Rashomon Capacity: A Metric for Predictive Multiplicity in Probabilistic Classification

Hsiang Hsu* and Flavio du Pin Calmon*

Abstract

Predictive multiplicity occurs when classification models with nearly indistinguishable average performances assign conflicting predictions to individual samples. When used for decision-making in applications of consequence (e.g., lending, education, criminal justice), models developed without regard for predictive multiplicity may result in unjustified and arbitrary decisions for specific individuals. We introduce a new measure of predictive multiplicity in probabilistic classification called Rashomon Capacity. Prior metrics for predictive multiplicity focus on classifiers that output thresholded (i.e., 0-1) predicted classes. In contrast, Rashomon Capacity applies to probabilistic classifiers, capturing more nuanced score variations for individual samples. We provide a rigorous derivation for Rashomon Capacity, argue its intuitive appeal, and demonstrate how to estimate it in practice. We show that Rashomon Capacity yields principled strategies for disclosing conflicting models to stakeholders. Our numerical experiments illustrate how Rashomon Capacity captures predictive multiplicity in various datasets and learning models, including neural networks. The tools introduced in this paper can help data scientists measure, report, and ultimately resolve predictive multiplicity prior to model deployment.

Keywords: Rashomon effect, Rashomon set, predictive multiplicity, channel capacity, Rashomon capacity, probabilistic classifier.

1 Introduction

Rashomon effect, introduced by Breiman (2001), describes the phenomenon where a multitude of distinct predictive models achieve similar training or test loss. Breiman reported observing the Rashomon effect in several model classes, including linear regression, decision trees, and small neural networks. In a foresighted experiment, Breiman noted that, when retraining a neural network 100 times on three-dimensional data with different random initializations, he “found 32 distinct minima, each of which gave a different picture, and having about equal test set error” (Breiman, 2001, Section 8). The set of almost-equally performing models for a given learning problem is called the Rashomon set (Fisher et al., 2019; Semenova et al., 2019).

The Rashomon effect has important—and potentially harmful—implications. We focus on a facet of the Rashomon effect in classification problems called predictive multiplicity. Predictive multiplicity occurs when competing models in the Rashomon set assign conflicting predictions to individual samples (Marx et al., 2020). Fig. 1 presents an updated version of Breiman’s neural network experiment and illustrates predictive multiplicity in three classification tasks with different data domains and neural network architectures. Here, models that achieve statistically-indistinguishable performance on a test set assign wildly different predictions to an input sample. If predictive

*H. Hsu and F. P. Calmon are with the John A. Paulson School of Engineering and Applied Sciences, Harvard University, Boston, MA 02134. E-mails: hsianghsu@g.harvard.edu, flavio@seas.harvard.edu.
Predictive multiplicity occurs on different data domains and learning models, including an image dataset (CIFAR-10 (Krizhevsky et al., 2009)) trained with VGG16 (Simonyan and Zisserman, 2014), a natural language dataset (AG News (Zhang et al., 2015)) trained with a simple neural networks after tokenization, and an audio dataset (UrbanSound8k (Salamon et al., 2014)) trained with LSTM (Greff et al., 2016).

Multiplicity is not accounted for, the output for this sample may ultimately depend on arbitrary choices made during training (e.g., parameter initialization).

Predictive multiplicity captures the potential individual-level harm introduced by an arbitrary choice of a single model in the Rashomon set. When such a model is used to support automated decision-making in sectors dominated by a few companies or Government—labeled Algorithmic Leviathans in (Creel and Hellman, 2021, Section 3)—predictive multiplicity can lead to unjustified and systemic exclusion of individuals from critical opportunities. For example, an algorithm used for lending may deny a loan to a specific applicant. However, during model development, there may have been a competing model which performs equally well on average, yet would have approved the loan for this individual. As another example, Governments are increasingly turning to algorithms for grading exams that grant access to higher-level education (see, e.g., UK (Smith, 2020) and Brazil (INEP, 2020)). Here, again, accounting for predictive multiplicity is critical: an arbitrary choice of a single model in the Rashomon set may lead to an unwarranted restriction of educational opportunities to an individual student. In applications such as criminal justice and healthcare, models that do not account for predictive multiplicity are also at risk of causing arbitrary individual-level harm.

We introduce new methods for measuring, reporting, and resolving predictive multiplicity for probabilistic classifiers. First, we postulate several properties that a predictive multiplicity metric must satisfy to simplify its interpretation by stakeholders. We then provide a new predictive multiplicity metric called Rashomon Capacity. Unlike prior metrics based on thresholded (i.e., 0 or

---

1A probabilistic classifier is a model that maps an input sample onto a probability distribution, referred as a score, over a discrete set of classes. Examples of probabilistic classifiers include logistic regression, random forests, and a neural network with a softmax output layer.
predictions, Rashomon Capacity captures more nuanced variations in scores among models in the Rashomon set for a target input sample. We derive properties of Rashomon Capacity and describe how it can be computed. Our numerical examples provide a methodology for reporting predictive multiplicity in probabilistic classifiers trained on a variety of datasets.

Remarkably, the computation of Rashomon Capacity also sheds light on a strategy for resolving predictive multiplicity. Different approaches have been proposed for dealing with multiplicity, including randomizing between competing classifiers (Creel and Hellman, 2021) and bagging (Breiman, 2001). However, the size of the Rashomon set can be large (see Semenova et al. (2019) for an example with 250k models), making strategies that require randomizing predictions across the entire Rashomon set potentially arbitrary and impractical. In contrast, we apply standard results from convex analysis to show that the predictive multiplicity for an input sample—as measured by Rashomon Capacity—can be entirely captured by at most \( c \) models in the Rashomon set, where \( c \) is the number of predicted classes. This result holds regardless of the size of the Rashomon set. Thus, when \( c \) is small, the predictions produced by the competing classifiers can be communicated to a stakeholder, empowering them to decide how to resolve conflicting scores. Note that the \( c \) models that capture predictive multiplicity can be different across samples. Consequently, we also provide a greedy algorithm for identifying a subset of models in the Rashomon set that captures most of the score variations across a dataset. In summary, our main contributions are:

1. We introduce a new score-based metric for quantifying predictive multiplicity in probabilistic classifiers called Rashomon Capacity. We provide a principled derivation of this metric and demonstrate how it can be computed in practice.

2. We describe a methodology for reporting predictive multiplicity in probabilistic classifiers using Rashomon Capacity. We give examples of this methodology using different datasets and learning models. Rashomon Capacity must be reported to stakeholders in, for example, model cards (Mitchell et al., 2019).

3. We propose a procedure for resolving predictive multiplicity in probabilistic classifiers. Even though the Rashomon set may span a large (potentially uncountable) number of models, we show that the score variation for a sample is fully captured by a small, discrete subset of models in the Rashomon set. Communicating these predictions to stakeholders can empower them to decide how to resolve predictive multiplicity.

2 Background and Related Work

The Rashomon effect impacts model selection (Rudin, 2019; Hancox-Li, 2020; D’Amour et al., 2020), explainability (Pawelczyk et al., 2020), and fairness (Coston et al., 2021). Rudin (2019) suggested that, given the choice of competing models, machine learning (ML) practitioners should select interpretable models a priori, rather than selecting a black-box model with conjectural explanations post-training. Hancox-Li (2020) and D’Amour et al. (2020) further argued that epistemic patterns, e.g., causality, should be specified in the ML pipeline, and the selected models from the Rashomon set should be able to reflect these patterns. Competing models in the Rashomon set may not only render conflicting explanations for predictions (Pawelczyk et al., 2020) and measures of feature importance (Fisher et al., 2019), but also have inconsistent performance across population subgroups. Consequently, the arbitrary choice of a single model without regard for the Rashomon effect may result in unnecessary and discriminatory bias against vulnerable population groups (Coston et al., 2021). See Appendix B.1 for further discussion, including connections with individual fairness (Dwork and Ilvento, 2018).
We outline the notation used in the paper next, then introduce existing metrics for predictive multiplicity and discuss their limitations. We consider a dataset \( D = \{(x_i, y_i)\}_{i=1}^n \), e.g., a training or test set, for a classification task with \( c \) classes/labels, where each sample pair \((x_i, y_i)\) is drawn i.i.d. from \( P_{X,Y} \) with support \( X \times \Delta_c \). Here, \( \Delta_c \) denotes the \( c \)-dimensional probability simplex, i.e., each \( y_i \) is one-hot encoded. Let \( e_k \) be a length-c indicator vector with one in the \( k \)th position and zero elsewhere, i.e., \([e_k]_j = 1\) and \([e_k]_j = 0 \forall j \neq k\), where \([\cdot]_j\) denotes the \( j \)th entry of a vector. Thus, we have \( y_i \in \{e_k\}_{k=1}^{c} \) and \( \Delta_c = \text{span}(e_1, e_2, \ldots, e_c) \). \( 1(\cdot) \) denotes the indicator function.

We denote by \( \mathcal{H} \) a hypothesis space, i.e., a set of candidate probabilistic classifier is parameterized by \( \theta \in \Theta \subseteq \mathbb{R}^d \) that approximate \( P_{Y|X=x_i} \), i.e., \( \mathcal{H} \triangleq \{h_\theta : \mathcal{X} \to \Delta_c : \theta \in \Theta\} \). The loss function used to evaluate model performance is denoted by \( \ell : \Delta_c \times \Delta_c \to \mathbb{R}^+ \) (e.g., cross-entropy) and \( L(\theta) \triangleq \mathbb{E}_{P_{X,Y}}[\ell(h_\theta(X), Y)] \) the population risk\(^2\). As usual, the population risk is approximated by the empirical risk \( \hat{L}(\theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(h_\theta(x_i), y_i) \). The definitions in the rest of this paper are cast in terms of empirical risk, and their population risk versions can be defined similarly.

\[ R(\mathcal{H}, \epsilon) \triangleq \{h \in \mathcal{H} : \hat{L}(h) \leq \epsilon\}. \] (1)

Note that the Rashomon set is determined by the hypothesis space, the Rashomon parameter, and also implicitly by the dataset due to the evaluation of \( \hat{L}(h) \). The cardinality \( |R(\mathcal{H}, \epsilon)| \) or the volume\(^4\) \( \text{vol}(R(\mathcal{H}, \epsilon)) \) of the Rashomon set (depending on whether \( R(\mathcal{H}, \epsilon) \) is finite) can then be used to quantify the size of the Rashomon set.

Estimating \( \text{vol}(R(\mathcal{H}, \epsilon)) \) is essentially a level set estimation problem (Mason et al., 2021), and is computationally infeasible when the hypothesis space \( \mathcal{H} \) is large (Bachoc et al., 2021). Semenova et al. (2019, Section 5.1) provide an exact computation of \( \text{vol}(R(\mathcal{H}, \epsilon)) \) on ridge regression (where the Rashomon set forms an ellipsoid), and suggest rejection-sampling-with-replacement and importance sampling of the Rashomon set when an exact form is not feasible (Semenova et al., 2019, Section 5.3). However, there is no sample complexity guarantee for the sampling methods provided, and a significant amount of models might be needed to estimate the Rashomon volume. For example, in (Semenova et al., 2019, Section 6), 250k decision trees are sampled in order to compute \( \text{vol}(R(\mathcal{H}, \epsilon)) \).

Given \( R(\mathcal{H}, \epsilon) \), the Rashomon ratio is defined as \( \mathcal{R}(\mathcal{H}, \epsilon) \triangleq \frac{\text{vol}(R(\mathcal{H}, \epsilon))}{\text{vol}(\mathcal{H})} \). \( \mathcal{R}(\mathcal{H}, \epsilon) \) represents the fraction of models in the hypothesis space that fit the data about equally well. Semenova et al. (2019) suggested that when the Rashomon ratio is large, models with various desirable properties, such as better generalizability, can often exist inside the Rashomon set.

For predictive multiplicity in classification problems, Semenova et al. (2019, Defn. 12) further proposed pattern Rashomon ratio for binary classification, which is the ratio of the count of all possible binary predicted classes given by the functions in the Rashomon set to that given by the functions in the hypothesis space. When the size of the hypothesis space goes to infinity, the pattern Rashomon ratio can be upper and lower bounded by the binary entropy function (Semenova et al.,

---

\(^2\)We use the notations \( L(h) \triangleq \mathbb{E}_{P_{X,Y}}[\ell(h(X), Y)] \) and \( L(\theta) \triangleq \mathbb{E}_{P_{X,Y}}[\ell(h_\theta(X), Y)] \) interchangeably, depending on whether we intend to emphasize the parameter space \( \Theta \).

\(^3\)When given an empirical risk minimizer \( \hat{\theta} \in \arg \min_{\theta \in \Theta} \hat{L}(h_\theta) \), one can set \( \epsilon = \hat{L}(h_{\hat{\theta}}) + \epsilon' \).

\(^4\)We can define a monotonic function \( \text{vol} : \mathcal{H} \to \mathbb{R}^+ \) for the volume of \( R(\mathcal{H}, \epsilon) \) in the hypothesis space. When \( \mathcal{H} \) is parameterized by \( \Theta \subseteq \mathbb{R}^d \), the volume of \( \{\theta \in \Theta : h_\theta \in R(\mathcal{H}, \epsilon)\} \) can be directly computed in \( \mathbb{R}^d \); however, computing the volume of a set in high-dimensional space is hard even if its analytic form is known.
Note that the computational complexity of the pattern Rashomon ratio grows exponentially with the number of samples, and could be an “expensive” metric for predictive multiplicity when applied on a large dataset. Moreover, even for a simplest logistic regression, the exact form of the pattern Rashomon ratio is not tractable due to the non-linearity of the maximum likelihood ratio (Hastie et al., 2009).

2.2 Ambiguity and Discrepancy of Models in a Rashomon Set

In an important paper (and the work closest to ours), Marx et al. (2020) motivated predictive multiplicity and proposed two metrics, ambiguity and discrepancy, for quantifying predictive multiplicity in binary classification problems. Marx et al. (2020) measure multiplicity in terms of the thresholded outputs (i.e., predicted classes) of a classifier, defining the Rashomon set (or $\epsilon$-level set in their terminology) in terms of classification accuracy. In contrast, the metrics proposed here are defined in terms of the raw outputs in $\Delta_\epsilon$ of a probabilistic classifier. As we shall see, this allows for a more nuanced characterization of multiplicity in probabilistic classification. We briefly overview the metrics by Marx et al. (2020) next.

Ambiguity is the proportion of samples in a dataset that can be assigned conflicting predictions by competing classifiers in the Rashomon set. Discrepancy is the maximum number of predictions that could change in a dataset if we were to switch between models within the Rashomon set. More precisely, given a model $\hat{h}$, the ambiguity $\alpha_\epsilon(\hat{h})$ and the discrepancy $\delta_\epsilon(\hat{h})$ are respectively defined as

$$
\alpha_\epsilon(\hat{h}) = \frac{1}{n} \sum_{i=1}^{n} \max_{h \in \mathcal{R}(\mathcal{H},\epsilon)} \mathbb{1} \left[ \arg \max h(x_i) \neq \arg \max \hat{h}(x_i) \right],
$$

$$
\delta_\epsilon(\hat{h}) = \max_{h \in \mathcal{R}(\mathcal{H},\epsilon)} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1} \left[ \arg \max h(x_i) \neq \arg \max \hat{h}(x_i) \right].
$$

(2)

Both quantities capture predictive multiplicity in terms of thresholded (i.e., 0 or 1) outputs of a classifier. For linear predictive models, the quantities in (2) can be estimated by mixed integer programming tools (Marx et al., 2020).

The above metrics for predictive multiplicity are based on predicted classes. Thus, they require finding the arg max or thresholding the scores at the output of a classifier. In probabilistic classification, thresholding may mask similar predictions produced by competing models and artificially increase multiplicity: output scores can be almost equal across different classes, yet the (thresholded) predicted classes can be very different. For example, two scores $[0.49, 0.51]$ and $[0.51, 0.49]$ for a binary classification problem can lead to entirely different predicted classes—1 and 0, respectively—and ultimately overestimate predictive multiplicity (see Fig. 2 (Left) for another multi-class example). In fact, predictive multiplicity metrics based on predicted classes may yield multiplicity even for a single fixed model when, for example, the threshold criteria for output scores is changed. This subtle, yet important difference motivates us to reconsider existing metrics and introduce a new predictive multiplicity metric for probabilistic classifiers that output scores (cf. Fig. 2 (Right) for an overview).

---

5The definitions in Marx et al. (2020) can be directly extended to multi-class classification.

6That is not to say that multiplicity of predicted classes should not be reported. Ambiguity and discrepancy in (2) are important metrics for stakeholders.
3 Measuring Predictive Multiplicity of Probabilistic Classifiers

We outline below desirable properties of predictive multiplicity metrics for probabilistic classifiers. These properties motivate our definition of Rashomon Capacity and provide guidelines for the creation of new multiplicity metrics in future research. Motivated by the potential individual-level harm incurred by an arbitrary choice of model in the Rashomon set, we focus on of per-sample multiplicity metrics. Next, we formally define Rashomon Capacity in terms of the KL-divergence between the output scores of classifiers in the Rashomon set. We then use Rashomon Capacity to define a predictive multiplicity metric for individual samples in a dataset (similar to ambiguity in (2)). Omitted proofs are in Appendix A, along with more information on Rashomon Capacity in Appendix B.2 and B.3.

3.1 Properties of Multiplicity Metrics for Probabilistic Classifiers

Consider a dataset \( \mathcal{D} \) and a corresponding Rashomon set \( \mathcal{R}(\mathcal{H}, \epsilon) \) for a classification problem with \( c \) classes. We collect all possible output scores for a sample \( x_i \in \mathcal{D} \) and define the \( \epsilon \)-multiplicity set as

\[
\mathcal{M}_\epsilon(x_i) \triangleq \{ h(x_i) \mid h \in \mathcal{R}(\mathcal{H}, \epsilon) \} \subseteq \Delta_c.
\]

Let \( m(\cdot) \) be a measure of predictive multiplicity, and \( m(\mathcal{M}_\epsilon(x_i)) \) be the predictive multiplicity of sample \( x_i \). Which properties should \( m(\cdot) \) have? Ideally, we expect \( m(\mathcal{M}_\epsilon(x_i)) \) to be a bounded value in \([1, c]\), since at least one class is assigned to sample \( x_i \), and at most \( c \) different classes could be assigned to \( x_i \). Moreover, if \( m(\mathcal{M}_\epsilon(x_i)) = 1 \) (i.e., a predictive multiplicity of 1), then one would expect that only one score is produced for \( x_i \) and, thus, all predictions in \( \mathcal{M}_\epsilon(x_i) \) are exactly the same. Similarly, if \( m(\mathcal{M}_\epsilon(x_i)) = c \) (i.e., predictive multiplicity equal to the number of classes), then there must exist \( c \) models \( \{h_1, \ldots, h_c\} \subseteq \mathcal{R}(\mathcal{H}, \epsilon) \) such that \( h_j(x_i) = e_j \). In other words, each of the \( c \) classes can be assigned to the sample, yielding a predictive multiplicity of \( c \). Finally, \( m(\mathcal{M}_\epsilon(x_i)) \) should be monotonic in \( \mathcal{M}_\epsilon(x_i) \), i.e., if \( \mathcal{M}_\epsilon(x_i) \subseteq \mathcal{M}_\epsilon'(x_i) \), then \( m(\mathcal{M}_\epsilon(x_i)) \leq m(\mathcal{M}_\epsilon'(x_i)) \). We summarize these desirable properties of predictive multiplicity metrics in the following definition.
Definition 1. Let $\mathcal{S}_c \triangleq \{\{y\} \mid y \in \Delta_c\}$ be the set of singleton sets in $\Delta_c$, and $\sigma(\mathcal{S}_c)$ its corresponding $\sigma$-algebra. We say that a function $m : \sigma(\mathcal{S}_c) \to \mathbb{R}$ is a predictive multiplicity metric if for any $A, B \in \sigma(\mathcal{S}_c)$

1. $1 \leq m(A) \leq c$;
2. $m(A) = 1$ if and only if $|A| \leq 1$;
3. $m(A) = c$ if and only if $e_k \in A$ for $k \in [c]$, i.e., $A$ contains the corner points of $\Delta_c$;
4. $m(A) \leq m(B)$ if $A \subseteq B$.

We introduce next a predictive multiplicity metric called Rashomon Capacity that satisfies all properties above. In the rest of the paper, when the $\epsilon$-multiplicity set $M_\epsilon(\cdot)$ is clear from context, we use $m(x_i)$ as shorthand for $m(M_\epsilon(x_i))$.

### 3.2 Rashomon Capacity

Our goal is to quantify predictive multiplicity in terms of the score difference assigned to each sample $x_i$ in $\mathcal{D}$. Note that an element in $M_\epsilon(x_i)$ is a probability distribution over $c$ classes. Thus, it is natural to adopt divergence measures for distributions to capture the “variation” of scores in $M_\epsilon(x_i)$. From a geometric viewpoint, a larger spread in scores indicates a greater amount of predictive multiplicity for a given sample $x_i$.

Assume a probability (or “weight”) distribution $P_M$ across models in $\mathcal{R}(\mathcal{H}, \epsilon)$ (and therefore each score in $M_\epsilon(x_i)$), where $M$ denotes the random variable of selecting/sampling the models in the Rashomon set. Intuitively, if $P_M$ assigns mass 1 to a single model and 0 to all other models in the Rashomon set, then the output of only one model is considered. Alternatively, if $P_M$ is the uniform distribution, then the outputs of every model in the set are equally weighed. Given a divergence measure between distributions $d(\cdot \| \cdot)$, we quantify the spread of the scores in $M_\epsilon(x_i)$ by

$$
\rho(M_\epsilon(x_i), P_M) \triangleq \inf_{q \in \Delta_c} \mathbb{E}_{h \sim P_M} d(h(x_i)) \| q).
$$

Here, the minimizing $q$ acts as a “center of gravity” or “centroid” for the outputs of the classifiers in the Rashomon set for a chosen distribution $P_M$ across models. Analogously, the quantity $\rho(M_\epsilon(x_i), P_M)$ can be understood as a measure of “spread” or “inertia” across model outputs. We select the distribution $P_M$ that results in the largest spread in scores:

$$
C_d(M_\epsilon(x_i)) \triangleq \sup_{P_M} \inf_{q \in \Delta_c} \mathbb{E}_{h \sim P_M} d(h(x_i)) \| q).
$$

The missing element is the choice of divergence measure $d(\cdot \| \cdot)$. A natural candidate is cross-entropy (or log-loss) $d(h(x_i)) \| q) = -h(x_i)^\top \log q$, since this is the standard loss used for training and evaluating probabilistic classifiers. Alas, the minimal cross-entropy $\min_q -h(x_i)^\top \log q = h(x_i)^\top \log h(x_i)$ is not 0 and depends on $h(x_i)$. Consequently, if one were to choose $d(\cdot \| \cdot)$ to be cross-entropy, the minimum value of (5) would not be consistent and would depend on $x_i$ — even when the outputs across all models in the Rashomon set match! Thus, we shift cross-entropy by its minimum $-h(x_i)^\top \log h(x_i)$. This results in KL-divergence as our divergence measure of choice:

$$
D_{KL}(h(x_i) \| q) = -h(x_i)^\top \log q + h(x_i)^\top \log h(x_i).
$$

We name the spread in scores measured using KL-divergence as Rashomon Capacity.

---

7In measure-theoretic notation, this corresponds to the set of all countable sets in $\Delta_c$.

8We use the term “metric” loosely here, not in the usual sense of distance between two elements of a set.

9We consider logarithms in base 2, and the unit of Rashomon Capacity is bit.
**Definition 2.** Given a sample $x_i$ and the corresponding $\epsilon$-multiplicity set $\mathcal{M}_\epsilon(x_i)$, the Rashomon Capacity is defined as

$$C(\mathcal{M}_\epsilon(x_i)) \triangleq \sup_{P_M} \inf_{q \in \Delta} \mathbb{E}_{h \sim P_M} D_{KL}(h(x_i) \| q).$$

Moreover, we define $m_C(x_i) \triangleq 2^{C(\mathcal{M}_\epsilon(x_i))}$.

The quantity $C(\mathcal{M}_\epsilon(x_i))$ is ubiquitous in information theory; in fact, $C(\mathcal{M}_\epsilon(x_i))$ is the channel capacity (Cover, 1999) of a channel $P_{Y|M}$ whose rows are the entries of $\mathcal{M}_\epsilon(x_i)$. This connection motivates the name “Rashomon Capacity” and is useful for proving that $m_C(x_i)$ is indeed a predictive multiplicity metric, as stated in the next proposition.

**Proposition 1.** The function $m_C(\cdot) = 2^{C(\mathcal{M}_\epsilon(\cdot))} : \mathcal{X} \to [1, c]$ satisfies all properties of a predictive multiplicity metric in Definition 1.

The Rashomon set is fundamentally different from Rashomon Capacity, in the sense that a larger Rashomon set does not necessarily lead to a larger Rashomon Capacity. Using a binary classification problem as an example, consider two Rashomon sets with scores

$$\mathcal{R}_1(\mathcal{H}, \epsilon) = \{h_1, h_2, h_3\}, \quad h_1(x_i) = [0.45, 0.55], \quad h_2(x_i) = [0.50, 0.50], \quad h_3(x_i) = [0.60, 0.40],$$

$$\mathcal{R}_2(\mathcal{H}, \epsilon) = \{h_1, h_2\}, \quad h_1(x_i) = [0.85, 0.15], \quad h_2(x_i) = [0.10, 0.90].$$

Equation (7)

$\mathcal{R}_2(\mathcal{H}, \epsilon)$ has a larger Rashomon Capacity than $\mathcal{R}_1(\mathcal{H}, \epsilon)$, albeit $|\mathcal{R}_2(\mathcal{H}, \epsilon)| = 2 < |\mathcal{R}_1(\mathcal{H}, \epsilon)| = 3$.

### 3.3 Computing Rashomon Capacity of Discrete Rashomon Sets

The definition of Rashomon Capacity does not assume a finite cardinality of the Rashomon set. However, in order to make Rashomon Capacity computationally tractable, we assume a discrete Rashomon set and, accordingly, a discrete $\epsilon$-multiplicity set. In this case, Rashomon Capacity can be computed by standard procedures such as the Blahut–Arimoto (BA) algorithm (Blahut, 1972; Arimoto, 1972). We discuss the BA algorithm in Appendix B.4 and provide an implementation at this GitHub link.

Remarkably, even when the Rashomon set has infinite cardinality, the value of Rashomon Capacity for a sample can be recovered by considering only a small number of models in the Rashomon set. In fact, for each sample $x_i$, there exists a subset of at most $c$ models that fully captures the variation in scores. This statement is formalized by the next proposition, which can be proven by applying Carathéodory’s theorem (Carathéodory, 1911).

**Proposition 2.** For each sample $x_i \in \mathcal{D}$, there exists a subset $\mathcal{A} \subseteq \mathcal{M}_\epsilon(x_i)$ with $|\mathcal{A}| \leq c$ that fully captures the spread in scores for $x_i$ across the Rashomon set, i.e., $m_C(x_i) = 2^{C(\mathcal{A})}$. In particular, there are at most $c$ models in $\mathcal{R}(\mathcal{H}, \epsilon)$ whose output scores yield the same Rashomon Capacity for $x_i$ as the entire Rashomon set.

The above result point towards a natural strategy for empowering a stakeholder to resolve predictive multiplicity when they provide a sample $x_i$: If $c$ is small, report the (at most) $c$ conflicting scores in $\mathcal{A} \subseteq \mathcal{M}_\epsilon(x_i)$ that yield $m_C(x_i) = 2^{C(\mathcal{A})}$. In terms of Rashomon Capacity, this small subset of scores captures the predictive multiplicity across the entire Rashomon set. The stakeholder can then choose to randomize between scores, accept the average score, or apply another appropriate strategy. The limitation of this approach is that it still requires evaluating outputs for all models in the Rashomon set. In the next section, we discuss a greedy algorithm to address this issue.
4 Empirical Study

We illustrate how to measure, report, and resolve predictive multiplicity of probabilistic classifiers using Rashomon Capacity on UCI Adult (Lichman, 2013), COMPAS (Angwin et al., 2016), HSLS (Ingels et al., 2011), and CIFAR-10 datasets (Krizhevsky et al., 2009). UCI Adult and COMPAS are two binary classification datasets on income and recidivism prediction, respectively, and are widely used in fairness research (Mehrabi et al., 2021). The HSLS is an education dataset, collected from high school students in the USA, whose features include student and parent information. We created a binary label $Y$ from students’ 9\textsuperscript{th}-grade math test scores (i.e., top 50\% vs. bottom 50\%). We select the first three datasets to illustrate the effect of predictive multiplicity on individuals. Finally, we include the CIFAR-10 dataset to demonstrate how to report Rashomon Capacity in multi-class classification.

For the classifiers, we adopt feed-forward neural networks for the first three datasets, and a convolutional neural network VGG16 (Simonyan and Zisserman, 2014) for CIFAR-10. All numbers reported are evaluated on the test set. For more information on the datasets, neural network architectures, and training details, see Appendix C. Code to reproduce our experiments are available at this GitHub link. To the best of the authors’ knowledge, this is the first time that predictive multiplicity metrics are formally measured and reported in neural networks\textsuperscript{10}.

4.1 Measuring and Reporting Predictive Multiplicity via Rashomon Capacity

The first challenge in computing Rashomon Capacity is obtaining the Rashomon set. We describe one simple method for navigating the Rashomon set next and, later in the section, we also consider sampling models in the Rashomon set via random initialization of parameters prior to training (see Section 2 for alternative strategies). Assume the models in $H$ are parameterized. Given a sample $x_i$,\textsuperscript{10}Breiman (2001, Section 8) reported the Rashomon effect in neural networks on a regression problems, yet did not formally measure this effect.
we obtain models with output predictions $p_k$ by approximately solving the following optimization problem which maximizes the output score for all class $k = [c]$:

$$p_k = h_{\tilde{\theta}}(x_i), \text{ where } \tilde{\theta} = \arg \max_{\theta \in \Theta, h_{\theta} \in R(H, \epsilon)} [h_{\theta}(x_i)]_k.$$  \hspace{1cm} (8)

To solve (8), for each $k$, we set the objective to be $\min_{\theta \in \Theta} -[h_{\theta}(x_i)]_k$, compute the gradients, and update the parameter $\theta$ until $h_{\theta} \notin R(H, \epsilon)$, i.e., $L(h_{\theta}) > \epsilon$. Given a pre-trained model in the Rashomon set, (8) can be viewed as an adversarial weight perturbations (AWP) technique to explore the Rashomon set (Wu et al., 2020; Tsai et al., 2021) (see Appendix B.5 for exact weight perturbation on logistic regression). With the discrete set of scores collected by solving (8), the Rashomon Capacity can be computed by the Blahut–Arimoto (BA) algorithm (Blahut, 1972; Arimoto, 1972) (implementation provided in this GitHub link).

We evaluate two methods\footnote{We provide another two methods, label flipping and Fast Sign Gradient Method (Goodfellow et al., 2014) on the network weights, to explore the Rashomon set in Appendix D.1.}, random sampling with different weight initialization seeds and AWP (8), to obtain 100 models from the Rashomon set, and report the Rashomon Capacity in Fig. 3. In particular, we show the mean of the largest 1% and 5% Rashomon Capacity, and the cumulative distribution of the Rashomon Capacity across the samples. As the Rashomon parameter increases, both sampling and AWP lead to higher Rashomon Capacity since the Rashomon set gets larger. The AWP (8) achieves higher Rashomon Capacity than random sampling as AWP intentionally explores the Rashomon set that maximizes the scores variations. It is important to keep in perspective that each sample in the high-Rashomon Capacity tail displayed in Fig. 3. corresponds to an individual who receives conflicting predictions. In applications such as criminal justice and education, conflicting predictions for even one individual should be reported in, e.g., model cards (Mitchell et al., 2019). We report other metrics of multiplicity such as ambiguity/discrepancy and Rashomon ratio in Appendix D.2. We also report Rashomon Capacity of UCI Adult, COMPAS and HSLS trained with classifiers that are not neural networks, e.g., decision tree and random forest, in Appendix D.3.

### 4.2 Resolving Predictive Multiplicity by Greedy Model Selection

We propose a greedy model selection procedure to reduce the number of competing classifiers for resolving predictive multiplicity. Given $R$ competing classifiers, the goal is to select $r$ models ($r < R$) that result in distributions of the Rashomon Capacity similar to that of the original $R$ models. Starting from a dataset $D$ and a Rashomon set $R(H, \epsilon)$, this can be implemented by (i) initializing a set $A$ of models by randomly selecting a model in $R(H, \epsilon)$, (ii) growing $A$ by adding one model...
from $\mathcal{R}(\mathcal{H}, \epsilon)$ that maximizes the average Rashomon Capacity across $\mathcal{D}$, and (iii) stopping until there are $r$ models in $\mathcal{A}$. This greedy model selection is inspired by Property 4 (monotonicity) in Definition 1, since including the models to the set $\mathcal{A}$ does not reduce capacity.

In Fig. 4, we sampled 163 and 52 models from the Rashomon sets for COMPAS and HSLS datasets respectively. Here, the hypothesis space are feed-forward neural networks (see details in Appendix C). Observe that only a small subset of the sampled models, selected by the greedy model selection procedure, is required to recover the distribution of the Rashomon Capacity. On COMPAS dataset, the 10 models obtained by the greedy model selection procedure capture the Rashomon Capacity computed with the original 163 models., i.e., these 10 models display most of the score variations.

5 Final Remarks and Limitations.

We conclude by discussing limitations and future directions. For the computation of the Rashomon Capacity, the adversarial weight perturbation (cf. (8)), despite capturing more score variations than random sampling (cf. Fig. 3), still requires training/perturbing a significant amount of models. This approach is computationally burdensome when scaling up to large datasets with millions of samples. One important limitation of Rashomon Capacity is that, due to the convexity of KL-divergence, in certain cases $m_{C}(\cdot)$ may seem small for already significant score variations across classes. Indeed, this issue will occur for any strictly convex measure of divergence in (4). We provide a further discussion of how to interpret the numerical values of Rashomon Capacity in Appendix B.6.

We highlight two future directions. First, one of the core problems in studying the Rashomon effect is how to efficiently explore the Rashomon set. This is not only the computational bottleneck for estimating Rashomon Capacity, but also for other metrics including the Rashomon ratio and ambiguity/discrepancy. Consequently, designing a more efficient algorithm, e.g., an algorithm similar to the Fast Gradient Sign Method (Goodfellow et al., 2014) in adversarial training, to explore the Rashomon set is an impactful direction. Second, the Rashomon Capacity could be generalized to other probability divergences, e.g., $f$-divergences (Csiszár, 1995), Rényi divergence (Rényi, 1961), or Wasserstein distance (Vaserstein, 1969). This generalization could potentially provide further operational significance and tunability for measuring multiplicity.
References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). {TensorFlow}: A system for {Large-Scale} machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pages 265–283.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias. ProPublica.

Arimoto, S. (1972). An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20.

Bachoc, F., Cesari, T., and Gerchinovitz, S. (2021). The sample complexity of level set approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 424–432. PMLR.

Blahut, R. (1972). Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory*, 18(4):460–473.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical science*, 16(3):199–231.

Cao, J., Hranilovic, S., and Chen, J. (2013a). Capacity-achieving distributions for the discrete-time poisson channel—part i: General properties and numerical techniques. *IEEE Trans. Commun.*, 62(1):194–202.

Cao, J., Hranilovic, S., and Chen, J. (2013b). Capacity-achieving distributions for the discrete-time poisson channel—part ii: Binary inputs. *IEEE Trans. Commun.*, 62(1):203–213.

Carathéodory, C. (1911). Über den variabilitätsbereich der fourier’schen konstanten von positiven harmonischen funktionen. *Rendiconti Del Circolo Matematico di Palermo (1884-1940)*, 32(1):193–217.

Coston, A., Rambachan, A., and Chouldechova, A. (2021). Characterizing fairness over the set of good models under selective labels. In *International Conference on Machine Learning*, pages 2144–2155. PMLR.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Creel, K. and Hellman, D. (2021). The algorithmic leviathan: Arbitrariness, fairness, and opportunity in algorithmic decision making systems. *Virginia Public Law and Legal Theory Research Paper*.

Csiszár, I. (1995). Generalized projections for non-negative functions. In *Proceedings of 1995 IEEE International Symposium on Information Theory*, page 6. IEEE.

D’Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. (2020). Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*.

Dauwels, J. (2005). Numerical computation of the capacity of continuous memoryless channels. In *Proceedings of the 26th Symposium on Information Theory in the BENELUX*, pages 221–228. Citeseer.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226.
Dwork, C. and Ilvento, C. (2018). Individual fairness under composition. *Proceedings of Fairness, Accountability, Transparency in Machine Learning.*

Farsad, N., Chuang, W., Goldsmith, A., Komninakis, C., Méard, M., Rose, C., Vandenberghhe, L., Wesel, E. E., and Wesel, R. D. (2020). Capacities and optimal input distributions for particle-intensity channels. *IEEE Trans. Mol. Biol. Multi-Scale Commun.*, 6(3):220–232.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81.

Goldwasser, S., Kim, M. P., Vaikuntanathan, V., and Zamir, O. (2022). Planting undetectable backdoors in machine learning models. *arXiv preprint arXiv:2204.06974.*

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572.*

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Hancox-Li, L. (2020). Robustness in machine learning explanations: does it matter? In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 640–647.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

INEP (2020). Instituto nacional de estudos e pesquisas educaionais anílio teixeira, microdados do ENEM. https://www.gov.br/inep/pt-br/assuntos/noticias/enem/ministerio-da-educacao-divulga-o-novo-enem.

Ingels, S. J., Pratt, D. J., Herget, D. R., Burns, L. J., Dever, J. A., Ottem, R., Rogers, J. E., Jin, Y., and Leinwand, S. (2011). High school longitudinal study of 2009 (hsls: 09): Base-year data file documentation. nces 2011-328. *National Center for Education Statistics.*

Kemperman, J. (1974). On the shannon capacity of an arbitrary channel. In *Indagationes Mathematicae (Proceedings)*, volume 77, pages 101–115. North-Holland.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images (technical report). *University of Toronto.*

Lichman, M. (2013). UCI machine learning repository.

Marx, C., Calmon, F., and Ustun, B. (2020). Predictive multiplicity in classification. In *International Conference on Machine Learning*, pages 6765–6774. PMLR.

Mason, B., Camilleri, R., Mukherjee, S., Jamieson, K., Nowak, R., and Jain, L. (2021). Nearly optimal algorithms for level set estimation. *arXiv preprint arXiv:2111.01768.*

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.
Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., and Gebru, T. (2019). Model cards for model reporting. In Proceedings of the conference on fairness, accountability, and transparency, pages 220–229.

Nielsen, F. (2020). On a generalization of the jensen–shannon divergence and the jensen–shannon centroid. Entropy, 22(2):221.

Pawelczyk, M., Broelemann, K., and Kasneci, G. (2020). On counterfactual explanations under predictive multiplicity. In Conference on Uncertainty in Artificial Intelligence, pages 809–818. PMLR.

Rényi, A. (1961). On measures of entropy and information. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics, volume 4, pages 547–562. University of California Press.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5):206–215.

Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In Proceedings of the 22nd ACM international conference on Multimedia, pages 1041–1044.

Semenova, L., Rudin, C., and Parr, R. (2019). A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. arXiv preprint arXiv:1908.01755.

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

Smith, H. (2020). Algorithmic bias: should students pay the price? AI & society, 35(4):1077–1078.

Somepalli, G., Fowl, L., Bansal, A., Yeh-Chiang, P., Dar, Y., Baraniuk, R., Goldblum, M., and Goldstein, T. (2022). Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective. arXiv preprint arXiv:2203.08124.

Tsai, Y.-L., Hsu, C.-Y., Yu, C.-M., and Chen, P.-Y. (2021). Formalizing generalization and adversarial robustness of neural networks to weight perturbations. Advances in Neural Information Processing Systems, 34.

Vaserstein, L. N. (1969). Markov processes over denumerable products of spaces, describing large systems of automata. Problemy Peredachi Informatsii, 5(3):64–72.

Wu, D., Xia, S.-T., and Wang, Y. (2020). Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems, 33:2958–2969.

Wu, Y. (2017). Lecture notes on information-theoretic methods for high-dimensional statistics. Lecture Notes for ECE598YW (UIUC), 16.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In International conference on machine learning, pages 7472–7482. PMLR.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in neural information processing systems, 28.
Appendix
This appendix includes omitted proofs for Proposition 1 and 2, additional explanations and discussions, details on experiment setups and training, and additional experiments.

A Omitted proofs
A.1 Proof of Proposition 1

Proposition 3. The function $m_C(\cdot) = 2^{C(M(\cdot))}: \mathcal{X} \to [1, c]$ satisfies all properties of a predictive multiplicity metric in Definition 1.

Proof. For clarity, we assume $|\mathcal{M}_c(x_i)| < \infty$. By the information inequality (Cover, 1999, Theorem 2.6.3) the mutual information $I(M; Y)$ between the random variables $M$ and $Y$ (defined in Section 3) is non-negative, i.e., $I(M; Y) \geq 0$. Moreover, since $I(M; Y) = H(Y) - H(Y|M) \leq \log c$, we have $0 \leq C(M_c(x_i)) \leq \log c$, and therefore $1 < 2^{C(M_c(x_i))} < c$ (since we pick the log base to be 2). If all rows in the transition matrix are identical, $Y$ is independent of $M$, and $H(Y|M) = \sum_{j=1}^{|M_c(x_i)|} P_M(j)H(Y|M = m_j) = \sum_{j=1}^{|M_c(x_i)|} P_M(m_j)H(Y) = H(Y)$; thus, $C(M_c(x_i)) \leq H(Y) - H(Y|M) = 0$ and $m_C(x_i) = 2^{C(M_c(x_i))} = 1$. On the other hand, we denote the $c$ models in $R(\mathcal{H}, \epsilon)$ which output scores are the “vertices” of $\Delta_c$ to be $m_1, \ldots, m_c$, then $H(Y|M = m_j) = 0$, $\forall j \in [c]$. $H(Y|M)$ is minimized to 0 by setting the weights $p_m$ on those $c$ models to be $\frac{1}{c}$ and the rest to be 0. Thus, $C(M_c(x_i)) = H(Y) - H(Y|M) = H(Y)$ and $m_C(x_i) = 2^{C(M_c(x_i))} = c$. Finally, let $\mathcal{M}_c^1(x_i) \subseteq \mathcal{M}_c^2(x_i)$ with random variables $M_1$ and $M_2$ respectively. Without loss of generality, assume that $\mathcal{M}_c^1(x_i) = \{h_1(x_i), \ldots, h_r(x_i)\}$ and $\mathcal{M}_c^2(x_i) = \mathcal{M}_c^1(x_i) \cup \{h_{r+1}(x_i)\}$, and we have

$$
\mathbb{E}_{h \sim P_{M_2}} D(h(x_i)||q) = \sum_{r=1}^{r+1} P_{M_2}(h_i)D_{KL}(h_i(x_i)||q) \\
= \sum_{i=1}^{r} P_{M_2}(h_i)D_{KL}(h_i(x_i)||q) + P_{M_2}(h_{r+1})D_{KL}(h_{r+1}(x_i)||q) \tag{9}
$$

Therefore, $I(M_1; Y) = \inf_{P_{M_1}} \mathbb{E}_{h \sim P_{M_1}} D(h(x_i)||q) \leq \inf_{P_{M_2}} \mathbb{E}_{h \sim P_{M_2}} D(h(x_i)||q) = I(M_2; Y)$, and

$$
I(M_1; Y) \leq I(M_2; Y) \Rightarrow \sup_{M_1} I(M_1; Y) \leq \sup_{M_2} I(M_2; Y) \\
\Rightarrow C(\mathcal{M}_c^1(x_i)) \leq C(\mathcal{M}_c^2(x_i)) \tag{10}
$$

since the power function is monotonic.

We now prove the converse statements. Assume $m(x_i) = c$ and, thus, $C(M_c(x_i)) = \log c$. Let $P_M$ be the capacity-achieving distribution over models. Then $I(M; Y) = \log c$ and, from non-negativity of entropy and the fact that the uniform distribution maximizes entropy, $H(Y) = c$ and $H(Y|M) = 0$. Consequently, again from non-negativity of entropy, $H(Y|M = m_i) = 0$ for all $i \in [\mathcal{M}_c(x_i)]$, and thus $P_{Y|M=m_i}$ is an indicator function (i.e., given $M$, $Y$ is constant w.p.1). Since $H(Y) = c$, the result follows.

Finally, let $C(M_c(x_i)) = 0$ and $P_M$ be the capacity-achieving input distribution. Then $Y$ and $M$ are independent and, thus, $P_{Y|M=m_i} = P_Y$ (i.e., all scores are identical) for all values $i \in [\mathcal{M}_c(x_i)]$. 

15
Since this holds for the capacity-achieving $P_M$, which in turn is the maximum across input distributions, the converse result follows.

**A.2 Proof of Proposition 2**

**Proposition 4.** For each sample $x_i \in D$, there exists a subset $A \subseteq M_i(x_i)$ with $|A| \leq c$ that fully captures the spread in scores for $x_i$ across the Rashomon set, i.e., $m_c(x_i) = 2^{|A|}$. In particular, there are at most $c$ models in $R(H, \epsilon)$ whose output scores yield the same Rashomon Capacity for $x_i$ as the entire Rashomon set.

**Proof.** Carathéodory’s theorem Carathéodory (1911) states that if a point $x$ of $\mathbb{R}^d$ lies in the convex hull of a set $X$, then $x$ can be written as the convex combination of at most $d + 1$ points in $X$. Namely, there is a subset $X'$ of $X$ consisting of $d + 1$ or fewer points such that $x$ lies in the convex hull of $X'$.

In our case, we consider the random variable $M$ of the Rashomon set $R(H, \epsilon)$ and $Y = \Delta_c \triangleq \{ g \in \mathbb{R}^c : \sum_{i=1}^c g_i = 1, \forall k \ g_k \geq 0 \}$ is a $(c - 1)$ dimensional space. We assume $|R(H, \epsilon)| = m$ in the following proof, but $R(H, \epsilon)$ could contain arbitrary large (or infinite) amount of output scores from the models in the Rashomon set. There are also $m$ output scores $\{ h_1(x_i), \cdots, h_m(x_i) \} \in \Delta_c$ in the $c$-multiplicity set $M_i(x_i)$ for each sample $x_i \in D$. By Carathéodory’s theorem, since $\Delta_c$ is $(c - 1)$ dimensional and is convex, any score $h(x_i)$ can be expressed by the convex combination of $(c - 1) + 1 = c$ scores. Moreover, let the $c$ scores be $\{ h_1(x_i), \cdots, h_c(x_i) \}$, since Rashomon Capacity measures the spread of the scores, adding any score $h(x_i) \in \text{convexhull}(h_1(x_i), \cdots, h_c(x_i))$ to the channel constructed by $\{ h_1(x_i), \cdots, h_c(x_i) \}$ would not affect Rashomon Capacity.

**B Additional Details**

**B.1 Predictive multiplicity: fairness, reproducibility, and security**

Predictive multiplicity and the Rashomon effect are related to individual fairness Dwork et al. (2012); Dwork and Ilvento (2018). A mechanism $M : \mathcal{X} \to \mathcal{Y}$ satisfies individual fairness if for every $x, x' \in \mathcal{X}$, $D(M(x), M(x')) \leq d(x, x')$, where $d$ and $D$ are metrics on $\mathcal{X}$ and $\mathcal{Y}$ respectively. It is also called the $(D, d)$-Lipschitz property. Individual fairness aims to ensure that “similar individuals are treated similarly.” The consequence of predictive multiplicity is that the same individual can be treated differently due to arbitrary and unjustified choices made during the training process (e.g., parameter initialization, random seed, dropout probability, etc.).

Predictive multiplicity could also hinder reproducibility of classifiers: conflicting predictions of samples from competing classifiers could lead to entirely different decision regions. Somepalli et al. (2022) studied the reproducibility of decision regions of almost-equally performing learning models, and observe that changes in model architecture (which reflect the inductive bias) lead to visible changes in decision regions. Notably, neural networks in the Rashomon set with very narrow or wide layers tend to have very similar decision regions. On the other hand, neural networks in the Rashomon set with “moderate” number of neurons in each layer are more likely to have distinct decision regions that are fragmented into many small pieces. The connection between predictive multiplicity, neural network architectures, and inductive bias is an interesting research direction. For example, a stronger inductive bias could restrict the arbitrariness of a training process, leading to smaller predictive multiplicity.

---

12In the main paper, we say $\Delta_c$ is a $c$-dimensional probability simplex, but it does not mean $\Delta_c$ is a $c$ dimensional space. In fact, $\Delta_c$ is a $(c - 1)$ dimensional space.
The fact that multiple classifiers may yield distinct predictions to a target a sample while having statistically identical average loss performance can also cause security issues in machine learning. The score variation could result from a malicious learner/designer who either plants an undetectable backdoor or carefully selects a specific model. This may result in intentional manipulation of the output scores for a sample without detectable performance changes Goldwasser et al. (2022).

B.2 Metrics for the spread of scores

The divergence measure between two distributions used in (4) is not restricted to KL-divergence. For example, given a convex function $f : (0, \infty) \rightarrow \mathbb{R}$ satisfying $f(1) = 0$, and assume that $P$ and $Q$ are two probability distributions over a set $\mathcal{X}$, and $P$ is absolutely continuous with respect to $Q$. The $f$-divergence between $P$ and $Q$ is given by Csiszár (1995)

$$D_f(P\|Q) \triangleq \mathbb{E}_Q \left[ f \left( \frac{P(X)}{Q(X)} \right) \right].$$

Different choices of $f$ lead to different divergence; for example, if $f(t) = t \log t$, $D_f(P\|Q) = D_{KL}(P\|Q)$; if $f(t) = (t - 1)^2$, $D_f(P\|Q) = \chi^2(P\|Q)$ is the chi-square divergence; if $f(t) = t \log t - (1 + t) \log(1 + t)/2$, $D_f(P\|Q) = D_{JS}(P\|Q)$ is the Jensen-Shannon divergence. Another example of a tunable probability divergence is the Rényi divergence $R_\alpha(P\|Q)$ of order $\alpha \in \mathbb{R}^+/\{1\}$, defined as Rényi (1961)

$$D_\alpha(P\|Q) \triangleq \frac{1}{\alpha - 1} \log \left( \sum_x \left( \frac{P(x)}{Q(x)} \right)^\alpha Q(x) \right).$$

Its continuous extensions for $\alpha = 1$ and $\infty$ can also be defined. In particular, for $\alpha = 1$, the Rényi divergence recovers KL divergence, and for $\alpha = \infty$, $D_\infty(P\|Q) = \max_x \log P(x)/Q(x)$ is called the max-divergence. Both the $f$-divergence and Rényi divergence generalize the usual notion of KL-divergence used in this paper, and these families of divergences could also be used to measure the spread of the scores in the probability simplex. For example, Nielsen (2020) reported an iterative algorithm to numerically compute a centroid for a set of probability densities measured by the Jensen–Shannon divergence. However, these generalizations of the KL divergence do not necessarily lead to multiplicity metrics that satisfy the properties outlined in Definition 1. More importantly, when taking the supremum over the input distributions (see (5)), we are unaware of a procedure as simple as the Blahut-Arimoto algorithm to estimate the corresponding Rashomon Capacity if the probability divergence is not the KL-divergence. Exploring alternative metrics for measuring score “spread” is a promising future research direction.

B.3 Geometric interpretation of Rashomon Capacity

In Section 3, we introduce the Rashomon capacity to measure the spread of scores from a geometric viewpoint. Here, we further discuss the pleasing geometric interpretations possessed by Rashomon Capacity, which can be found in information theory. Particularly, given a sample $\mathbf{x}_i$, let the information radius $\text{rad}(\mathcal{M}_\epsilon(\mathbf{x}_i))$ and information diameter $\text{diam}(\mathcal{M}_\epsilon(\mathbf{x}_i))$ of the $\epsilon$-multiplicity set $\mathcal{M}_\epsilon(\mathbf{x}_i)$ be Wu (2017)

$$\text{rad}(\mathcal{M}_\epsilon(\mathbf{x}_i)) = \inf_{q \in \Delta} \sup_{p \in \mathcal{M}_\epsilon(\mathbf{x}_i)} D_{KL}(p\|q), \quad \text{diam}(\mathcal{M}_\epsilon(\mathbf{x}_i)) = \sup_{p, p' \in \mathcal{M}_\epsilon(\mathbf{x}_i)} D_{KL}(p\|p'),$$

we have

$$C(\mathcal{M}_\epsilon(\mathbf{x}_i)) \leq \text{rad}(\mathcal{M}_\epsilon(\mathbf{x}_i)) \leq \text{diam}(\mathcal{M}_\epsilon(\mathbf{x}_i)),$$

17
where $C(\mathcal{M}_c(x_i)) = \text{rad}(\mathcal{M}_c(x_i))$ if $\mathcal{M}_c(x_i)$ is a convex set Kemperman (1974).

The proof of (14) is straightforward:

\[
C(\mathcal{M}_c(x_i)) = \sup_{\alpha \in \Delta_m} \inf_{q \in \Delta_c} \sum_{j=1}^{m} \alpha_j D_{KL}(p_j \| q) \\
\leq \inf_{q \in \Delta_c} \sup_{\alpha \in \Delta_m} \sum_{j=1}^{m} \alpha_j D_{KL}(p_j \| q) \\
= \inf_{q \in \Delta_c} \sup_{\alpha \in \Delta_m} \mathbb{E}_{m \sim \alpha} D_{KL}(P_{Y|M=m} \| q) \\
\leq \inf_{q \in \Delta_c} \sup_{\alpha \in \Delta_m} D_{KL}(p \| q) \triangleq \text{rad}(\mathcal{M}_c(x_i)) \\
\leq \sup_{p, p' \in \mathcal{M}_c(x_i)} D_{KL}(p \| p') \triangleq \text{diam}(\mathcal{M}_c(x_i)).
\]

At first glance, the information radius or diameter seem to be more intuitive metrics to measure the “spread” of the all possible scores from the Rashomon set; however, both of them do not satisfy the properties in Definition 1. More importantly, (14) shows that Rashomon capacity is a tighter metric, and is less likely to overestimate the spread of scores, i.e., the predictive multiplicity. Moreover, maximizing the KL divergence is in general an ill-posed problem since the KL divergence is (jointly) convex, and could diverge to infinity. Zhang et al. (2019) demonstrated that maximizing the KL divergence between the scores generated by a classifier with two different samples is solvable if the Euclidean distance between the two samples is upper bounded. In our case, we do not have control over $p, p' \in \mathcal{M}_c(x_i)$ and the underlying models that output $p, p'$ since two models could be very different from each other (in terms of, e.g., the Euclidean distance of the model parameters), but still yield similar test loss due to the existence of multiple local minima.

### B.4 The Blahut-Arimoto algorithm

For the sake of completeness, we describe the Blahut-Arimoto (BA) algorithm Blahut (1972); Arimoto (1972) used in Section 4 for computing channel capacity. For a discrete memoryless channel (DMC) $X \rightarrow Y$ with transition probabilities $P_{Y|X}$ and input probability $Q$, where $\mathcal{X} = \{1, \cdots, m\}$ and $\mathcal{Y} = \{1, \cdots, c\}$. The mutual information $I(X; Y)$ between $X$ and $Y$ is defined as

\[
I(X; Y) \triangleq \sum_{i=1}^{m} \sum_{j=1}^{c} P_{X,Y}(i,j) \log \frac{P_{X,Y}(i,j)}{Q(i)P_Y(j)} = \sum_{i=1}^{m} \sum_{j=1}^{c} P_{Y|X}(j|i)Q(i) \log \frac{P_{X|Y=j}(i)}{Q(i)}. \tag{16}
\]

By definition, the capacity of the channel $P_{Y|X}$ is defined as

\[
C(P_{Y|X}) = \max_Q I(X; Y) = \max_{Q} \sum_{i=1}^{m} \sum_{j=1}^{c} P_{Y|X}(j|i)Q(i) \log \frac{P_{X|Y=j}(i)}{Q(i)}, \tag{17}
\]

where $P_{X|Y}(i|j) = \frac{P_{Y|X}(j|i)Q(i)}{\sum_k P_{Y|X}(j|k)Q(k)}$. Since $P_{X|Y}(i|j)$ can be viewed as a function of the channel $P_{Y|X}$ and $Q$, from (17), it is clear that for a fixed channel $P_{Y|X}$, the channel capacity is a convex function of the input probabilities $Q$. Denote any $P_{X|Y}(i|j) = \Phi(i|j)$, we can alternatively express the mutual information as

\[
I(X; Y) = \sum_{i=1}^{m} \sum_{j=1}^{c} P_{Y|X}(j|i)Q(i) \log \frac{\Phi(i|j)}{Q(i)} = J(Q, \Phi). \tag{18}
\]

It can be proven that Blahut (1972); Arimoto (1972)
1. For a fixed $Q$, $J(Q, \Phi) \leq J(Q, P_{X|Y})$, i.e., $J(Q, P_{X|Y}) = \max_{\Phi} J(Q, \Phi)$, and therefore $C(P_{Y|X}) = \max_{Q} \max_{\Phi} J(Q, \Phi)$.

2. For a fixed $\Phi$, $J(Q, \Phi) \leq \log (\sum_{i=1}^{m} r(i))$, $r(i) = \exp \left( \sum_{j=1}^{c} P_{Y|X}(j|i) \log \Phi(i|j) \right)$, where equality holds if and only if $Q(i) = r(i)/\sum_{k=1}^{m} r(k)$.

The BA algorithm is built upon these two properties, and doubly maximizes $J(Q, \Phi)$. More specifically, let $t$ be the iteration index, and let $Q^{0}$ be a choose initialization of the input distribution, for each iteration, we update $\Phi$ and $Q$ by

1. $\Phi^{l+1}(i|j) = \frac{Q^{l}(i)P_{Y|X}(j|i)}{\sum_{k=1}^{c} Q^{l}(k)P_{Y|X}(j|k)}$, $\forall i, j$.

2. $r^{l+1}(i) = \exp \left( \sum_{j=1}^{c} P_{Y|X}(j|i) \log \Phi^{l+1}(i|j) \right)$.

3. $Q^{l+1}(i) = \frac{\sum_{j=1}^{m} r^{l+1}(i)}{\sum_{k=1}^{c} r^{l+1}(k)}$.

4. $J(Q^{l+1}, \Phi^{l+1}) = \log (\sum_{i=1}^{m} r^{l+1}(i))$.

5. $l = l + 1$.

For the stopping criteria, let $c^{l}(i) = r^{l}(i)/Q^{l}(i)$, we have $J(Q^{l}, \Phi^{l}) = \log (\sum_{i=1}^{m} Q^{l}(i)c^{l}(i))$. Since $J(Q^{l}, \Phi^{l})$ is the logarithm of the average of the $c^{l}(i)$, we have

$$\log \left( \sum_{i=1}^{m} Q^{l}(i)c^{l}(i) \right) \leq C(P_{Y|X}) \leq \max_{i} \log c^{l}(i),$$

(19)

and therefore we update $Q^{l+1}(i)$ and $\Phi^{l+1}(i|j)$ until the stopping criteria is matched,

$$\max_{i} \log c^{l}(i) - \log \left( \sum_{i=1}^{m} Q^{l}(i)c^{l}(i) \right) \leq \epsilon,$$

(20)

where $\epsilon > 0$ is a pre-defined accuracy parameter.

The BA algorithm has also been extended to channels with continuous input and output alphabets, i.e., $|X| = \infty$ and $|Y| = \infty$, based on sequential Monte-Carlo integration methods (i.e., particle filters) Dauwels (2005); Cao et al. (2013a,b); Farsad et al. (2020). Since we deal with finite predicted classes and discrete Rashomon sets, Proposition 2 allows us to circumvent the use of more sophisticated variations of the BA algorithm.

### B.5 Adversarial weight perturbation on unregularized logistic regression

In (8), we introduce an adversarial weight perturbation procedure to estimate Rashomon Capacity in the Rashomon set. In general, the problem in (8) is difficult to analyze, and is usually optimized by using automated gradient computation tools such as Tensorflow Abadi et al. (2016). Here, we provide a special case of unregularized logistic regression, which gradient and Hessian can be analytically computed. We start with a more general case by considering a set of features and labels $\{z_i, y_i\}_{i=1}^{m}$ for a binary classification problem, where $z_i \in \mathbb{R}^{m}$ and $y_i \in \{0, 1\}$. Logistic regression assumes the output scores

$$P(\hat{Y} = 1|Z = z_i; w) = \frac{e^{z_{i}^\top w}}{1 + e^{z_{i}^\top w}}$$

and

$$P(\hat{Y} = 0|Z = z_i; w) = 1 - \frac{1}{1 + e^{z_{i}^\top w}}$$

(21)
We use the Newton–Raphson algorithm to update the weights, i.e., where

\[ \lambda_t \]

\[ t \]

\[ w \]

\[ R \]

\[ C \]

\[ B.6 \] Convexity of the channel capacity

where \( w \in \mathbb{R}^m \) is the vector of weights. The loss in logistic regression (without regularization) is defined as Hastie et al. (2009)

\[
\ell(w) = -\sum_{i=1}^{n} \left( y_i \log P(\hat{Y} = 1|Z = z_i; w) + (1 - y_i) \log(1 - P(\hat{Y} = 1|Z = z_i; w)) \right) \\
= \sum_{i=1}^{n} \left( y_i w^T z_i - \log(1 + e^{w^T z_i}) \right).
\]

Let \( Z = [z_1, \cdots, z_n]^\top \in \mathbb{R}^{n \times m} \) be the feature matrix, \( y = [y_1, \cdots, y_n]^\top \) the label vector, \( p = [P(\hat{Y} = 1|Z = z_1; w), \cdots, P(\hat{Y} = 1|Z = z_n; w)] \) be the score vector, and \( W = \text{diag}(w_1, \cdots, w_m) \) be the weight matrix with diagonal entries equal to \( w \). The gradient and Hessian of \( \ell(w) \) with respect to \( w \) can be expressed as

\[
\nabla \ell(w) = -\sum_{i=1}^{n} z_i(y_i - P(\hat{Y} = 1|Z = z_i; w)) = Z^\top (y - p),
\]

\[
\nabla^2 \ell(w) = -\sum_{i=1}^{n} z_i z_i^\top P(\hat{Y} = 1|Z = z_i; w)(1 - P(\hat{Y} = 1|Z = z_i; w)) = -Z^\top WZ.
\]

We use the Newton–Raphson algorithm to update the weights, i.e.,

\[
w^{t+1} = w^t - (\nabla^2 \ell(w))^{-1} \nabla \ell(w)
\]

\[
= w^t + (Z^\top WZ)^{-1} Z^\top (y - p),
\]

where \( t \in [1, T] \) is the index of the iterations, and \( w^t \) is the weight at iteration \( t \). Note that the features \( z_i \) could be kernel transformation of a sample \( x_i \), logits outputed from a neural network of a sample \( x_i \), or even the sample \( x_i \) itself. When \( z_i = x_i \), it is the vanilla logistic regression.

In order to perform adversarial weight perturbation on \( w \) (i.e., to maximize scores of different classes in (8)), for a target feature input \( z_t \), when \( y_t = 0 \), we aim to maximize \( w^T z_t \) such that \( P(\hat{Y} = 1|Z = z_t; w) \) is maximized. Similarly, when \( y_t = 1 \), we aim to minimize \( w^T z_t \) such that \( P(\hat{Y} = 0|Z = z_t; w) \) is maximized. Therefore, we modify the gradient in (23) to

\[
\nabla \ell(w) = Z^\top (y - p) + \lambda_t z_t,
\]

where \( \lambda_t \) is a regularization parameter, and \( \lambda_t > 0 \) if \( y_t = 0 \), and \( \lambda_t < 0 \) if \( y_t = 1 \). When \( \lambda = 0 \), (25) degenerates to (23). Therefore, the adversarial weight perturbation on logistic regression could be performed by keep updating the weights with

\[
w^{t+1} = w^t + \left( Z^\top WZ \right)^{-1} \left( Z^\top (y - p) + \lambda_t z_t \right),
\]

until convergence. The reason we introduce the features \( z_i \) in the beginning instead of the samples \( z_i \) if that if \( z_i = f(x_i) \) for a neural network \( f(\cdot) \), (26) can be used for last-layer weight perturbation of the neural network Tsai et al. (2021).

**B.6 Convexity of the channel capacity**

In the last paragraph of Section 4, we mention an important limitation of KL-divergence based Rashomon Capacity due to the convexity of KL-divergence: in certain cases \( C(M_\ell(x_i)) \) (and
therefore \( m_C(x_i) = 2^{C(M_e(x_i))} \) may seem small for already significant score variations across the classes. Here, we use an example the binary asymmetric channel Cover (1999) to illustrate this phenomenon. Given \( p, q \in [0, 1] \), a binary asymmetric channel \( X \rightarrow Y \) has a channel transition matrix \( P = \begin{bmatrix} p, & 1-p \\ q, & 1-q \end{bmatrix} \in [0, 1]^{2 \times 2} \). When \( p = q \), the binary asymmetric channel matches the binary symmetric channel. In Fig. 5, we show the channel capacity algorithm, with different pairs \((p, q)\). We observe that the channel capacity is a very “flat” convex function of \( p \) and \( q \); for example, when \( p = 0.5 \) and \( q = 0.1 \), the channel capacity is 1.3, and the channel capacity is larger than 1.8 if the difference \(|p - q|\) is larger than 0.7. When the channel transition matrix to be the estimated scores in a binary classification problem, Rashomon Capacity must be interpreted accordingly. For example, the difference of the scores of a sample for class 0 and class 1 needs to be larger than 0.7 such that the Rashomon Capacity exceeds 1.8. In fact, a Rashomon capacity above 1.1 already corresponds to a potentially significant score variation in practice.

C Datasets and experiments setups.

C.1 Dataset descriptions and pre-processing procedures.

**UCI adult dataset.** The UCI Adult dataset Lichman (2013) contains multiple domestic factors including an individual’s education level, age, gender, occupation, and etc. We drop missing values,
and obtain 46447 samples with 20 features. The 20 features include the one-hot encoded version of the originally selected features [age, education, marital-status, relationship, race, gender, capital-gain, capital-loss, hours-per-week]; note that the features 'race' and 'gender' are binarized. The label is the income, and is divide into two classes: <=50K and >50K.

**COMPAS recidivism dataset.** The COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) dataset Angwin et al. (2016) is a widely used algorithm for judges and parole officers to score criminal defendant’s likelihood of reoffending (i.e., recidivism). The features include [age, charge degree, race, sex, priors crime count, days before screening/arrest, jail in date, jail out date], and the label is the binary prediction on recidivism. We pre-processed the features by binarizing 'race', 'sex', 'charge degree' (felony or others), and 'days before screening/arrest' (<= 30 days or > 30 days); creating a new feature call 'length of stay', which is duration between 'jail in date' and 'jail out date'. The resulting dataset has 52878 samples with 6 features.

**HSLS dataset.** The HSLS (High School Longitudinal Study) dataset Ingels et al. (2011) is collected from 23,000+ participants across 944 high schools in the USA, and it includes thousands of features such as student demographic information, school information, and students' academic performance across several years. We pre-processed the dataset (e.g., dropping rows with a significant number of missing entries and students taking repeated exams, performing k-NN imputation, normalization), and the number of samples reduced to 14,509 and the number of features is 59. For the labels, we created a binary label Y from students’ 9th-grade math test score (i.e., top 50% vs. bottom 50%).

**CIFAR-10 dataset.** The CIFAR-10 dataset Krizhevsky et al. (2009) contains 50,000 colored images for training and 10,000 for test, where each images has $32 \times 32$ pixels, and has a label 10 classes [airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks]. The samples are distributed evenly on the 10 classes for both training and test set.

### C.2 Training details and experimental setups

For UCI Adult, COMPAS and HSLS datasets, the hypothesis space is composed of simple feed-forward neural networks with ReLU activations, and the optimizer is gradient descent trained with the whole datasets, and the training loss is the cross-entropy loss, and the learning rate is 0.001. For UCI Adult dataset, the neural networks have 5 layers/100 neurons per layer, and is trained with 100 epochs. For COMPAS dataset, the neural networks have 5 layers/200 neurons per layer, and is trained with 200 epochs. For HSLS dataset, the neural networks have 5 layers/200 neurons per layer, and is trained with 500 epochs.

For CIFAR-10 dataset, the hypothesis space is composed of VGG16 convolutional neural networks Simonyan and Zisserman (2014), and the optimizer is stochastic gradient descent with batch size 40. The VGG16 models are trained with the cross-entropy loss for 3 epochs and the learning rate is 0.001.

**Sampling.** For UCI Adult, COMPAS and HSLS datasets, we did 5 repeated experiments with difference random seeds for 70%/30% train/test split, and in each experiments, we trained 100 models, and evaluated on the test set. We select the smallest test loss, and select models that have test losses smaller than the smallest test loss plus the Rashomon parameter $\epsilon = \{0.01, 0.02, 0.05, 0.1\}$. For CIFAR-10 dataset, we did 2 repeated experiments with difference random seeds for 90%/10%
train/test split, and in each experiments, we trained 50 models, and evaluated on the test set. The mean accuracy for UCI Adult, COMPAS, HSLS and CIFAR-10 datasets are 0.8034, 0.6540, 0.6247 and 0.8380 respectively. The Rashomon Capacity of all test samples can then be computed by the scores generated by the selected models for difference $\epsilon$, and the mean and standard errors of the largest 1% and 5% Rashomon Capacity, i.e., the statistics on the tails of the Rashomon Capacity, are reported in Fig. 3.

**Adversarial weight perturbation.** For UCI Adult, COMPAS and HSLS datasets, we did 3 repeated experiments with difference random seeds for 95%/5% train/test split, 90%/10% train/test split and 90%/10% train/test split respectively. We first trained a base classifier, and perturbed the weights of the neural networks for each test sample (cf. (8)) with learning rates 0.001 (for UCI Adult and COMPAS datasets) and 0.01 (for HSLS dataset). We require the perturbation procedure to stop updating the weights if either the perturbed scores exceed 0.9, or the test loss is larger than the base test loss plus the Rashomon parameter $\epsilon = [0.01, 0.02, 0.05, 0.1]$. Similarly, for CIFAR-10 dataset, we did 2 repeated experiments with difference random seeds for 99%/1% train/test split. The mean accuracy of the base classifiers for UCI Adult, COMPAS, HSLS and CIFAR-10 datasets are 0.8028, 0.6458, 0.7039 and 0.8167 respectively. Therefore, for each sample, we computed the Rashomon Capacity of all test samples with scores from the base classifier and from the perturbed classifier.

### D Additional experiments

#### D.1 Other methods to explore the Rashomon sets

We introduce other optimization techniques to explore the Rashomon sets, i.e., creating conflicting predictions/scores of samples while maintaining the almost equal test loss.

**Training with label flipping.** We first train a base classifier, and then trained other classifiers on the datasets where only the label of a sample is flipped, with the goal of producing conflicting scores of the sample. In Fig. 6 (Left), we performed the label flipping procedure and report Rashomon Capacity with different Rashomon parameters $\epsilon$ on 1k random samples in the test set of COMPAS and HSLS datasets. The accuracy of the base classifier and the mean accuracy of the classifiers trained with a flipped label are 0.68029 and 0.6660 for COMPAS dataset, and 0.7337 and 0.7324 for HSLS dataset. The Rashomon Capacity are all small since a miss classification of a single sample does not significantly influence the overall test loss and accuracy. Therefore, the classifiers are more likely to ignore the sample with a flipped label.

**Fast gradient sign method (FSGM).** The FSGM (Goodfellow et al., 2014) is different from the proposed adversarial weight perturbation in (8) in two aspects. First, FSGM is applied to create an imperceivable perturbation on the samples instead of the weights. Second, FSGM only uses the $sign$ of the gradient (times a scalar $\beta$) to update the weights. We implemented FSGM on the weights to adversarially change the scores of 1k random samples in the test set of COMPAS and HSLS datasets, and report Rashomon Capacity in Fig. 6 (Right). Note that even with a small scalar $\beta = 0.0001$, the update on the weights often significantly changes the test loss, and most classifiers updated with the FSGM would not belongs to the Rashomon set defined by the Rashomon parameter. Therefore, Rashomon Capacity is almost 0, as observed in Fig. 6 (Right).
D.2 Predictive multiplicity scores based on predicted classes: ambiguity and discrepancy

The computation of the ambiguity and discrepancy in (2) requires searching over the entire Rashomon set, which is computationally infeasible when the hypothesis space is composed of neural networks. However, we can restrict the search in the entire Rashomon set to the sampled Rashomon set, and approximate the ambiguity and discrepancy. In Fig. 7, we report both ambiguity and discrepancy of the 100 sampled models used to produce Fig 3 for UCI Adult, COMPAS, and HSLS datasets. Note that both ambiguity and discrepancy report high predictive multiplicity. For example, in COMPAS dataset, 38% of the samples can be assigned conflicting predictions by switching between classifiers with test loss difference less than 0.05, and in HSLS dataset, the proportion goes to 50%. The reason for such a high predictive multiplicity measured by the ambiguity and discrepancy is that the classifiers, despite having high accuracy, often produce similar scores across the classes for most of the samples.

D.3 Training without neural networks: UCI Adult, COMPAS, and HSLS datasets

We report Rashomon Capacity with learning models that are not neural networks; particularly, we adopt decision tree/ random forest classifiers, and logistic classifiers with no, \( \ell_1 \), \( \ell_2 \) and elastic net regularizations, trained with UCI Adult, COMPAS, and HSLS datasets. We sampled 100 classifiers for each model, and report the distribution of Rashomon Capacity among the samples from Fig. 8 to Fig. 22 (UCI Adult: Fig. 8 - Fig. 12; COMPAS: Fig. 13 - Fig. 17; HSLS: Fig. 18 - Fig. 22).

We observe that for all datasets, Rashomon Capacity is significantly reduced with random forest classifiers, comparing to the decision tree classifiers, i.e., predictive multiplicity is alleviated by ensemble methods with a multitude of decision trees methods. On UCI Adult and HSLS datasets,
we observe that regularization for logistic regression could also reduce Rashomon Capacity. These preliminary numerical results could serve as future directions on the study of reducing predictive multiplicity via ensemble methods and weight regularization.
Figure 8: UCI Adult dataset with decision tree classifiers.

Figure 9: UCI Adult dataset with random forest classifiers.

Figure 10: UCI Adult dataset with logistic regression and no regularization.

Figure 11: UCI Adult dataset with logistic regression and $\ell_1$-regularization.

Figure 12: UCI Adult dataset with logistic regression and $\ell_2$-regularization.
Figure 13: COMPAS recidivism dataset with decision tree classifiers.

Figure 14: COMPAS recidivism dataset with random forest classifiers.

Figure 15: COMPAS recidivism dataset with logistic regression and no regularization.

Figure 16: COMPAS recidivism dataset with logistic regression and $\ell_1$-regularization.

Figure 17: COMPAS recidivism dataset with logistic regression and $\ell_2$-regularization.
Figure 18: HSLS dataset with decision tree classifiers.

Figure 19: HSLS dataset with random forest classifiers.

Figure 20: HSLS dataset with logistic regression and no regularization.

Figure 21: HSLS dataset with logistic regression and $\ell_1$-regularization.

Figure 22: HSLS dataset with logistic regression and $\ell_2$-regularization.