^*$ using p_x instead of the true conditional p_x^*, measuring the misalignment between the hypothesis and the true conditional distribution.

Since $H(p^*)$ is constant with respect to p, the learner effectively solves:

$$\hat{p}_N = \arg\min_{p \in \mathcal{M}} \Big[\underbrace{L_{\mathcal{L}}(p)}_{\text{model cost}} + N \cdot \underbrace{\mathbb{E}_{x \sim p^*(x)} D_{KL}(p_x^* \| p_x)}_{\text{excess data cost}} \Big] \tag{6}$$

This formulation explicitly quantifies the trade-off between model complexity[1] and average goodness-of-fit. Because this objective is a linear function of N, the optimal hypothesis $\hat{p}_N$ necessarily shifts qualitatively as the volume of training data grows. We provide a visual illustration of this phenomenon in Figure 1. This compression-centric view provides a **principled account of how data scarcity** (through N) **and the characteristics of different candidate predictors** $p \in \mathcal{M}$ (through $L_{\mathcal{L}}(p)$ and $D_{KL}(p_x^* \| p_x)$) **govern the preference of an MDL-optimal learner**.

2.2 ANALYSIS OF LEARNING REGIMES

Recalling that the idealized learner $\mathcal{L}$ chooses the model with the shortest total codelength, the decomposition above shows how the learner's preference for a solution p shifts as the amount of training data N increases (see Figure 1): the total expected description length is the sum of a **fixed bit-cost** $L_{\mathcal{L}}(p)$ and a **variable bit-cost** that grows linearly with the amount of data N. Depending on the data regime, different components of the total description length dominate:

- **Low-Data Regime:** The fixed cost $L_{\mathcal{L}}(p)$ dominates. The learner prioritizes models with short description length (i.e., *simpler* models) even if they only incompletely capture the structure in the data. This can result in overfitting – memorization of the few training datapoints – or reliance on simple spurious features.
- **High-Data Regime:** The variable linear cost of encoding labels $N \cdot \mathbb{E}_{x \sim p^*(x)} D_{KL}(p_x^* \| p_x)$ dominates. The idealized learner $\mathcal{L}$ is driven to minimize the conditional Kullback-Leibler (KL) divergence for all x in the support of $p^*(x)$, selecting the most predictive model p_x^* (or its closest approximation in $\mathcal{M}$), regardless of its complexity. Despite learning the true generative process in the limit of infinite data, this can also result in reliance on complex environment-specific features that may not be robust under distribution shifts (e.g., observations under a novel environment at test time).

[1] $L_{\mathcal{L}}$ depends on $\mathcal{L}$, since the learning algorithm can contain arbitrary prior knowledge about the task in its primitives that impact the description length of p.

The analysis above shows how the MDL-optimal compressor's preference is not fixed: it depends on the dataset size N. As N grows, preference shifts from simple, rudimentary hypotheses toward more predictive but complex ones. Assuming modern learning algorithms behave as MDL-optimal compressors [Goldblum et al., 2023, Wilson, 2025], this observation has direct implications for practitioners because in many real-world tasks inputs carry multiple predictive features – some causal and robust to distribution shift, others spurious and environment-dependent – and the learner implicitly chooses among them. Selecting features that generalize beyond the training distribution is notoriously difficult, precisely because spurious features can be both simpler and sufficiently predictive of the training data. Our framework suggests that this selection is governed by a compression trade-off: the feature the learner relies on at any given N is the one whose corresponding model minimizes the total description length. This perspective offers a principled account of *what drives feature selection across data regimes*, and can be used to predict when a learner will favor robust features over spurious shortcuts, or vice versa. In Sections 3 and 4, we empirically test these predictions on a controlled benchmark.

2.3 IMPLICATIONS FOR ROBUST LEARNING

Now that we have introduced the objective function of our idealized learner $\mathcal{L}$ and discussed how this formulation accounts for feature selection across data regimes, we next discuss its implications for robust learning.

Recall that inputs x are generated from a mixture of environments $e \in \mathcal{E}$, each inducing different correlations between features and labels. In this setting, x simultaneously carries (i) causal features (e.g., feather, spout, claw) that are invariant across environments, (ii) simple spurious features (e.g., ocean texture) whose correlation with labels varies by environment, and (iii) environment-specific signatures from which e itself can potentially be inferred. Each type of feature gives rise to a qualitatively different predictive model, and the compression framework in Section 2.1 prescribes which one the idealized learner selects at a given N. We now discuss the compression trade-off for three archetypal models that span this feature spectrum:

- **Robust Model (p_{robust}):** Uses only invariant, causal features (e.g., animal attributes). Moderate complexity, but generalizes across environments.
- **Bayes-Optimal Model (p_{bayes}):** Integrates all available information and predicts via the posterior predictive distribution:

  $$p_{\text{bayes}}(y \mid x) = \sum_{e \in \mathcal{E}} p(y \mid x, e)\, p(e \mid x)$$

  This achieves minimal empirical error, but can fail under distribution shift.

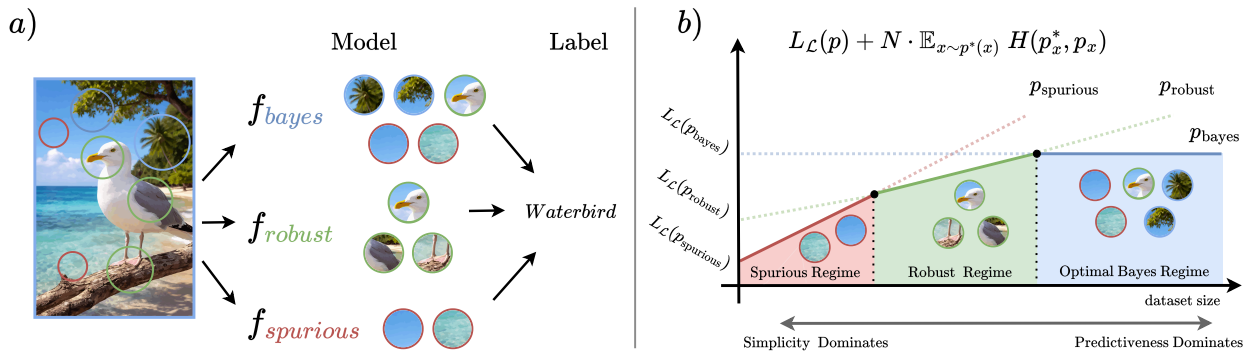


Figure 1: **(a)** In a supervised setting, some features causally relate to the label (e.g., phenotypical characteristics), while others are spurious (e.g., background of the image). A Bayesian solution leverages all available features. **(b)** Total expected compression cost as a function of training dataset size N. Each line represents the two-part description length of a candidate model. The MDL-optimal learner selects the model achieving the lowest total cost at each N, inducing transitions between qualitatively different solutions as the amount of data increases.

- **Spurious Model (p_{spur}):** Uses simple, non-robust features (e.g., background textures) and marginal environment probability $p(e)$. Low complexity, but fails under distribution shift. The spurious model is a degenerate Bayesian model: when the environment cannot be inferred from x, the predictor defaults to using the marginal $p_{\text{spur}}(y \mid x) = \sum_{e \in \mathcal{E}} p(y \mid x, e)\, p(e)$.

We consider two opposing scenarios relevant to practitioners. In each, the causal solution appears either as a simple solution with limited predictiveness, or as the most predictive but also more complex solution. For each scenario, we estimate the total description length of competing solutions and test whether, as we sweep across data regimes, the predicted solution of our idealized learner – i.e. the one that yields the lowest total description length – matches the neural network's learned solution, as reflected by shifts in feature reliance and generalization performance (see Figure 1).

Scenario A: Spurious vs. Robust solution In this scenario, the robust model p_{robust} is the most predictive candidate for modeling the data, but it is also more complex than available spurious solutions ($L_{\mathcal{L}}(p_{\text{robust}}) > L_{\mathcal{L}}(p_{\text{spur}})$ but $D_{KL}(p^* \| p_{\text{robust}}) < D_{KL}(p^* \| p_{\text{spur}})$). This is the classic "shortcut learning" case where a simple feature (e.g., a background texture), usually benefiting from strong environment imbalance, arises as a "good-enough" predictor within the amount of training data.

- **Behavior:** At low N, the idealized learner favors p_{spur} because its low description cost $L_{\mathcal{L}}(p_{\text{spur}})$ outweighs its higher divergence $D_{KL}(p^* \| p_{\text{spur}})$. The transition to the robust model only occurs once N is large enough to *pay* for the complexity overhead of p_{robust}. Said differently, increasing N makes the selection of the robust model increasingly appealing.
- **Transition Dynamics:** The amount of data required to reach the robust regime *increases* with the complexity gap $L_{\mathcal{L}}(p_{\text{robust}}) - L_{\mathcal{L}}(p_{\text{spur}})$, and *decreases* as the predictive advantage of robust features over spurious ones grows, which can be achieved for instance by increasing the diversity of training environments.

Scenario B: Robust vs. Bayes-Optimal solution In this scenario, the comparative advantage of the robust model is reversed: it is *not* necessarily the most predictive model, but stands as a simpler alternative to the more complex Bayes-optimal model p_{bayes} that uses all available latent features in the input x to infer the environment e and achieve minimal empirical risk.

- **Behavior:** If the Bayes-optimal solution is significantly more complex – for instance, because environment inference requires extracting complex background features – then at low N, the marginal predictive gain of p_{bayes} does not yet justify its higher description cost $L_{\mathcal{L}}(p_{\text{bayes}})$. However, as $N \to \infty$, the learner will inevitably transition to the best predictive model p_{bayes}, which may fail under OOD conditions such as unseen environments.
- **Transition Dynamics:** The amount of data required to transition *away* from the robust regime to the Bayes-optimal one *increases* with the complexity gap $L_{\mathcal{L}}(p_{\text{bayes}}) - L_{\mathcal{L}}(p_{\text{robust}})$, and *decreases* with the predictive advantage of environmental cues $D_{KL}(p^* \| p_{\text{robust}}) - D_{KL}(p^* \| p_{\text{bayes}})$.

Taken together, these two scenarios illustrate the **nuanced role of simplicity bias**, where the interplay between model complexity and predictive power determines the appropriate data regime for robustness. Specifically, Scenario A defines a lower bound N_{min} on the data required to overcome simplicity bias and rule out spurious shortcuts, while Scenario B

defines an upper bound N_{max} beyond which the learner will sacrifice robustness for the superior predictive power of a more complex, environment-dependent solution. Ultimately, this information-theoretic perspective underscores that simplicity bias is a double-edged sword: it can either hinder generalization by favoring shortcuts or promote it by ruling out overly complex environment-dependent hypotheses.

3 EXPERIMENTAL SETTING

In this section, we now aim to empirically test the hypothesis that neural networks behave as MDL-optimal compressors as posited in Section 2. Our goal is to test whether the data-regime transitions predicted by our theory, i.e. the crossover points where one hypothesis begins to yield a shorter total description length than another (see Figure 1b), line up with empirical shifts in feature reliance and generalization performance of a trained neural network. To this end, we first (i) introduce a semi-synthetic visual benchmark where we can precisely control aspects of the data-generating process, such as feature complexity and predictiveness. We then (ii) show how to estimate the total compression cost (3) of a candidate model p in practice. Finally, we (iii) present evaluation metrics for feature reliance used to confirm whether the learned model p_N relies on specific features as predicted by our theory.

3.1 BENCHMARK DESIGN

To validate our theoretical framework, we design a semi-synthetic visual task derived from Colored MNIST [Arjovsky et al., 2020] (see Figure C.1 for examples): The task is to predict whether a handwritten digit is smaller or greater than 5 where each sample is drawn from a mixture of discrete environments. Every input simultaneously carries three types of features: (i) the **digit shape**, the causal determinant of the label; (ii) an environment-dependent **color** applied to the digit that spuriously correlates with labels; and (iii) a **binary watermark** drawn from an environment-specific bank of K random patterns embedded in the rightmost pixel column of the image. The predictiveness of each feature is controlled by independent noise parameters, and the bank size K directly controls the complexity of the watermark feature. See Section C for the full data-generating procedure and the role of each parameter. By varying these knobs, we explore the full span of scenarios identified in Section 2:

- **Scenario A: Spurious vs. Robust solution** The spurious color signal acts as an easily exploitable shortcut; it is simpler to encode than the robust digit features ($L_{\mathcal{L}}(p_{\text{spur}}) < L_{\mathcal{L}}(p_{\text{robust}})$), despite usually being less predictive of the label. This specifically mirrors cases where the environment is not directly inferrable from the input, but a skewed marginal environment distribution $p(e)$ allows the learner to achieve decent predictive performance through a trivial feature (e.g., water $\Leftrightarrow$ water-bird). This allows us to study simplicity bias in its most common form: the preference for a trivial but non-robust signal over a more sophisticated causal mechanism.
- **Scenario B: Robust vs. Bayes-optimal solution** Watermarks (drawn from environment-specific banks) act as highly predictive features, but complex to learn since exploiting them requires memorizing an arbitrary number of distinct watermark patterns. Here, our theory predicts a protective effect: the learner is prevented from exploiting these signals until N is large enough to justify the high encoding cost of $L_{\mathcal{L}}(p_{\text{bayes}})$.

3.2 ESTIMATING THE TOTAL COMPRESSION COST

With the benchmark defined, we now describe how the MDL predictions are computed. The two-part MDL objective (3) decomposes into a fixed-cost and a variable cost, and one of the main challenges is to find tractable methods to estimate these two components for any p. We detail the estimation procedure for each component below.

Fixed cost: $L_{\mathcal{L}}(p)$ The candidate predictors p_{spur}, p_{robust}, and p_{bayes} introduced in Section 2.3 are abstract objects defined by the features they exploit. To obtain a concrete instantiation whose description length can be computed, we operationalize each candidate $p_{\texttt{feature}}$ as a neural network p_N trained on a custom dataset $\mathcal{D}_N$, where all variables except `feature` are randomly shuffled, which guarantees that only `feature` gets extracted in the process. When a model is obtained through a learning procedure, its description length $L_{\mathcal{L}}(p_N)$ can be tightly estimated via *prequential coding* [Blier and Ollivier, 2018]. In this view, simplicity relates to *ease of learning*: a simple model is one that can be learned quickly – with just a few samples. Its description length corresponds to the cumulative excess log-likelihood on unseen data incurred before the model has enough training data to converge, which requires N to be large enough in practice. We provide the full derivation in Section A.

Variable cost: $N \cdot \mathbb{E}_{(x,y)\sim p^*}[-\log p(y \mid x)]$ The variable cost corresponds to the expected number of bits required to encode labels sampled from the true distribution p^* using the model p, it corresponds to the remaining intrinsic randomness of the data that cannot be modeled by p. In practice, we compute this quantity using the empirical cross-entropy loss of the candidate model p evaluated on a held-out test set $\mathcal{D}_{\text{test}}$ sampled from p^*:

$$\mathbb{E}_{(x,y)\sim p^*}[-\log p(y \mid x)] \approx \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} -\log p(y \mid x)$$

Finally, for each candidate model p, it is also possible to estimate the fixed and variable costs of *intermediate models*

$\{p_1, \cdots, p_N\}$ that have not yet converged. We provide a detailed explanation in Section B. Intuitively, those intermediate models have absorbed fewer bits of structure from the data, yielding lower complexity but higher compression rate; together they define an *envelope* of compression lines.

3.3 EVALUATION METRICS

In order to detect the phase transition in the learning procedure, we develop a set of metrics designed to quantify the model's reliance on specific input features through evaluation on custom OOD datasets.

1. **Accuracy on held-out data**: We evaluate the model's performance across different data distributions. `Training` and `Validation` respectively track memorization and ID generalization ability of the trained model. We introduce `Digit`, `Color` and `Watermark` test sets, where the label correlates exclusively with a single `feature`, measuring the model's reliance on that specific `feature`. By breaking the correlation between the label and complementary features, these tests measure OOD generalization under targeted covariate shifts.
2. **Permutation feature importance**: We assess model reliance on each feature by randomly shuffling the values of a single feature of interest (e.g., digit, color, or watermark) across the test set, breaking its correlation with the label while preserving its marginal distribution. The *accuracy gap*, defined as the drop in accuracy between the original and permuted datasets, quantifies how much the model depends on that feature for prediction.

4 RESULTS

In Section 4.1, we present a side-by-side comparison of the MDL-optimal compression view and the empirical behavior of trained neural networks on Scenarios A and B, directly testing the core claim of the paper: that the feature transitions predicted by the compression envelope coincide with empirical shifts in feature reliance. In Section 4.2, we provide quantitative evidence that varying feature predictiveness and complexity impacts feature selection as predicted, supporting our central hypothesis.

4.1 DISSECTING A SINGLE EXPERIMENT

We begin by examining Scenarios A and B, where the same task is modeled both as a problem of optimal compression and as a problem of learning. For each scenario, we compare the preferences of an MDL-optimal compressor side-by-side with those of a trained neural network. The results are reported in Figure 2. We provide a detailed description of the experimental setup in Section D.

Compression view We first estimate the total description length for a set of candidate predictors and their intermediate interpolations, each corresponding to a distinct `feature` type. The fixed cost $L_{\mathcal{L}}(p_{\texttt{feature}})$ is estimated with prequential code length on a custom dataset as explained in Section 3.2, while the variable cost $N \cdot \mathbb{E}_{(x,y)\sim p^*}[-\log p(y \mid x)]$ is estimated via cross-entropy on the original test dataset. Plotted on a log-log scale (Figure 2a), the resulting *compression lines* reveal for any given dataset size N which solution locally yields the shortest description length and show the predicted transition point (vertical marker) where the MDL envelope shifts from one dominant feature type to another. In this envelope, each line corresponds to a model that extracts the corresponding feature with increasing accuracy: for example, there is a range of models that extract digit features, going from lossily extracting some simple geometric primitives to learning the full subtlety of handwriting.

Learning view In parallel, we train randomly initialized neural networks for different training dataset sizes N. We then evaluate them alongside the solutions selected by the MDL-optimal compressor on the set of metrics described in Section 3.1. These metrics reveal which feature those models actually rely on at each data regime (Figure 2b–c). We provide additional details about model class, optimizer and training in Section E.

Alignment Comparing both views, we highlight two findings. First, the *hard transition* predicted by the compression envelope – i.e. the dataset sizes at which the MDL-optimal solution switches from one dominant feature to another – coincide precisely with the empirical transition in feature reliance. Second, while the OOD performance of the theoretical solution matches that of the neural network on the dominant feature, our current framework evaluates candidate models that exploit strictly disjoint subsets of features. Consequently, the idealized MDL solution exhibits discrete transitions in feature reliance (Figure 2b), entirely discarding a previously favored feature. In practice, however, neural networks shift continuously between features. In the data regime surrounding a transition, there may exist a hybrid model that partially exploits the complex feature while still leveraging residual signal from the simpler one, leading to a gradual crossover around the predicted transition point.

4.2 VARYING TASK CHARACTERISTICS

In this second part, we now study the effect of varying key task characteristics such as feature complexity and correlation strength on the feature selection behavior of neural networks, and compare the observations with the predictions derived from our theory. Results are summarized in Figure 3.

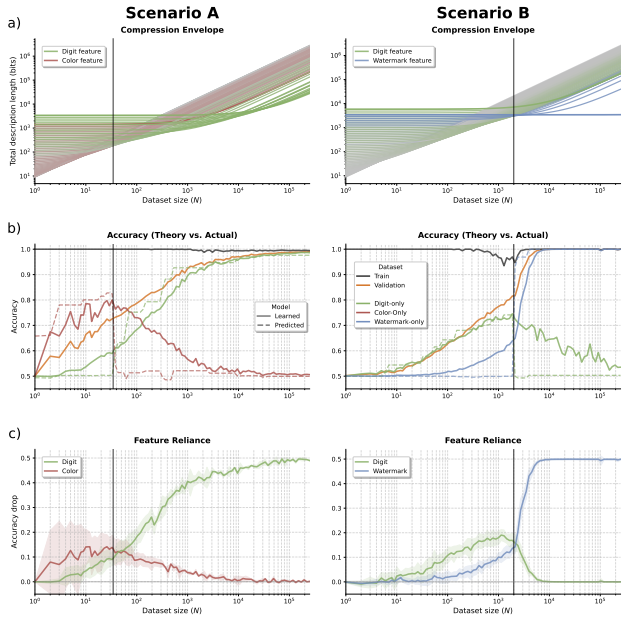


Figure 2: Comparison of the compression view and the learning view for Scenario A and B. **(a)** Compression envelopes on a log-log scale: each line shows the total two-part description length of a candidate predictor as a function of dataset size N. **(b)** Accuracy on held-out datasets as a function of N: solid lines are empirical measurements, dashed lines are reported metrics of the MDL-optimal solution. **(c)** Permutation feature importance (accuracy drop after shuffling each feature): a higher drop indicates stronger reliance on that feature.

For each configuration, we extract two scalar statistics:

- **Theoretical transition** N_{theory}: the dataset size at which the MDL compression envelope switches between different feature types, i.e. the point where the two-part codelength of one feature-based predictor falls below that of the competing one.
- **Empirical transition** $N_{\text{empirical}}$: the dataset size at which the trained neural network's dominant feature reliance switches, detected as the crossover point where the accuracy gap of one feature overtakes the other.

Effect of feature predictiveness: In both scenarios, reducing the predictiveness of a feature lowers its compression rate advantage over the competing solution, causing the transition to occur earlier. In Scenario A, we vary the noise on the spurious color/label correlation: a noisier spurious feature increases its per-sample bit-cost, so the idealized learner abandons it sooner and N_{theory} decreases (Figure 3a). Symmetrically, in Scenario B, as the predictiveness of the robust model decreases, the gap in compression rate between the robust and the Bayes-optimal model increases. This widening gap rapidly eclipses the robust model's initial cost advantage, causing the transition point N_{theory} to decrease (Figure 3a).

Effect of feature complexity: Finally, in Scenario B we vary the number of distinct watermark patterns the model must learn to exploit the complex watermark/label correlation, which necessarily inflates $L_{\mathcal{L}}(p_{\text{bayes}})$. Concretely, larger banks should delay the transition away from the robust regime: N_{theory} increases (Figure 3b).

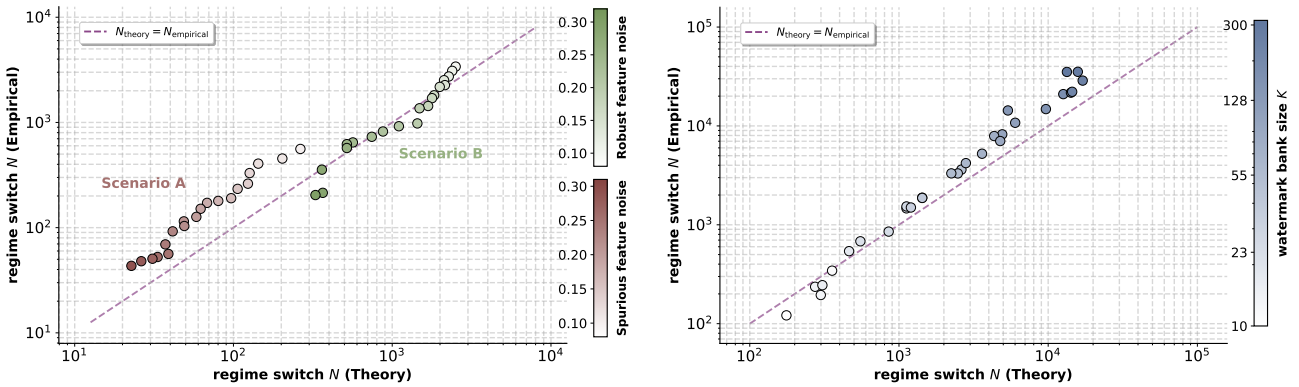
Results show that N_{theory} and $N_{\text{empirical}}$ correlate well across all configurations, with a Pearson correlation of 0.976, especially considering the noise in the estimation of the crossover points and model complexity $L_{\mathcal{L}}(p)$. This supports that our MDL-based theory captures the main factors driving feature selection across data regimes.

5 DISCUSSION

Simplicity and generalization As we noticed in the experiment Section, simplicity bias does not necessarily promote nor prevent generalization. Paradoxically, in low-data regimes, it can encourage overfitting and reliance on spurious shortcuts as the 'most compact' explanation for the finite set of samples[2] (Figure 2). Importantly, this is not a failure of the learner: from the MDL perspective, memorization is the *rational* solution when the available data are too few to justify the description cost of any structured model. As the number of samples grows, however, the cumulative extra-cost in bits of memorizing samples or using a suboptimal model p becomes increasingly unbearable, eventually forcing the learner to switch to more predictive competing models, regardless of their inherent complexity.

A complexity-based regularization An important implication arises when the robust model is not the most predictive candidate – a common scenario where spurious features are highly predictive within the training distribution. In such cases, a more complex Bayes-optimal model incorporating these non-robust features would achieve a lower compression rate asymptotically. However, our theory predicts that a simplicity-seeking learner will not select this complex environment-dependent model while the dataset size N remains below a theoretical threshold: as long as the marginal predictive gain of the Bayes-optimal solution does not offset its higher description cost $L_{\mathcal{L}}(p)$. This is precisely what we observe in Scenario B, where reliance on the robust digit feature (Figure 2c, green line) peaks in the intermediate data regime before the model abandons it in favor of the more predictive but more complex Bayesian solution. Paradoxically, **limiting the amount of training data can act as a form of complexity-based regularization**: a constrained data budget keeps the learner in the regime where simpler

[2] although the No Free Lunch theorem suggests that no learning algorithm likely provides generalization guarantees in such low-data regimes



(a) Effect of feature predictiveness

(b) Effect of feature complexity

Figure 3: Empirical vs. theoretical transition point N for varying task characteristics. Each point is one experimental configuration; the dashed line is the identity $N_{\text{theory}} = N_{\text{empirical}}$. **(a)** Reducing the predictiveness of a feature increases its compression rate disadvantage, causing the transition to occur earlier. **(b)** Increasing the complexity of a feature inflates its description cost, causing the transition to occur earlier. In both cases, the proximity to the diagonal indicates that our MDL-based theory accurately predicts the observed feature switch as measured by our proxy.

robust mechanisms are favored, because the more predictive alternatives remain unaffordable from a compression standpoint. Finally, our framework also suggests a compression-theoretic rationale for pretraining: through unsupervised exposure to diverse environments, a pretrained model has already absorbed, for free, bits of structure about the data into its weights, which lower the operational description cost $L_{\mathcal{L}_{\text{pretrain}}}(p_{\text{robust}})$ before fine-tuning begins, making complex solutions accessible at smaller dataset sizes [Hendrycks et al., 2019].

6 RELATED WORK

This paper adopts an information-theoretic perspective on machine learning, grounded in the established equivalence between statistical learning and lossless compression [Shannon, 1948, Rissanen, 1978, Chaitin, 2002]: to *comprehend* the underlying structure of a dataset is identical to *compressing* that dataset into a more efficient representation. This duality has birthed an information-theoretic paradigm for model selection with applications to robust machine learning: the Minimum Description Length (MDL) principle [Grünwald, 2007].

This paper relates most closely to a line of papers that empirically analyze the tendency of NNs to prefer simple features for prediction, referring to this phenomenon as simplicity bias [Rahaman et al., 2019, De Palma et al., 2019], and their implications for ID and OOD generalization. The simplicity bias of neural networks presents a fundamental duality: while it is an instrumental mechanism for ID generalization, it is simultaneously the root cause of systemic failure of OOD generalization. On the one hand, SB relates to robust ID performance by favoring the discovery of underlying patterns and preventing memorization or the learning of redundant features [Arpit et al., 2017, Valle-Pérez et al., 2018, Pezeshki et al., 2021, Mingard et al., 2025], aligning with Occam's razor. On the other hand, this same preference for simplicity often compels models to rely on spurious correlations and low-level shortcuts [Geirhos et al., 2018, 2020, Shah et al., 2020, Pezeshki et al., 2021]. Because these non-robust rules are typically "simpler" to optimize, NNs remain vulnerable to adversarial samples that exploit spurious covariate shifts [Teney et al., 2022, Vasudeva et al., 2023].

7 LIMITATIONS & FUTURE WORK

Our framework currently considers candidate models that exploit strictly disjoint subsets of features, some of which are perfectly predictive. While this idealization enables a clean theoretical analysis, real-world tasks typically involve multiple *partially* predictive features that contribute to label prediction, interact with one another, and may not be neatly separable. In such settings, the compression trade-off is richer: the MDL-optimal learner may blend several features simultaneously rather than transition sharply between them, and the notion of a single "dominant" feature may not hold. Extending the framework to predictors that handle multiple partially predictive features would bring the theory closer to realistic settings and enable the study of more nuanced generalization failure modes.

Acknowledgements

The authors would like to thank Avery Hee-Woon Ryoo for their useful comments. TM acknowledges support from

the Fonds de recherche du Québec (FRQNT 0000364322, https://doi.org/10.69777/0000364322). This research was enabled in part by computing resources and support provided by Mila – Quebec AI Institute.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization, 2020. URL https://arxiv.org/abs/1907.02893.

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 233–242, 2017.

Léonard Blier and Yann Ollivier. The description length of deep learning models. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

Gregory J. Chaitin. On the intelligibility of the universe and the notions of simplicity, complexity and irreducibility. *arXiv preprint math/0210035*, 2002.

Giacomo De Palma, Bobak Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2018.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Micah Goldblum, Marc Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, Kolmogorov complexity, and the role of inductive biases in machine learning. *arXiv preprint arXiv:2304.05366*, 2023.

Peter D. Grünwald. A tutorial introduction to the minimum description length principle. advances in minimum description length: Theory and applications, 2004.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pages 2712–2721. PMLR, 2019.

Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A. Louis. Deep neural networks have an inbuilt Occam's razor. *Nature Communications*, 16(1):220, 2025.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C. Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:1256–1272, 2021.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 5301–5310, 2019.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8346–8356. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sagawa20a.html.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9573–9585, 2020.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

Damien Teney, Ehsan Abbasnejad, Simon Lucey, and Anton Van den Hengel. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior OOD generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16761–16772, 2022.

Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

Bhavya Vasudeva, Kameron Shahabi, and Vatsal Sharan. Mitigating simplicity bias in deep learning for improved OOD generalization and robustness. *arXiv preprint arXiv:2310.06161*, 2023.

Andrew Gordon Wilson. Position: Deep learning is not so mysterious or different. In *Forty-second International Conference on Machine Learning (ICML) Position Paper Track*, 2025.

A PREQUENTIAL CODING DETAILS

Prequential coding is a category of universal coding schemes that builds on the idea of sequentially using past data to encode future data as it arrives. In this scheme, assuming the prior knowledge of a learning algorithm $\mathcal{L}$ (architecture, optimizer, initialization scheme, etc.), we recursively encode a label y_i using entropy coding derived from a parametric conditional model $p_i(\cdot \mid x_i)$ fitted on the already transmitted labeled data $\mathcal{D}_{1:i-1}$. The total cost of encoding data with prequential coding is given by:

$$L_{\text{preq}}(\mathcal{D}_N; \mathcal{L}) = \sum_{i=1}^{N} -\log p_i(y_i \mid x_i)$$

With $p_i \leftarrow \mathcal{L}(\mathcal{D}_{1:i-1})$ and p_1 an initial predictor defined by $\mathcal{L}$. At the end of the procedure, both the sender and the receiver share the knowledge of $\mathcal{D}_{1:N-1}$. A final model p_N can then be fitted by the receiver using $\mathcal{L}$ at *zero additional cost*. In this perspective, prequential coding is a program that jointly encodes the data $\mathcal{D}_{1:N-1}$ and the final predictive model p_N. We then estimate the description length of p_N alone by rewriting the prequential codelength as follows:

$$L_{\text{preq}}(\mathcal{D}_N; \mathcal{L}) = \underbrace{\sum_{i=1}^{N} -\Big(\log p_i(y_i \mid x_i) - \log p_N(y_i \mid x_i)\Big)}_{\text{Excess loss}} + \underbrace{\sum_{i=1}^{N} -\log p_N(y_i \mid x_i)}_{\text{Asymptotic Loss}}$$

Intuitively, the cumulative excess loss incurred before the predictor has converged corresponds to bits spent on describing the final model p_N through $L_{\mathcal{L}}$. In this regard, a simple model is a model that is learned *quickly*, i.e., that requires fewer samples to be learned. Visually, the description length of the final model is the "area" under the curve above the asymptotic loss:

$$L_{\mathcal{L}}(p_N) \approx \sum_{i=1}^{N} \Big(-\log p_i(y_i \mid x_i) + \log p_N(y_i \mid x_i)\Big)$$

Block-wise approximation. Computing the exact prequential codelength would require retraining a model from scratch for every new sample. To make this tractable with neural networks, we use a block-wise approximation. The idea is to only retrain the model at a sparse set of dataset sizes and use the resulting predictor to encode the following block of samples.

Let $1 = t_0 < t_1 < \cdots < t_S = N$ be a sequence of block boundaries. In practice we use exponentially increasing block sizes to minimize the estimation error since training loss usually correlates log-linearly with the amount of training data. At each boundary t_s, we train a randomly initialized model on the already transmitted data $\mathcal{D}_{1:t_s-1}$. We then use this model to encode the next block $\mathcal{D}_{t_s:t_{s+1}-1}$.

Formally, let $p^{(s)}$ be the predictor trained on $\mathcal{D}_{1:t_s-1}$ for $s \geq 1$ and $p^{(0)}$ the untrained predictor from $\mathcal{L}$. The block-wise prequential description length is:

$$L_{\text{preq}}(\mathcal{D}_N; \mathcal{L}) = \sum_{s=0}^{S-1} \sum_{i=t_s}^{t_{s+1}-1} -\log p^{(s)}(y_i \mid x_i), \tag{7}$$

B INTERMEDIATE MODELS AND THE COMPRESSION ENVELOPE

The construction of the PCL curve in Section 3.2 yields, for each feature type, a single fully-converged predictor p_N with complexity $L_{\mathcal{L}}(p_N)$ and asymptotic loss $\ell_N = \mathbb{E}_{(x,y)\sim p^*}[-\log p_N(y \mid x)]$. However, behind each fully-converged predictor, lies a whole family of intermediate predictors that rely on *some* partial information about the feature (e.g. only the low-frequency geometric patterns of the digit feature). In practice, nothing prevents the learner from selecting one of those intermediate predictors instead of the fully-converged one. In fact, for the picture to be complete, we need to consider these intermediate predictors as potential candidates for compression. To this end, we derive a set of intermediate models by truncating the PCL curve for exponentially large AUC.

Let $\ell(n)$ denote the prequential test loss of a feature-specific predictor trained on n samples, and let $\ell_{\text{orig}}(n)$ denote its cross-entropy on the original distribution at training size n. For a truncation point $N_t \leq N$, we define the intermediate model p_{N_t} with:

- **Model complexity** (fixed-cost):

$$L_{\mathcal{L}}(p_{N_t}) \approx \int^{N_t} \left[\ell(n) - \ell(N_t)\right] dn$$

- **Compression rate** (variable-cost): $\ell_{\text{orig}}(N_t)$.

As we approach the fully-converged model, the complexity increases while the compression rate decreases, and as $N_t \to N$, we recover the fully-converged model. Each intermediate model defines an affine compression line $\mathcal{J}(p_{N_t}, N) = L_{\mathcal{L}}(p_{N_t}) + N \cdot \ell_{\text{orig}}(N_t)$, and the MDL-predicted feature at dataset size N is determined by the *lower envelope* over all intermediate models across all feature types:

$$\hat{p}_N = \underset{p_{N_t}^{(f)},\, f \in \mathcal{F}}{\arg\min} \left[L_{\mathcal{L}}(p_{N_t}^{(f)}) + N \cdot \ell_{\text{orig}}^{(f)}(N_t) \right]$$

where $\mathcal{F}$ indexes the set of candidate feature types. The feature-type transitions reported in our experiments correspond to the points where this envelope switches between models derived from different feature types.

C TASK DEFINITION AND SAMPLE GENERATION

Benchmark design: Given a handwritten digit image x, the task consists in predicting whether the digit value is $d \geq 5$ (class $y = 1$) or $d < 5$ (class $y = 0$). We use the EMNIST digits dataset, which extends MNIST with a total of $280,000$ samples. Prequential code length being computationally very expensive to evaluate, we intentionally keep the visual task simple (binary classification on 32×32 images) due to limited compute resources.

Each sample (x, y, e) is generated according to this procedure:

1. **Binary label assignment (robust feature):** Given the original digit value $d \in \{0, \ldots, 9\}$, the label is assigned as $y = \mathbb{1}[d \geq 5] \oplus \text{Bernoulli}(p_{\text{flip}})$, where $\oplus$ denotes the XOR operation. With probability p_{flip}, the label is flipped, introducing noise in the digit-label relationship. The digit shape is therefore the causal determinant of the label, and p_{flip} controls how predictive this robust feature is.
2. **Environment assignment:** Each sample is assigned to one of two environments using $e \sim \text{Bernoulli}(p_e)$, where p_e controls the environment proportion.
3. **Color assignment (spurious feature):** The digit is colorized based on environment and label. In **Environment 0**, $y = 0 \to$ green and $y = 1 \to$ red. In **Environment 1**, $y = 0 \to$ red and $y = 1 \to$ green. The color thus creates a low-complexity spurious correlation with y; its predictive strength is governed by the environment imbalance p_e. When $p_e = 0.5$, the two environments are balanced and the marginal color–label correlation vanishes. The further p_e is from 0.5, the stronger the environment imbalance, and the stronger the marginal spurious correlation between color/digit becomes. Thus, p_e controls the predictiveness of the color feature.
4. **Watermark assignment (complex environmental cue):** A binary watermark pattern is embedded in the rightmost pixel column of each image, encoding the environment through two non-overlapping banks of K random patterns (one per environment). Exploiting this feature requires memorizing all $2K$ patterns, so the bank size K serves as a direct control knob on the complexity of this environmental signal. Details of the watermark generation are given below.

Watermark generation We pre-generate two watermark banks $\mathcal{B}_0$ and $\mathcal{B}_1$, each containing B unique binary vectors of length b bits (where $b = 32$ matches the image height). The two banks are guaranteed to have no overlap, i.e. $\mathcal{B}_0 \cap \mathcal{B}_1 = \emptyset$. Each pattern is sampled uniformly at random from $\{0, 1\}^b$, with uniqueness enforced via rejection sampling. For each sample with environment $e \in \{0, 1\}$, a watermark is drawn uniformly at random from the corresponding bank: $\mathbf{b} \sim \text{Uniform}(\mathcal{B}_e)$, and the rightmost pixel column of the image is set to $\mathbf{b}$.

The **bank size** B directly controls the complexity of the watermark feature: a model exploiting watermarks must memorize all $2B$ distinct patterns and their environment assignments. Small values of B (e.g., $B = 2$) yield a simple spurious feature that is easy to memorize, while large values (e.g., $B = 50$) produce a complex feature requiring substantially more model capacity to exploit.

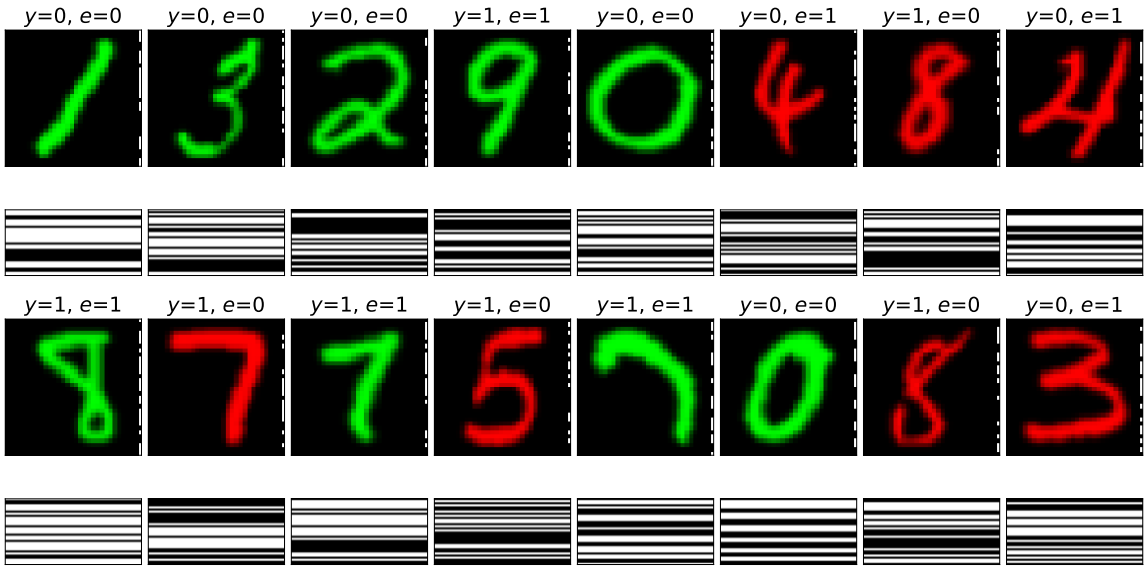


Figure C.1: Example samples from the EMNIST-based benchmark we curated, featuring the three types of features we presented in Section 2.3: the digit shape (robust feature), the color (simple spurious feature) and the watermark (complex environmental-cue). The label is determined by the digit value, while the color and watermark are correlated with the label in an environment-dependent way.

C.1 TASK EXAMPLES

D QUALITATIVE STUDY: EXPERIMENTAL DETAILS

The side-by-side comparison in Figure 2 considers different configurations of Color-MNIST:

Scenario A $p_e = 0.25$, $p_{\text{flip}} = 0$, no watermark. The color feature is simpler but less predictive than the digit feature. The theory predicts that color dominates at small N before the learner switches to digit as the suboptimal compression rate becomes prohibitive.

Scenario B $p_e = 0.5$, $p_{\text{flip}} = 0.15$, watermark bank size $K = 50$. Balanced environments eliminate the marginal color-label correlation. The digit feature is simpler but noisy (85% ceiling). The theory predicts digit dominance at intermediate N, with a transition to the watermark-based model at large N.

E TRAINING PROCEDURE AND HYPERPARAMETERS

We train neural networks using standard Empirical Risk Minimization (ERM) with AdamW. ERM aims to minimize the cross-entropy loss on the training data, which corresponds to minimizing the negative log-likelihood of the model's predictions. For all experiments, we use the following training configuration (see Table E.1 for a summary):

Architecture Our predictive model is a simple Multi-Layer Perceptron (MLP) with a featurizer consisting of 2 hidden layers of dimension 256, followed by a linear classification head. The featurizer processes flattened 32×32 RGB images into 3072-dimensional vectors, uses ReLU activations, and Xavier (Glorot) uniform initialization. We apply no dropout. While our theory is architecture-agnostic, we restrict this study to simple MLPs and toy datasets due to the extreme computational cost of prequential code length estimation (each point in the PCL curve requires training a model from scratch until convergence)

Optimization We use AdamW optimizer with learning rate $\eta = 10^{-3}$ and L_2 weight decay (regularization) $\lambda = 10^{-4}$. The batch size is set to $B = 64$ for all experiments. We train until convergence using early stopping with a patience of 3 epochs and a minimum improvement threshold of $\Delta = 5 \times 10^{-4}$.

Data Sampling For each dataset size N, we train the model on a random subset of size N sampled from the full training set. To reduce variance in estimates for small dataset sizes ($N \leq 500$), we perform 10 independent runs with different random subsets and seeds, averaging the resulting metrics over the runs. For larger datasets ($N > 500$), we use 3 independent runs per dataset size.

Implementation All experiments are implemented in Python 3.12 using PyTorch 2.9. Training is performed on NVIDIA GPUs with CUDA 11.8. The source code is available in the `complicity` repository.

Table E.1: Hyperparameters for the Watermarked CMNIST Experiments

Parameter	**Symbol**	**Description**	**Values**
Environment & Spurious Correlation			
`flip_prob`	p_{flip}	Label noise probability	$\{0, 0.01, \ldots, 0.1\}$
`spur_prob`	p_e	Relative proportion of environments	$\{0, 0.02, \ldots, 0.2\}$ or 0.5
`uninformative_majority`	—	Randomize color in majority env	`True`, `False`
Watermark Configuration			
`watermark_bank_size`	K	Distinct patterns per environment	1 to 100 (log-spaced)
`watermark_bits`	b	Length of each binary pattern	32
`random_watermark`	—	Use random (non-informative) watermark	`True`, `False`
Image Configuration			
`input_size`	—	Image resolution	32×32
`Digit`	—	Remove color information	`True`, `False`
`noise_digit`	—	Replace digit with Gaussian noise	`True`, `False`
Model Architecture (MLP)			
`n_outputs`	h	Hidden layer dimension	256
`n_layers`	L	Number of hidden layers	2
`mlp_dropout`	p_{drop}	Dropout probability	0.0
Optimization			
`optimizer`	—	Optimization algorithm	AdamW
`lr`	η	Learning rate	10^{-3}
`weight_decay`	λ	L2 regularization	10^{-4}
`batch_size`	B	Mini-batch size	64