Monoculture Robust Learning

Lee Cohen
Stanford University

Jon Kleinberg
Cornell University

Omer Reingold
Stanford University

Abstract

We introduce a notion of monoculture robustness in Probably Approximately Correct (PAC) learning, as a way to mitigate the problem of *algorithmic monoculture* (Kleinberg and Raghavan, 2021): the tendency of models sharing the same components (e.g., training data) to fail on the same inputs. Specifically, we ask whether a learner with access to a single dataset can output k individually accurate hypotheses whose joint failure probability is comparable to training on k independent datasets. We say that such a learning procedure achieves *monoculture robustness*, and we quantify this by bounding the *conjunctive error* that all k hypotheses misclassify points in an individual sense. We design methods for achieving this type of guarantee by drawing a surprising connection between replicability (Impagliazzo et al., 2022) and monoculture robustness. Specifically, we show that if the base learner is ρ-replicable, then running it independently k times on the same sample yields individual monoculture error that shrinks exponentially with k, controlled by the single-run error of the base learner at the point and ρ. We complement these positive results with a lower bound: for individual monoculture robustness, we prove a worst-case lower bound showing that, for learners, driving the individual conjunctive error down to γ can require a sample-size multiplier of $t = \Omega(\log(1/\gamma))$.

1 Introduction

In many real-world settings, people find themselves in situations where k competing agents (or firms) offer the same type of opportunity or benefit, and each agent uses a learned decision rule to decide whether to extend an offer to a given individual. The individual may receive offers from multiple agents but ultimately accepts at most one. Examples include loan approvals across multiple banks, screening across multiple employers, ticket access for the same concert through different agencies, discounted travel offers during the holidays across competing platforms, or promotional discounts (e.g., a discounted haircut) across several salons in the same promotion network. If these agents rely on the same model, e.g., a single ad system that serves the ads for all the agents, their mistakes can become correlated, and the same individual can be rejected by all k agents, even though the environment appears to provide multiple "chances."

This is the problem of *algorithmic monoculture* (Kleinberg and Raghavan, 2021), in which the use of a common underlying model by multiple competing firms leads to correlated errors and consequently adverse consequences for individuals (and sometimes for the firms themselves). It is a common concern with modern machine learning systems, which are often deployed in settings where multiple providers use a common training pipeline and/or common dataset, and therefore their algorithms may *repeatedly misclassify the same inputs*. This means that even if an individual appears to be getting k "tries" at an opportunity when they apply to k different providers, they might in fact be getting something that more resembles a single try at the opportunity, since the misclassifications of all k providers' models are very similar.

In this paper, we seek strategies to mitigate the negative effects of algorithmic monoculture, asking: if we only have access to one dataset, can we still produce k accurate hypotheses whose *joint*

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

failure probability is significantly lower than a single hypothesis? We formalize this question in the Probably Approximately Correct (PAC) setting by introducing *individual monoculture robustness* and its average analogue, *expected monoculture robustness*. We consider a learner that, given a single sample $S \sim \mathcal{D}^n$, outputs k hypotheses $h_1, \ldots, h_k$, and measure monoculture failure via the *conjunctive error*: the event that *all* k hypotheses misclassify a fixed point x or, in expectation, a random point $x \sim \mathcal{D}_X$. This captures the "shared blind spot" that motivates monoculture concerns: beyond standard accuracy guarantees, it cares about the *correlation* of errors across the returned hypotheses, not just their marginal risks.

This has a direct connection to *algorithmic fairness* (Dwork et al., 2012). Fairness and accuracy guarantees are typically stated for a single predictor, but in the settings above an individual effectively faces k predictors in parallel (or over short time horizons), and what matters is the chance that all of them reject the individual. When errors are correlated across agents, "apply to more places" or "try another place" provides far less protection than one would expect under independence, and the harm can be concentrated on the same people or groups across many decision-makers. Monoculture robustness is designed to capture this concern by shifting attention from marginal error of each model to the joint failure probability across the k models.

A notable conceptual point is that monoculture failure can arise from two sources of randomness. Sometimes the error is *sample-driven*: for a point x, most internal random seeds fail whenever the drawn sample is unrepresentative for x. In this regime, rerunning the same algorithm on the same sample will not help much, and reducing the conjunctive error requires genuinely more data. Other times the error is *internal randomness driven*: for a typical sample S, there are many seeds that predict the correct label, so rerunning with fresh randomness can reduce the probability that *all* k runs land in the "bad" seeds. This separation guides our lower bound and the positive results. Throughout this work, we study both fundamental limits on monoculture robustness and concrete procedures for achieving it, including ERM-based and replicable learners.

Contributions.

Motivated by practical monoculture failures and the desire to "simulate" independence from a single dataset, we outline our contributions as follows.

1. Problem formulation. In section 2, we formalize monoculture robustness for PAC learning by modeling a learner that, given a single dataset, outputs k accurate hypotheses intended to serve k agents. We quantify monoculture failure via conjunctive error and distinguish expected monoculture error from the stronger individual version. We also clarify when individual guarantees are meaningful, namely for a point x with nontrivial single-run error of the base learning algorithm q_x (it is inherently uninformative to demand small joint failure when the single-run error is large).

2. Lower bounds and unavoidable sample increase. We show that, in the realizable setting, individual monoculture robustness cannot in general be improved beyond what additional independent data provides: there exist simple classes and an input x for which driving the individual monoculture error at x down to γ can require a sample-size multiplier $t = \Omega(\log(1/\gamma))$. In particular, for sample-driven failures, this sample-size increase is generally unavoidable.

3. Replicability and same-sample joint error reduction. In section 4 we analyze (pointwise) replicability (Impagliazzo et al., 2022; Hopkins et al., 2025) as a black-box condition under which rerunning a learner independently k times on the same sample provably helps: the individual monoculture error shrinks exponentially with k, controlled by q_x and the replicability parameter. This section also makes two conceptual points: (i) Replicability might appear counterintuitive for achieving monoculture robustness. After all, replicability is a stability property where a learning algorithm produces w.h.p. *the exact same hypothesis* when trained on different samples, provided the internal randomness seed remains fixed. We show that it can still produce diversity across seeds on a fixed sample; and (ii) monoculture robustness is weaker than replicability, so while replicability provides individual monoculture robustness guarantees "for free", it is useful only if we require replicability regardless.

4. Expected Monoculture Error. When relaxing the goal to minimize the expected monoculture error, we consider a simple sample-splitting strategy from Aden-Ali et al. (2024) in Appendix A. We observe that in the realizable proper learning case, a result from Aden-Ali et al. (2024) implies that splitting the sample S of size $2n$ into two independent halves and running an ERM on each

yields a convergence rate of $O(d/n)$, where d is the VC dimension of the hypothesis class. This improves upon the standard expected error lower bound of $\Omega\left(\frac{d}{n}\log\frac{n}{d}\right)$ for a single ERM (Haussler et al., 1994), effectively "gaining" a logarithmic factor. It demonstrates a real separation between duplicating a single hypothesis trained on all the data and producing independent hypotheses by splitting the same data budget. This observation also highlights that if we do not require replicability and merely want low expected monoculture error for a given data budget, subsampling/splitting can be the right tool.

For brevity, omitted proofs are relegated to Appendix C.

1.1 Related Work

The phenomenon of algorithmic monoculture was formalized by Kleinberg and Raghavan (2021), who demonstrated that while shared scoring rules may appear locally optimal for decision-makers, they introduce correlated failures that can reduce overall social welfare compared to noisy, independent evaluators. Bommasani et al. (2022) provided crucial empirical grounding for this theory, introducing the metric of outcome homogenization to quantify the extent to which distinct systems, specifically those relying on shared pre-trained components, systematically exclude the same individuals. This work supports the "component sharing hypothesis", suggesting that the widespread adoption of the same foundation model or models trained on the same data by different agents exacerbates systemic exclusion.

Complementing these findings, Black et al. (2022) and Marx et al. (2020) explore the "Rashomon Effect" or model multiplicity, highlighting that for many tasks, there exists a multiplicity of equally accurate models that nonetheless yield conflicting predictions for individual subjects. While multiplicity implies that diversity is theoretically achievable, standard training pipelines often fail to realize it without explicit intervention. Extending monoculture analysis to strategic settings, Peng and Garg (2024) examined two-sided matching markets, showing that while monoculture can help applicants identify viable matches, it disproportionately increases outcome variance for high-risk individuals. Unlike these works, which primarily characterize the existence and societal impact of correlated errors, our paper focuses on the learning-theoretic limits of overcoming them.

Our work is also related to self-consistency Cooper et al. (2023), which measures agreement between predicted labels across training samples without reference to the true label, while our goal is to control the label-dependent joint error of individually accurate predictors.

We provide a detailed discussion and additional related work on self-consistency, model multiplicity, predictive multiplicity, algorithmic fairness, and replicability in Appendix B.

2 Model

We focus on the binary classification task in the Probably Approximately Correct (PAC) learning setting. Let $\mathcal{X}$ be the input space, and $\mathcal{Y} = \{0,1\}$ the label space. Let $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ represent a hypothesis class with finite Vapnik-Chervonenkis (VC) dimension, d. Consider a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and let $f^* : \mathcal{X} \to \mathcal{Y}$ be the target classifier. In the realizable setting, we assume $f^* \in \mathcal{H}$. An Empirical Risk Minimization (ERM) algorithm takes as input a sample $S = \{(x_i, f^*(x_i))\}_{i=1}^n$ and returns a hypothesis $h \in \mathcal{H}$. While the goal is to minimize the population loss (the true error rate), $\mathcal{L}_D(h) = \mathbb{E}_{\mathcal{D}}[\mathbb{1}(h(x) \neq f^*(x)]$, an ERM algorithm works by minimizing the loss on the provided sample, S. In the realizable case, an ERM algorithm returns a hypothesis $h \in \mathcal{H}$ that is consistent with the sample. The returned hypothesis minimizes the empirical loss on S, where the empirical loss is $\mathcal{L}_S(h) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}[h(x_i) \neq f^*(x_i)]$. To accommodate algorithms that output an ensemble of hypotheses, we extend the definition of (realizable or agnostic) PAC learnability to the setting where an algorithm returns k hypotheses, requiring that standard learning guarantees hold for each.

Definition 2.1 (PAC Learnability)**.** A hypothesis class $\mathcal{H}$ is PAC learnable in the agnostic setting (or realizable, if $\min_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}}(h) = 0$) if there exists a learning algorithm $\mathcal{A}$ and a sample complexity function $n_{\mathcal{H}}(\varepsilon, \delta)$, such that for any $\varepsilon, \delta \in (0,1)$ and any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the following holds. When $\mathcal{A}$ runs on a sample $S \sim \mathcal{D}^n$ of size $n \geq n_{\mathcal{H}}(\varepsilon, \delta)$, it returns hypotheses $h_1, \ldots, h_k$ for some $k \in \mathbb{N}^+$ such that for every $i \in [k]$,

$$\Pr_{S\sim\mathcal{D}^n}\left[\mathcal{L}_{\mathcal{D}}(h_i) \leq \inf_{h\in\mathcal{H}} \mathcal{L}_{\mathcal{D}}(h) + \varepsilon\right] \geq 1 - \delta.$$

In this case $\mathcal{A}$ is an (ε, δ)-learner for $\mathcal{H}$.

We remark that optimal sample complexity bounds are known for any hypothesis class $\mathcal{H}$ with finite VC dimension d. Specifically, $n_{\mathcal{H}}(\varepsilon, \delta) = \Theta\left(\frac{d+\log(1/\delta)}{\varepsilon}\right)$ in the realizable setting, and $n_{\mathcal{H}}(\varepsilon, \delta) = \Theta\left(\frac{d+\log(1/\delta)}{\varepsilon^2}\right)$ in the agnostic setting (see Shalev-Shwartz and Ben-David (2014) and Hanneke (2016a)).

We quantify monoculture failure via the conjunctive error, which measures the simultaneous misclassification by all hypotheses.

Conjunctive error. Suppose algorithm maps samples into a k-tuple of hypotheses $\mathbf{h} = (h_1, \ldots, h_k)^T$. Fix any $x \in \mathcal{X}$. We define the event *k-conjunctive error* of $\mathcal{A}$ on x as

$$e_k(\mathbf{h}, x) = \mathbb{1}\left[\wedge_{i=1}^{k}(h_i(x) \neq f^*(x))\right].$$

Next, we define monoculture robustness and individual monoculture robustness as algorithmic properties of learners.

Definition 2.2 (Expected Monoculture-Robustness)**.** An algorithm $\mathcal{A}$ is a $(\gamma, \varepsilon, \delta)$-*expected monoculture-robust* if $\mathcal{A}$ is an (ε, δ)-learner and

$$\mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}, S\sim\mathcal{D}^n, \mathcal{A}}\left[e_k(\mathbf{h}, x) = 1\right] \leq \gamma.$$

We call $\mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}, S\sim\mathcal{D}^n \mathcal{A}}\left[e_k(\mathbf{h}, x) = 1\right]$ the *expected monoculture error* of $\mathcal{A}$.

Namely, a $(\gamma, \varepsilon, \delta)$-expected monoculture robust algorithm is an (ε, δ)-learner with small expected monoculture error. Appendix Section A interprets this expected-error objective through a sample-splitting result of (Aden-Ali et al., 2024), showing how sample splitting can improve the expected monoculture error in the realizable proper setting.

Definition 2.3 (Individual Monoculture-Robustness)**.** An algorithm $\mathcal{A}$ is $(\gamma_x, \varepsilon, \delta)$-*individual-monoculture-robust* for $x \in \mathcal{X}$ if $\mathcal{A}$ is an (ε, δ)-learner and

$$\Pr_{S\sim\mathcal{D}^n, \mathcal{A}}\left[e_k(\mathbf{h}, x) = 1\right] \leq \gamma_x.$$

We call $\Pr_{S\sim\mathcal{D}^n, \mathcal{A}}\left[e_k(\mathbf{h}, x) = 1\right]$ the *individual monoculture error* of $\mathcal{A}$ on x.

That is, an algorithm is $(\gamma_x, \varepsilon, \delta)$-individual-monoculture-robust for x if it is an (ε, δ)-learner with small individual monoculture error at x.

We note that if for every $x \in \mathcal{X}$ an algorithm satisfies $(\gamma_x, \varepsilon, \delta)$-individual monoculture robustness, it is also $(\mathbb{E}_{x\sim D_{\mathcal{X}}}[\gamma_x], \varepsilon, \delta)$-expected monoculture robust.

For the special case of $k = 1$, we denote the *single-run error* of $\mathcal{A}$ on x as

$$q_x := q_x(\mathcal{A}, n, D) = \Pr_{S\sim\mathcal{D}^n, \mathcal{A}}[e_1] = \Pr_{S\sim\mathcal{D}^n, \mathcal{A}}[f^*(x) \neq (\mathcal{A}(S))(x)].$$

That is, q_x is the probability that $\mathcal{A}$ errs on x when trained on a random sample of size n and returns a single hypothesis.

Sample Partitioning Meta Algorithm. A special type of algorithm is the following: given a random sample S of size tn, take an ERM algorithm A, and run it on an arbitrary or random partition of S into $t \leq k$ independent samples $S_1, \ldots, S_t$. If $k > t$, it returns the k-tuple $\mathbf{h} = (A(S_1), \ldots, A(S_t), A(S_1), \ldots, A(S_1))^T$.

Compared to q_x, the individual monoculture error of a sample partitioning meta algorithm for each $x \in \mathcal{X}$ decays exponentially with t.

Observation 2.1. For every $x \in \mathcal{X}$, a sample partitioning meta algorithm holds

$$\Pr_{S_1,\ldots,S_t\sim\mathcal{D}^n}[e_k(\mathbf{h}, x)] = q_x^t.$$

Moreover, we use this type of algorithm as a baseline for individual monoculture robustness with $n = n_{\mathcal{H}}(\varepsilon, \delta)$ and $t = k$. The rationale is that training each agent on an independent sample ensures that their prediction errors on any given individual are mutually independent.

Goal. Given parameters $\varepsilon, \delta \in (0, 1)$ and $k \in \mathbb{N}^+$, and a sample size $n = n_{\mathcal{H}}(\varepsilon, \delta)$, our objective is to design a meta-algorithm that uses a base (ε, δ)-learner and is:

- Expected Monoculture Robust. Here, our goal is to minimize the expected monoculture error. Specifically, we aim for γ to outperform the standard ERM expected error.

- Individual Monoculture Robust. For any $x \in \mathcal{X}$ with a bounded single-run error (e.g., $q_x < 1/2$), we seek to minimize the individual monoculture error.[1] We aim for algorithms where γ_x decays exponentially in k, i.e., $\gamma_x \leq (g(q_x))^k$ for a polynomial g with high probability.

The focus on points where the single-run error is bounded (e.g., $q_x < 1/2$) can be justified. The following observation confirms that this condition is not a restrictive assumption, but rather a guaranteed property for the vast majority of the domain for any valid PAC learner.

Observation 2.2. Let $\mathcal{A}$ be an (ε, δ)-learner. Then, the single-run error q_x satisfies $q_x < 1/2$ for at least a $1 - 2(\varepsilon + \delta)$ fraction of the distribution $\mathcal{D}$.

3 Sample Complexity Lower Bound

In this section, we lower bound the sample complexity required to guarantee $(\gamma, \varepsilon, \delta)$-individual monoculture robustness. Given that such algorithms must already satisfy standard (ε, δ)-learnability (which requires $n_{\mathcal{H}}(\varepsilon, \delta)$ samples), we quantify the cost of robustness by lower bounding the necessary sample size multiplier, t, and ask:

What is the minimal t required for a sample of size $t \cdot n_{\mathcal{H}}(\varepsilon, \delta)$ to guarantee $(\gamma, \varepsilon, \delta)$-individual monoculture robustness?

We establish a worst-case lower bound on t, demonstrating that a logarithmic dependence on $1/\gamma$ is strictly necessary in this setting.

Theorem 3.1. *For every $d \geq 1$, every $k < \frac{1}{2\varepsilon}$, and every $\delta \leq \frac{\gamma}{4k}$, there exists a hypothesis class $\mathcal{H}$ with $\mathrm{VC}(\mathcal{H}) = d$, a distribution $\mathcal{D}$ with $f^* \in \mathcal{H}$, and a point $x \in \mathcal{X}$ such that for any (ε, δ)-learning algorithm $\mathcal{A}$, the sample size multiplier t required for a single random sample S of size $t \cdot n_{\mathcal{H}}(\varepsilon, \delta)$ to guarantee individual monoculture error $\Pr_{\mathcal{D},\mathcal{A}}[e_k(\mathbf{h}, x) = 1] \leq \gamma$ is lower bounded by $t = \Omega\left(\log \frac{1}{\gamma}\right)$.*

The theorem implies that the sample complexity for individual monoculture robustness scales as $\log(1/\gamma) \cdot n_{\mathcal{H}}(\varepsilon, \delta)$.

We describe the proof using a singleton hypothesis class $\mathcal{H} = \{h_z \mid z \in \mathcal{X}\}$ where $h_z(x) = \mathbb{1}[x = z]$, defined over a finite domain with a uniform marginal distribution. To obtain arbitrary finite VC dimension $d \geq 1$, the full proof adds $d-1$ padding points that can be labeled arbitrarily and have zero probability under the marginal distribution. The target function is $f^* = h_{x^*}$, where x^* is selected uniformly at random from $\mathcal{X}$. We then analyze the probability that the target point x^* is entirely missing from the training sample. The probability of this event decreases exponentially with the sample-size multiplier t. If x^* is missing from the sample, the learner sees only 0 labels, and therefore has no information distinguishing x^* from the other unseen points.

With high probability, each returned hypothesis has population loss at most ε. On this good event, having loss at most ε with respect to the singleton target implies that each returned hypothesis can predict 1 only on a small fraction of the domain; otherwise it would incur too many errors. Hence, the union of the points on which any of the k hypotheses predicts 1 cannot cover too many unseen points. Since the hidden target x^* is uniform among the unseen points, we show that with probability at least $1/2$, none of the k hypotheses predicts 1 on x^*, so all k hypotheses fail simultaneously. Thus, unless $t = \Omega(\log(1/\gamma))$, the individual monoculture error remains larger than γ.

A simple interpretation of the lower-bound construction is a rare qualified subpopulation of mass p. If the shared training sample contains no member of this subpopulation, all k models may reject the same qualified individual because the data provide no evidence distinguishing that subpopulation. This event has probability $(1-p)^{tn}$. For a baseline sample size $n = \Theta(1/p)$, reducing it below γ requires $t = \Omega(\log(1/\gamma))$. Thus, the construction captures a practical failure mode: models can have high overall accuracy while repeatedly failing on individuals from a subpopulation missing from their shared data.

[1] Since we use existing algorithms as black boxes (e.g., ERM), it is only meaningful to ask for small joint failure at points where a single run is already has non-zero probability to be correct; if the single-run error is 1, no procedure can make the "all k fail" event reliably small without additional information.

4 Monoculture Robustness via Replicability

Consider a randomized learning algorithm $\mathcal{A}$ that, given a sample of size n, returns a single hypothesis. A key observation motivating this section is that monoculture errors stem from two distinct sources: the internal randomness of the algorithm and the randomness of the sample.

In general, when the single-run error satisfies $q_x < 1$, there can be two extreme causes for high individual monoculture error:

1. Error driven by sample randomness ($\mathcal{D}$). In this case, the failure depends primarily on the realization of the training set S. If S is unrepresentative or "bad" for a specific input x, the algorithm might misclassify x for most of its internal randomness. Consequently, running the algorithm with fresh internal seeds will provide a small benefit, and the individual monoculture error remains high. This type of error drives the lower bound that appears in Section 3.

2. Error driven by internal randomness. In this case, for a typical training sample S, failure at x is not inherent to S: changing the internal random seed is likely to change the prediction, since many seeds lead to the correct label. Therefore, running the learner k times on the same sample with fresh random seeds can significantly reduce the joint failure probability.

We capture the latter case via *replicability*, which formalizes that for each fixed seed r, the prediction of $\mathcal{A}$ on x is stable across most samples S. This lets us argue that on a good sample, a fresh randomness seed fails only with small probability. An algorithm is called *ρ-replicable* if, by holding its internal randomness r fixed and running it on two independent samples S_1, S_2 drawn from the same distribution $\mathcal{D}$, it returns identical outputs (hypotheses) with probability at least $1 - \rho$.

Definition 4.1 (Replicability, Impagliazzo et al. (2022))**.** For $\rho \in [0, 1/2)$, a randomized algorithm $\mathcal{A}$ is ρ-replicable if for every distribution $\mathcal{D}$,

$$\Pr_{S_1, S_2 \sim \mathcal{D}^n, r} \big[\mathcal{A}(S_1; r) = \mathcal{A}(S_2; r)\big] \;\geq\; 1 - \rho.$$

To formally address base learners that are replicable, we define the following *Same-Sample Meta-Algorithm*: given a single dataset S, we execute the base algorithm $\mathcal{A}$ on S independently k times (with fresh internal randomness) to generate a tuple of hypotheses $(h_1, \ldots, h_k)$.

Replicability alone is insufficient. It merely ensures that the failure probability is determined mostly by the internal seed. If the algorithm is biased towards the wrong label across most seeds, replicability will simply make it *consistently* wrong. Therefore, to guarantee individual monoculture robustness via the meta-algorithm, we rely on the base algorithm $\mathcal{A}$ to be sufficiently accurate on individuals. Specifically, we require the single-run error q_x, which depends on the algorithm $\mathcal{A}$, to be small (e.g., $q_x \leq 1/2$).

While the notion of replicability is valuable on its own merit, it is a stringent requirement that is often impossible to satisfy (e.g., even for simple one-dimensional threshold functions, see Bun et al. (2023)). A weaker notion called *Pointwise Replicability*, recently introduced by Hopkins et al. (2025), relaxes this condition. While replicability demands that the algorithm outputs the same hypothesis h w.h.p., pointwise replicability only requires that the algorithm's predictions be consistent on every fixed individual x.

Definition 4.2 (Pointwise Replicability, Hopkins et al. (2025))**.** For $\rho \in [0, 1/2)$, an algorithm $\mathcal{A}$ is ρ-pointwise replicable if for every distribution $\mathcal{D}$ and every $x \in \mathcal{X}$,

$$\Pr_{S_1, S_2 \sim \mathcal{D}^n, r} \big[\mathcal{A}(S_1; r)(x) = \mathcal{A}(S_2; r)(x)\big] \geq 1 - \rho.$$

We are going to rely on this weaker notion for our results in this section for two reasons. First, replicability strictly implies pointwise replicability (the converse does not hold), so by showing that pointwise replicability implies individual monoculture robustness w.h.p. we also show that replicability implies individual monoculture robustness w.h.p.. Second, Hopkins et al. (2025) show that pointwise replicability is feasible for a much broader class of problems than replicability. Specifically, unlike replicability, *any* hypothesis class with finite VC dimension admits a pointwise replicable learner (see discussion at the end of the section).

We now show that pointwise replicability is sufficient to guarantee monoculture robustness. To facilitate the analysis, let R denote the set of the algorithm's internal random seeds, and let $\mathcal{S}$ denote

the (possibly infinite) multiset of samples representing the distribution $\mathcal{D}^n$. I.e., $\mathcal{S}$ represents the sample space weighted by the probability mass of $\mathcal{D}^n$.

4.1 The Monomatrix Argument

We adopt a matrix-based perspective (Definition 4.3) we call monomatrix to disentangle the two sources of randomness. Our argument proceeds by characterizing "stability" via the rows and "representativeness" via the columns:

1. **Defining Canonical Behavior (Row Majority):** First, we examine the rows (internal seeds) of the monomatrix. Pointwise replicability implies that for any fixed seed r, the algorithm's output on each point in the domain is stable across most samples. This allows us to define a *row majority* value v_r—the "canonical" label that seed r outputs most of the time. In Lemma 4.2, we show that if the algorithm is accurate on x (small q_x), then most seeds usually vote for the correct label (i.e., $v_r = f^*(x)$).
2. **Identifying Representative Data (Good Samples):** Next, we examine the columns (samples). We need to ensure that the specific sample S we draw preserves the canonical behavior of the seeds. We define a *good sample* (formalized as not τ-bad) as one where the algorithm outputs the canonical label v_r for the vast majority of seeds r. Lemma 4.1 establishes that such samples are highly probable, meaning strictly "bad" samples (that trigger widespread deviation from the canonical outputs) are rare.
3. **Synthesis (Monoculture Robustness):** Finally, Theorem 4.6 combines the two previous steps. Conditioned on drawing a good sample (which is likely), the learner's error is driven purely by the small fraction of seeds that have the *wrong* canonical majority. Since these seeds are rare, the probability that k independent runs all hit these bad seeds decays exponentially.

We begin. To systematically analyze the interaction between internal randomness and sample randomness for a fixed input x, we organize the algorithm's correctness on an individual x into a matrix structure defined over these two sources of randomness. See illustration in Table 1.

Definition 4.3 (Monomatrix)**.** Fix $x \in \mathcal{X}$. The *monomatrix* of x, denoted M_x, is the $\{0,1\}$-valued array indexed by $r \in R$ and $S \in \mathcal{S}$ defined by

$$[M_x]_{r,S} \;=\; \mathbb{1}\big[\mathcal{A}(S;r)(x) = f^*(x)\big].$$

We note that while we visualize this as a matrix with columns in $\mathcal{S}$, which can technically be infinite, our analysis treats S as a random variable distributed according to $\mathcal{D}^n$.

Pointwise replicability implies that for any fixed internal seed r, the algorithm's prediction on x is stable across most samples. In the monomatrix view, this corresponds to a row-majority label property: most entries in row r share the same value.

Definition 4.4 (Row-majority and τ-bad samples)**.** Fix a point x. For each internal seed $r \in R$, let $v_r \in \{0,1\}$ be the value that is most likely to be output by the algorithm over the choice of sample $S \sim \mathcal{D}^n$:

$$v_r \;=\; \operatorname*{argmax}_{y \in \{0,1\}} \Pr_{S \sim \mathcal{D}^n}[\mathcal{A}(S;r)(x) = y].$$

For a sample S, define the deviation score $\Delta(S)$ as the fraction of seeds that disagree with their row majority:

$$\Delta(S) \;=\; \frac{1}{|R|} \sum_{r \in R} \mathbb{1}\big[\mathcal{A}(S;r)(x) \neq v_r\big].$$

We say that a sample S is *τ-bad* if $\Delta(S) > \tau$.

The following lemma utilizes Markov's inequality to translate the pointwise replicability parameter into a bound on the fraction of bad columns in the monomatrix.

Lemma 4.1. *Suppose $\mathcal{A}$ is ρ-pointwise replicable. Then for every $\tau \in (0,1)$,*

$$\Pr_{S \sim \mathcal{D}^n}[S \textit{ is } \tau\textit{-bad}] \;\leq\; \frac{\rho}{\tau}.$$

	$\mathcal{S}$			
		$\vdots$		
R	$v_r = \mathbb{1}[\mathcal{A}(s,r)(x) = f^*(x)]$	$\dots$		$\leq \rho$
		$\vdots$		

Table 1: Illustration of a monomatrix for a replicable algorithm, $\mathcal{A}$. The solid entries represent rows where the majority prediction v_r is the same across samples.

While Lemma 4.1 controls how often the algorithm deviates from its majority behavior, we must also quantify the error rate of the majority vote itself.

Definition 4.5 (Row-majority error at x)**.** Define the row-majority error at x by

$$m \;:=\; \frac{1}{|R|}\,\#\big\{r \in R : v_r \neq f^*(x)\big\}.$$

By union bound, the single-run error on any sample S is at most the sum of the majority error rate m and the deviation rate $\Delta(S)$. Consequently, for any "good" sample (where $\Delta(S) \leq \tau$), we obtain the following guarantee:

Corollary 4.1. *If a sample S is not τ-bad, then for any $k \in \mathbb{N}$, running $\mathcal{A}(S;\cdot)$ independently k times yields*

$$\Pr\left[e_k(\boldsymbol{h}, x)|\, S\right] \;\leq\; \big(m + \tau\big)^k.$$

To utilize Corollary 4.1, we must bound m. The following lemma shows that if q_x, the algorithm's baseline misclassification error on x, is sufficiently small, then the fraction of internal seeds yielding the wrong majority must also be small.

Lemma 4.2. *Let $u_r := \Pr_S[\mathcal{A}(S;r)(x) \neq f^*(x)]$ be the fraction of wrong classification values in a row r. If $q_x = \mathbb{E}_r[u_r] < \frac{1}{2}$, then*

$$\Pr_r[v_r \neq f^*(x)] \leq 2q_x.$$

We now synthesize Lemmas 4.1 and 4.2, and Corollary 4.1 into the main theorem of this section.

Theorem 4.6. *Assume $q_x < 1/2$. Let $\mathcal{A}$ be a ρ-pointwise replicable (ε, δ)-learner. Let $\mathcal{M}$ be the same-sample meta-algorithm that, given a single sample $S \sim \mathcal{D}^n$, runs $\mathcal{A}$ independently k times with fresh internal randomness to return an ensemble $h = (h_1, \dots, h_k)$. Fix $\tau \in (0,1)$.*

If a sample S is not τ-bad, then the conditional individual monoculture error is bounded by:

$$\Pr[e_k(h, x) = 1 \mid S] \leq (2q_x + \tau)^k.$$

Consequently, with probability at least $1 - \rho/\tau$ over the choice of $S \sim \mathcal{D}^n$, this bound holds. In particular, for $\tau = \sqrt{\rho}$, with probability at least $1 - \sqrt{\rho}$:

$$\Pr[e_k(h, x) = 1 \mid S] \leq (2q_x + \sqrt{\rho})^k.$$

Proof. Consider a sample S that is not τ-bad. By Corollary 4.1, the failure probability is bounded by $(m + \tau)^k$. Using Lemma 4.2, if $q_x \leq \frac{1}{2}$, then $m \leq 2q_x$, yielding the bound $(2q_x + \tau)^k$. Finally, Lemma 4.1 guarantees that S is not τ-bad with probability at least $1 - \rho/\tau$. □

This means that if we have a (pointwise) replicable algorithm which is sufficiently accurate on x, w.h.p. over the draw of the training set S, the individual monoculture error on x decays exponentially with the ensemble size k.

Remark 4.7. While the above theorem is stated pointwise, Observation 2.2 implies that the condition $q_x < 1/2$ holds for at least a $1 - 2(\varepsilon + \delta)$ mass of points under $\mathcal{D}_X$. Therefore, for each point x in this set, Theorem 4.6 gives a high-probability guarantee over the draw of S.

Next, we discuss the efficiency of achieving individual monoculture robustness by designing a pointwise replicable algorithm. We first state a result from Hopkins et al. (2025).

Theorem 4.8. *Let $\mathcal{H}$ be a hypothesis class with VC dimension d. For any $\rho, \varepsilon, \delta \in (0,1)$ there exists a ρ-pointwise replicable learner that, with probability at least $1-\delta$, outputs a hypothesis with error at most $\inf_{h\in\mathcal{H}} \mathcal{L}_D(h) + \varepsilon$. The sample complexity is $O\left(\frac{d\log(1/\delta)}{\rho^3\varepsilon}\right)$ in the realizable setting, and $O\left(\frac{d\log(1/\delta)}{\rho^4\varepsilon^2}\right)$ in the agnostic setting. In addition, there exists a constant $c \in \mathbb{R}$ such that any ρ-pointwise replicable (ε, c)-learner requires $\Omega\left(\frac{d}{\rho^2\varepsilon\sqrt{\log(1/\rho)}}\right)$ samples in the realizable setting and $\Omega\left(\frac{d}{\rho^2\varepsilon^2\sqrt{\log(1/\rho)}}\right)$ in the agnostic setting .*

By combining the robustness guarantee of Theorem 4.6 with the above, we get a bound over the sample complexity required to ensure individual monoculture robustness for any hypothesis class.

Monoculture robustness is a weaker notion than pointwise replicability. Consequently, (pointwise) replicable algorithms provide individual monoculture robustness "for free". This reveals a counter-intuitive connection: enforcing consistency across samples guarantees diversity across internal seeds for a fixed sample. However, unless replicability is explicitly required,a sample partitioning meta algorithm that runs on a sample of size kn is practically superior. It avoids the heavy sample complexity of replicability (scaling with $1/\rho^2$), making it the efficient choice in the typical regime where $k \ll 1/\rho^2$.

5 Discussion

In this paper, we introduced a learning framework to rigorously quantify algorithmic monoculture in learning theory. By formalizing the notion of conjunctive error within the PAC setting, we disentangled the two primary sources of correlated failure: sampling and internal algorithmic randomness. This reveals a fundamental dichotomy in addressing monoculture.

When errors are driven by the specific realization of the training data (e.g., the absence of a subpopulation in the sample), our lower bound (Theorem 3.1) establishes that learning cannot, in general, manufacture independence from missing information. In this regime, the "shared blind spot" is information-theoretic; to drive the individual monoculture error down in this model, a logarithmic multiplier in sample size is unavoidable. This suggests that for hard inputs, organizational redundancy (running k models) provides false security unless backed by genuinely independent data streams.

Conversely, when errors stem from internal randomness, we uncovered a surprising utility for Replicability. While typically studied as a tool for scientific reproducibility or privacy, we showed that replicability essentially "stabilizes" the learner's behavior on good samples, thereby ensuring that fresh random seeds yield independent trajectories of success. This reframes replicability not just as a constraint for stability, but as a sufficient condition to control conjunctive errors. Our monomatrix perspective gives a useful viewpoint on replicability, not explicitly present in the literature to the best of our knowledge: it separates sample randomness from internal randomness and interprets replicability as a row-stability property, which may also be useful for studying how stability interacts with randomness, diversity, and failure correlation more broadly.

Finally, our interpretation of the Aden-Ali et al. (2024) sample-splitting lemma under the expected-monoculture-error objective in Appendix A offers a practical prescription. In settings where the data budget is fixed, training two weaker hypotheses on independent halves of the data often yields a lower joint failure rate than training a single duplicated hypothesis on the full dataset. This challenges the prevailing dogma that maximizing individual model accuracy is the sole path to system reliability. It suggests that in high-stakes environments where joint failure is catastrophic (e.g., loan approvals), deliberately inducing independence via data partitioning may be superior to maximizing the marginal accuracy of a single monoculture.

Future Work. Future work includes developing "monoculture-aware" training objectives that directly penalize correlated errors without requiring strong replicability constraints, extending data-partitioning guarantees to the agnostic case, and studying settings beyond binary classification, such as multiclass classification, regression, ranking, and generative tasks. Another important direction is to analyze deployment settings where independently trained models share pretrained components (see Appendix D), training data, or other parts of the training pipeline (Bommasani et al., 2022).

Acknowledgments

Lee Cohen and Omer Reingold are supported by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness, and the Simons Foundation investigators award 17351.

References

Aden-Ali, I., Høandgsgaard, M. M., Larsen, K. G., and Zhivotovskiy, N. (2024). Majority-of-three: The simplest optimal learner? In Agrawal, S. and Roth, A., editors, The Thirty Seventh Annual Conference on Learning Theory.

Black, E., Raghavan, M., and Barocas, S. (2022). Model multiplicity: Opportunities, concerns, and solutions. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22.

Blum, A., Stangl, K., and Vakilian, A. (2022). Multi stage screening: Enforcing fairness and maximizing efficiency in a pre-existing pipeline. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22.

Bommasani, R., Creel, K. A., Kumar, A., Jurafsky, D., and Liang, P. S. (2022). Picking on the same person: Does algorithmic monoculture lead to outcome homogenization? In Advances in Neural Information Processing Systems.

Breiman, L. (2001). Statistical modeling: The two cultures. Statistical Science.

Bun, M., Gaboardi, M., Hopkins, M., Impagliazzo, R., Lei, R., Pitassi, T., Sivakumar, S., and Sorrell, J. (2023). Stability is stable: Connections between replicability, privacy, and adaptive generalization. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023.

Cohen, L., Lipton, Z. C., and Mansour, Y. (2020). Efficient Candidate Screening Under Multiple Tests and Implications for Fairness. In 1st Symposium on Foundations of Responsible Computing (FORC 2020).

Cooper, A., Barocas, S., De Sa, C., and Sen, S. (2023). On variance, self-consistency, and arbitrariness in fair classification.

Dai, J., Fazelpour, S., and Lipton, Z. (2021). Fair machine learning under partial compliance. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, AIES '21.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., Mitani, A., Montanari, A., Nado, Z., Natarajan, V., Nielson, C., Osborne, T. F., Raman, R., Ramasamy, K., Sayres, R., Schrouff, J., Seneviratne, M., Sequeira, S., Suresh, H., Veitch, V., Vladymyrov, M., Wang, X., Webster, K., Yadlowsky, S., Yun, T., Zhai, X., and Sculley, D. (2022). Underspecification presents challenges for credibility in modern machine learning. J. Mach. Learn. Res.

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.

Dwork, C. and Ilvento, C. (2019). Fairness Under Composition. In 10th Innovations in Theoretical Computer Science Conference (ITCS 2019).

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. Journal of Machine Learning Research.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. Inf. Comput.

Gomez, J. F., Machado, C., Paes, L. M., and Calmon, F. (2024). Algorithmic arbitrariness in content moderation. In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency.

Gradwohl, R., Shapira, E., and Tennenholtz, M. (2025). Fairness under competition. In The Thirty-ninth Annual Conference on Neural Information Processing Systems.

Hanneke, S. (2016a). The optimal sample complexity of PAC learning. J. Mach. Learn. Res., 17:38:1–38:15.

Hanneke, S. (2016b). Refined error bounds for several learning algorithms. The Journal of Machine Learning Research, 17(1):4667–4721.

Haussler, D., Littlestone, N., and Warmuth, M. (1994). Predicting 0, 1-functions on randomly drawn points. Information and Computation.

Hopkins, M., Impagliazzo, R., and Ye, C. (2025). Approximate replicability in learning. CoRR.

Impagliazzo, R., Lei, R., Pitassi, T., and Sorrell, J. (2022). Reproducibility in learning. In Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing.

Kalavasis, A., Karbasi, A., Moran, S., and Velegkas, G. (2023). Statistical indistinguishability of learning algorithms. In Proceedings of the 40th International Conference on Machine Learning, ICML'23.

Kannan, S., Roth, A., and Ziani, J. (2019). Downstream effects of affirmative action. In Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19.

Kleinberg, J. M. and Raghavan, M. (2021). Algorithmic monoculture and social welfare. Proc. Natl. Acad. Sci. USA, 118(22):e2018340118.

Liu, L. T., Dean, S., Rolf, E., Simchowitz, M., and Hardt, M. (2019). Delayed impact of fair machine learning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19.

Marx, C., Calmon, F., and Ustun, B. (2020). Predictive multiplicity in classification. In Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research. PMLR.

Paes, L. M., Cruz, R., Calmon, F. P., and Diaz, M. (2023). On the inevitability of the rashomon effect. In 2023 IEEE International Symposium on Information Theory (ISIT).

Peng, K. and Garg, N. (2024). Monoculture in matching markets. In Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24.

Schapire, R. E. (1990). The strength of weak learnability. Mach. Learn.

Semenova, L., Rudin, C., and Parr, R. (2022). On the existence of simpler machine learning models. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22.

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

Watson-Daniels, J., Parkes, D. C., and Ustun, B. (2023). Predictive multiplicity in probabilistic classification. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'23/IAAI'23/EAAI'23.

Yona, G. and Rothblum, G. (2018). Probably approximately metric-fair learning. In Proceedings of the 35th International Conference on Machine Learning.

A Minimizing Expected Monoculture Error via Sample Splitting

Consider the realizable, proper learning setting. When minimizing the expected monoculture error, a natural question arises regarding the optimal use of a fixed data budget when training an ERM. Given a sample of size $2n$, is it better to train a single, highly accurate hypothesis on all the data and use it twice, or to partition the data into two independent sets of size $O(n)$ and produce separate hypotheses? In this section, we interpret this question through an existing sample-partitioning result of Aden-Ali et al. (2024). Their Lemma 5 implies that, in the realizable proper learning setting, splitting the sample into two independent halves and running ERM on each yields expected monoculture error $O(d/n)$, effectively gaining a logarithmic factor over the standard expected error bound for a single ERM trained on the full dataset.

For completeness, we start by recalling the expected error bound for ERM by Haussler et al. (1994).

Fact A.1 (ERM expected error)**.** For realizable classes of VC dimension d, ERM satisfies

$$\sup_{\mathcal{D}} \mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_D(\mathcal{A}(S))] = O\Big(\tfrac{d}{n}\log\tfrac{n}{d}\Big).$$

Moreover, there exist $\mathcal{H}, \mathcal{D}$ where the lower bound $\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_D(\mathcal{A}(S))] = \Omega\Big(\tfrac{d}{n}\log\tfrac{n}{d}\Big)$ holds.

Combining the expected-monoculture-error viewpoint with Aden-Ali et al. (2024), one obtains a logarithmic-factor improvement relative to the standard single-ERM expected-error benchmark. To maintain consistency with the definition of q_x (which assumes a training set of size n), we assume a total data budget of $2n$. While this scales the sample size by a constant factor, it does not alter the asymptotic order of the convergence bounds.

Fact A.2. [Aden-Ali et al. (2024), Lemma 5, restated for conjunctive error] Let $\mathcal{A}$ be a sample partitioning meta algorithm that uses a fixed ERM algorithm A on a realizable class $\mathcal{H}$ with VC dimension d. Given a sample S of $2n$ i.i.d. samples, algorithm $\mathcal{A}$ splits S uniformly at random into two disjoint halves S_1, S_2 of size n and returns $h_1 = A(S_1)$, $h_2 = A(S_2)$. Then,

$$\mathbb{E}_{x\sim D}\big[e_2((h_1,h_2)^T, x)\big] = O\Big(\tfrac{d}{n}\Big).$$

We remark that they take the majority of 3 such hypotheses (which is no longer proper), and guarantee an error rate of $O(d/n)$ using this lemma.

This interpretation of Aden-Ali et al. (2024) connects our expected monoculture-error objective to boosting and optimal learning literature (e.g., Schapire (1990); Freund (1995); Hanneke (2016b)). In those works, aggregation methods are used to reduce the marginal error of a final improper predictor. Here, we use the same underlying independence phenomenon differently: rather than aggregating the hypotheses into one predictor, we keep the (ERM proper) hypotheses separate and evaluate their joint failure probability. Thus, two proper ERMs trained on disjoint halves may have joint failure $O(d/n)$, even though duplicating a single ERM would inherit the larger standard expected-error benchmark.

Agnostic learning and label noise. For two learners trained on independent sample halves, their expected conjunctive error is $\mathbb{E}_x[q_x^2]$, whereas standard agnostic PAC guarantees control only the marginal error $\mathbb{E}_x[q_x]$. From Jensen's inequality,

$$\big(\mathbb{E}_x[q_x]\big)^2 \le \mathbb{E}_x[q_x^2] \le \mathbb{E}_x[q_x].$$

Misspecification errors may be shared across learners or may occur on different inputs, so conjunctive error may be close to either endpoint. A sharper agnostic rate therefore requires assumptions controlling the overlap of approximation errors, not only marginal excess risk. Our current framework assumes a fixed target f^*. Explicitly modeling stochastic label noise or irreducible Bayes error would require an extension because the shared label randomness can itself induce correlated failure.

B Additional Related Works

Model multiplicity and predictive multiplicity. Our work is related to the existence and characterization of near-optimal models (Breiman, 2001; Fisher et al., 2019; Paes et al., 2023; Semenova et al.,

2022), and to predictive multiplicity, underspecification, and empirical arbitrariness (Black et al., 2022; D'Amour et al., 2022; Gomez et al., 2024; Watson-Daniels et al., 2023).

Paes et al. (2023) study the number of samples needed to statistically distinguish the performance of models in a Rashomon set, while Semenova et al. (2022) study when a large Rashomon set contains (accurate) simpler models. These results concern properties of a set of near-optimal models. Our question begins after such a set may exist: can a learner with a single dataset produce k accurate hypotheses whose errors are not jointly concentrated on the same individuals?

Thus, the two perspectives are complementary rather than opposites. A large Rashomon set provides potential diversity that a monoculture-robust algorithm might exploit. However, multiplicity alone does not imply monoculture robustness. Many structurally distinct or equally accurate models may still misclassify the same point x, resulting in high conjunctive error. Conversely, low monoculture error does not require models to differ substantially everywhere. The hypotheses may agree on almost the entire domain and differ only in ways that prevent their errors from overlapping.

Moreover, existence results for the Rashomon set do not by themselves provide a procedure for selecting k models with complementary errors. Our framework adds these algorithmic and learning-theoretic requirements.

Algorithmic Fairness in Composition and Sequential Systems. Our work is also closely related to and motivated by the rigorous study of fairness. Dwork et al. (2012) introduced the notion of individual fairness, positing that fairness requires treating similar individuals similarly according to a task-specific metric. Yona and Rothblum (2018) subsequently adapted this to the statistical learning setting, providing sample complexity bounds for learning hypotheses that are "probably approximately" metric-fair on the population. Algorithmic fairness has also been studied in the context of multiple classifiers.

Dwork and Ilvento (2019) formally analyzed fairness under composition, identifying that classifiers which are individually fair in isolation do not necessarily satisfy fairness guarantees when composed into a larger system. This line of inquiry has been extended to sequential settings and other fairness notions; for instance, Blum et al. (2022) showed that standard fairness constraints applied to individual stages of a hiring pipeline can paradoxically reduce the overall diversity of the selected cohort. Similarly, Cohen et al. (2020) further demonstrated that when individual people come from groups with different noise levels, administering the screening policy to all candidates leads to disparate outcomes, necessitating dynamic policies that test noisier groups more often to achieve true equality. Relatedly, Dai et al. (2021) study fair machine learning under partial compliance, showing that when only some employers adopt fairness-conscious policies, strategic applicant behavior can lead to sublinear fairness gains and segregation across institutions.

Liu et al. (2019) and Kannan et al. (2019) demonstrate that single-step fairness constraints can fail to promote long-term equity when decisions are repeated or interactive. More recently, Gradwohl et al. (2025) studied competitive environments, showing that even when fair firms are competing, the resulting ecosystem equilibrium may still be unfair, and that unilaterally improving one classifier's fairness can degrade the ecosystem's overall equity. Unlike these works, which analyze the composition of existing classifiers, our paper focuses on the learning of an ensemble that is explicitly robust to correlated errors.

Replicability in Learning. Replicability was formally introduced into learning theory by Impagliazzo et al. (2022) as an algorithmic property to address the reproducibility crisis in science; it requires that a randomized algorithm produces the exact same output with high probability when run on two independent samples from the same distribution. This notion was subsequently shown to be intimately connected to differential privacy and adaptive data analysis: Bun et al. (2023) proved that replicability is equivalent to approximate differential privacy in the finite sample regime, establishing that "stability is stable" and implying strong generalization bounds. To characterize the limits of this stability, Kalavasis et al. (2023) introduced the notion of Total Variation (TV) indistinguishability, providing information-theoretic equivalences between replicability and other stability concepts. While strict replicability can be restrictive—imposing polynomial sample complexity overheads or impossibility results for basic tasks—recent work by Hopkins et al. (2025) explores relaxations such as approximate and pointwise replicability. These weaker notions retain the spirit of reproducible science while enabling sample-efficient learning for any class with finite VC dimension.

Self-consistency Our problem is related to the notion of self-consistency Cooper et al. (2023). For a fixed input x, self-consistency is defined as $SC(x) = \Pr_{S,S'}[A(S)(x) = A(S')(x)]$, where the two models are produced by the same learning process using different training samples. This notion is label-independent and measures stability or lack of arbitrariness. In contrast, our individual monoculture error is $\Pr_{S,\mathcal{A}}[e_k(h, x) = 1]$. It is label-dependent and measures the joint failure of k individually accurate predictors.

The objectives are aligned when consistency reflects correctness. If all models agree on the correct label, self-consistency is maximal and monoculture error is zero. However, agreement alone is insufficient. Models may consistently predict the wrong label, giving maximal self-consistency and, in our terminology, maximal monoculture failure. Conversely, disagreement can reduce monoculture error when it ensures that at least one model is correct, but diversity for its own sake may merely introduce incorrect or arbitrary predictions. Thus, monoculture robustness does not seek disagreement itself. It seeks diversity specifically in the models' errors while maintaining their individual PAC guarantees.

There is also a direct technical connection. Under Cooper et al. (2023) setup, the varying source is the training sample (simulated via bootstrap replicates). This makes their self-consistency condition closely resemble the pointwise agreement across samples captured by pointwise replicability. The key technical distinction is how each handles internal randomness: pointwise replicability achieves this cross-sample stability by explicitly fixing the algorithm's internal randomness seed, whereas self-consistency does not condition on a shared internal seed. Our main theorem leverages this exact distinction: we use seed-conditioned stability across samples, together with individual accuracy, to prove that running the algorithm multiple times with fresh internal seeds on a fixed shared sample yields an individual monoculture error that decreases exponentially with k .

Therefore, stability across samples (for a fixed seed) can coexist with useful diversity across internal seeds (for a fixed sample). The two goals consequently address complementary deployment concerns. Cooper et al. (2023) seek to avoid arbitrary individual decisions, using ensembling and abstention when agreement is insufficient. We seek to prevent several decision-makers from sharing the same harmful error. Combining high self-consistency across samples with low joint failure across independently randomized predictors is an interesting direction for future work.

C Omitted Proofs

Observation 2.2. Let $\mathcal{A}$ be an (ε, δ)-learner. Then, the single-run error q_x satisfies $q_x < 1/2$ for at least a $1 - 2(\varepsilon + \delta)$ fraction of the distribution $\mathcal{D}$.

Proof. Recall that $q_x = \Pr_{S\sim\mathcal{D}^n}[\mathcal{A}(S)(x) \neq f^*(x)]$ is the probability that the algorithm errs on x over the randomness of the sample. By linearity of expectation, the expected value of q_x over the domain is equivalent to the expected generalization error of the algorithm:

$$\mathbb{E}_{x\sim\mathcal{D}}[q_x] = \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{S\sim\mathcal{D}^n}[\mathbb{I}(\mathcal{A}(S)(x) \neq f^*(x))] = \mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_{\mathcal{D}}(\mathcal{A}(S))].$$

Since $\mathcal{A}$ is an (ε, δ)-learner, we know that $\Pr_S[\mathcal{L}_{\mathcal{D}}(\mathcal{A}(S)) \leq \varepsilon] \geq 1 - \delta$. Bounding the loss by 1 in the failure event, we have:

$$\mathbb{E}_S[\mathcal{L}_{\mathcal{D}}(\mathcal{A}(S))] \leq \varepsilon(1 - \delta) + 1 \cdot \delta \leq \varepsilon + \delta.$$

Applying Markov's inequality to q_x, we obtain:

$$\Pr_{x\sim\mathcal{D}}[q_x \geq 1/2] \leq \frac{\mathbb{E}_x[q_x]}{1/2} \leq 2(\varepsilon + \delta).$$

Thus, $q_x < 1/2$ holds with probability at least $1 - 2(\varepsilon + \delta)$ over the choice of x. □

C.1 Omitted Proofs from Section 3

Theorem 3.1. *For every $d \geq 1$, every $k < \frac{1}{2\varepsilon}$, and every $\delta \leq \frac{\gamma}{4k}$, there exists a hypothesis class $\mathcal{H}$ with $\mathrm{VC}(\mathcal{H}) = d$, a distribution $\mathcal{D}$ with $f^* \in \mathcal{H}$, and a point $x \in \mathcal{X}$ such that for any (ε, δ)-learning algorithm $\mathcal{A}$, the sample size multiplier t required for a single random sample S of size $t \cdot n_{\mathcal{H}}(\varepsilon, \delta)$ to guarantee individual monoculture error $\Pr_{\mathcal{D},\mathcal{A}}[e_k(\mathbf{h}, x) = 1] \leq \gamma$ is lower bounded by $t = \Omega\left(\log \frac{1}{\gamma}\right)$.*

Proof. Fix $d \geq 1$. Consider a domain of the form

$$\mathcal{X} = \mathcal{X}_{\mathrm{hard}} \sqcup \mathcal{X}_{\mathrm{pad}},$$

where

$$\mathcal{X}_{\mathrm{hard}} = \{1, \ldots, M\}$$

and

$$\mathcal{X}_{\mathrm{pad}} = \{p_1, \ldots, p_{d-1}\}.$$

The marginal distribution is uniform on $\mathcal{X}_{\mathrm{hard}}$ and assigns probability 0 to $\mathcal{X}_{\mathrm{pad}}$.

Define the hypothesis class $\mathcal{H}$ as follows. For every $z \in \mathcal{X}_{\mathrm{hard}}$ and every labeling $b \in \{0,1\}^{d-1}$ of the padding points, let $h_{z,b}$ be the hypothesis satisfying

$$h_{z,b}(x) = \mathbb{I}[x = z] \qquad \text{for } x \in \mathcal{X}_{\mathrm{hard}},$$

and

$$h_{z,b}(p_j) = b_j \qquad \text{for } j = 1, \ldots, d-1.$$

Thus, on $\mathcal{X}_{\mathrm{hard}}$, the class behaves like the singleton class, while the padding points can be labeled arbitrarily.

We have $\mathrm{VC}(\mathcal{H}) = d$. Indeed, for any $p' \in \mathcal{X}_{\mathrm{hard}}$ the set $\{p', p_1, \ldots, p_{d-1}\}$ is shattered: the padding points can be labeled arbitrarily, and the point p' can be labeled either by choosing $z = p'$ or by choosing some $z \neq p'$. On the other hand, no set of size $d+1$ can be shattered, since such a set must contain at least two points from $\mathcal{X}_{\mathrm{hard}}$, and the class cannot label two distinct hard-domain points by 1 simultaneously.

Let the target function be $f^* = h_{x^*,0}$, where x^* is selected uniformly at random from $\mathcal{X}_{\mathrm{hard}}$, and 0 denotes the all-zero labeling of the padding points.

Let $n = n_{\mathcal{H}}(\varepsilon, \delta)$ be the sufficient sample size for a standard (ε, δ)-learner for this class $\mathcal{H}$. Let

$$T := \left\lfloor \alpha \log \frac{1}{\gamma} \right\rfloor$$

for a sufficiently small universal constant $\alpha > 0$, and set

$$M = \left\lceil \frac{4Tn + 8k}{1 - 2k\varepsilon} \right\rceil.$$

Since $k < 1/(2\varepsilon)$, the domain size is well-defined.

We prove the lower bound in expectation over the random choice of x^* from $\mathcal{X}_{\mathrm{hard}}$. It then implies that there exists a fixed $x^* \in \mathcal{X}_{\mathrm{hard}}$ for which the same lower bound holds.

Consider a single random sample S of size $N = tn$, where $t \leq T$. Let E be the event that the target point x^* is not present in S. Since the distribution is supported on $\mathcal{X}_{\mathrm{hard}}$, the probability of this event is

$$\Pr[E] = \Pr[x^* \notin S] = \left(1 - \frac{1}{M}\right)^{tn}.$$

Since $M \geq 2$, define

$$c = \left(1 - \frac{1}{M}\right)^{n} \in (0, 1),$$

so $\Pr[E] = c^t$.

By our choice of M, we have $M \geq 4Tn$, and therefore, since $1 \leq t \leq T$,

$$\frac{n}{M} \leq \frac{tn}{M} \leq \frac{1}{4}.$$

By Bernoulli's inequality,

$$c = \left(1 - \frac{1}{M}\right)^{n} \geq 1 - \frac{n}{M} \geq \frac{3}{4}. \tag{1}$$

Conditioned on the event E, the sample S contains only 0-labeled examples. The algorithm $\mathcal{A}$ returns an ensemble of k hypotheses $\{h_1, \ldots, h_k\}$. The conjunctive error event occurs if all hypotheses fail on x^*. Since $f^*(x^*) = 1$, failure means $h_i(x^*) = 0$ for all $i = 1, \ldots, k$.

Let G_i be the event that the i-th returned hypothesis has loss at most ε, and let $G = \cap_i G_i$. Since $\mathcal{A}$ is an (ε, δ)-learner, by a union bound,

$$\Pr[G] \geq 1 - k\delta.$$

Let

$$U = \bigcup_{i=1}^{k} \{x \in \mathcal{X}_{\text{hard}} : h_i(x) = 1\}$$

be the set of hard-domain points predicted as 1 by at least one hypothesis in the ensemble. We now bound the probability that the ensemble covers the target point, namely the event $x^* \in U$.

The following observation is where we use the singleton construction on $\mathcal{X}_{\text{hard}}$. This argument applies also to improper hypotheses. Indeed, for any hypothesis h, if $L_D(h) \leq \varepsilon$ with respect to the singleton target $h_{x^*,0}$, then

$$|\{x \in \mathcal{X}_{\text{hard}} : h(x) = 1\}| \leq \varepsilon M + 1.$$

To see this, if $h(x^*) = 1$, then every point in $\{x \in \mathcal{X}_{\text{hard}} : h(x) = 1\} \setminus \{x^*\}$ is a false positive, so

$$|\{x \in \mathcal{X}_{\text{hard}} : h(x) = 1\}| - 1 \leq \varepsilon M.$$

If $h(x^*) = 0$, then all points in $\{x \in \mathcal{X}_{\text{hard}} : h(x) = 1\}$ are false positives, and the bound is εM. Therefore, on the event G,

$$|U| \leq \sum_{i=1}^{k} |\{x \in \mathcal{X}_{\text{hard}} : h_i(x) = 1\}| \leq k(\varepsilon M + 1).$$

Now condition on the event E, on the realized unlabeled sample points, and on the internal randomness of the learner. Since x^* was chosen uniformly from $\mathcal{X}_{\text{hard}}$ and does not appear in S, its conditional distribution is uniform over the unseen set $\mathcal{X}_{\text{hard}} \setminus S$. Moreover, on the event E, the learner only observes 0 labels, so its output is determined by S and its internal randomness, but not by the identity of x^* except through the fact that $x^* \notin S$.

We need to bound the probability of the event $G \cap \{x^* \in U\}$. For any fixed realization of S and of the learner's internal randomness, consider the set of unseen points x for which both G would hold with target $h_{x,0}$ and $x \in U$. For every such point, the preceding observation implies

$$|U| \leq k(\varepsilon M + 1).$$

Therefore,

$$\Pr[G \cap \{x^* \in U\} \mid E] \leq \frac{k(\varepsilon M + 1)}{M - N}.$$

Since $N = tn \leq Tn$, our choice of M gives

$$M(1 - 2k\varepsilon) > 2Tn + 2k \geq N + 2k.$$

Equivalently,

$$2k(\varepsilon M + 1) < M - N.$$

Thus,

$$\Pr[G \cap \{x^* \in U\} \mid E] \leq \frac{k(\varepsilon M + 1)}{M - N} < \frac{1}{2}.$$

Therefore,

$$\Pr[e_k(\mathbf{h}, x^*)] \geq \Pr[E] - \Pr[G^c] - \Pr[E \cap G \cap \{x^* \in U\}] \geq c^t - k\delta - \frac{1}{2}c^t = \frac{1}{2}c^t - k\delta.$$

Thus, to satisfy the robustness condition $\Pr[e_k(\mathbf{h}, x^*)] \leq \gamma$, we must have

$$\frac{1}{2}c^t - k\delta \leq \gamma,$$

and hence

$$c^t \le 2(\gamma + k\delta).$$

Using the assumption $\delta \le \gamma/(4k)$, we get

$$c^t \le \frac{5}{2}\gamma.$$

Taking logarithms gives

$$t \log \frac{1}{c} \ge \log \frac{2}{5\gamma}.$$

By Equation (1), we have $c \ge 3/4$, and therefore

$$\log \frac{1}{c} \le \log \frac{4}{3}.$$

Consequently,

$$t \ge \frac{\log(2/(5\gamma))}{\log(4/3)} = \Omega\left(\log \frac{1}{\gamma}\right).$$

Thus, for every learner that uses sample-size multiplier $t < \alpha \log(1/\gamma)$, for a sufficiently small universal constant $\alpha > 0$, there exists a target point $x^* \in \mathcal{X}_{\text{hard}} \subseteq \mathcal{X}$ such that the individual monoculture error at x^* is larger than γ. Equivalently, the sample-size multiplier required to guarantee individual monoculture error at most γ is

$$t = \Omega\left(\log \frac{1}{\gamma}\right).$$

□

C.2 Omitted Proofs from Section 4

Lemma 4.1. *Suppose $\mathcal{A}$ is ρ-pointwise replicable. Then for every $\tau \in (0,1)$,*

$$\Pr_{S\sim\mathcal{D}^n}[S \text{ is } \tau\text{-bad}] \;\le\; \frac{\rho}{\tau}.$$

Proof. Fix a row (seed) $r \in R$. Recall that

$$v_r = \arg\max_{y\in\{0,1\}} \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) = y\big].$$

Therefore, for this fixed r,

$$\begin{aligned}
\Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) \neq v_r\big] &= 1 - \max_{y\in\{0,1\}} \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) = y\big] \\
&\le 1 - \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) = y_{x,r}\big] \\
&= \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) \neq y_{x,r}\big],
\end{aligned}$$

where $y_{x,r} = \text{argmax}_{y\in\{0,1\}} \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) = y\big]$ is the pointwise-replicable label guaranteed by Definition 4.2.

Taking expectation over $r \sim R$ w.r.t. the seed distribution and using linearity of expectation,

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^n}[\Delta(S)] &= \mathbb{E}_r \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) \neq v_r\big] \\
&\le \mathbb{E}_r \Pr_{S\sim\mathcal{D}^n}\big[\mathcal{A}(S;r)(x) \neq y_{x,r}\big] \\
&= \Pr_{S\sim\mathcal{D}^n,\, r}\big[\mathcal{A}(S;r)(x) \neq y_{x,r}\big] \\
&\le \rho,
\end{aligned}$$

where the last inequality follows from ρ-pointwise replicability.

By Markov's inequality, for every $\tau > 0$,

$$\Pr_{S\sim\mathcal{D}^n}\big[\Delta(S) > \tau\big] \le \frac{\mathbb{E}_{S\sim\mathcal{D}^n}[\Delta(S)]}{\tau} \le \frac{\rho}{\tau}.$$

□

Lemma 4.2. *Let $u_r := \Pr_S[\mathcal{A}(S;r)(x) \neq f^*(x)]$ be the fraction of wrong classification values in a row r. If $q_x = \mathbb{E}_r[u_r] < \frac{1}{2}$, then*

$$\Pr_r[v_r \neq f^*(x)] \leq 2q_x.$$

Proof. A row r has $v_r \neq f^*(x)$ only if the incorrect bit occurs in more than half of the probability mass of that row, i.e., only if $u_r > \frac{1}{2}$. By Markov's inequality,

$$\Pr_r\big[u_r > \tfrac{1}{2}\big] \;\leq\; \frac{\mathbb{E}_r[u_r]}{1/2} \;=\; 2q_x.$$

□

Corollary 4.1. *If a sample S is not τ-bad, then for any $k \in \mathbb{N}$, running $\mathcal{A}(S;\cdot)$ independently k times yields*

$$\Pr\left[e_k(\boldsymbol{h}, x)|\, S\right] \;\leq\; \big(m + \tau\big)^k.$$

Proof. If S is not τ-bad, then $\Delta(S) \leq \tau$. For a fresh seed r,

$$\{\mathcal{A}(S;r)(x) \neq f^*(x)\} \subseteq \{v_r \neq f^*(x)\} \cup \{\mathcal{A}(S;r)(x) \neq v_r\}.$$

Hence, the single-run error over r is at most $m + \Delta(S) \leq m + \tau$. Since the meta-algorithm uses k independent fresh seeds, the conjunctive error is at most $(m + \tau)^k$. □

D Shared Pretrained Components

Our framework focuses on models sharing training data. We briefly discuss how sharing a pretrained backbone can introduce an additional source of dependence between their errors.

A fixed pretrained backbone B can be absorbed into the learning algorithm by defining $A_B(S;r)$, so our results apply unchanged whenever the required accuracy and pointwise-replicability conditions hold for the fine-tuning procedure with B fixed.

A random backbone shared by all k models introduces an additional shared source of error. This can be formalized by conditioning on B. For example, suppose the fine-tuning stages are independent conditional on B, and let

$$q_{x,B} = \Pr_{S,r}[A_B(S;r)(x) \neq f^*(x) \mid B]\,.$$

The conjunctive error with a shared backbone is then $\mathbb{E}_B[q_{x,B}^k]$, whereas independently trained backbones would give $(\mathbb{E}_B[q_{x,B}])^k$. By Jensen's inequality,

$$\mathbb{E}_B[q_{x,B}^k] \geq \big(\mathbb{E}_B[q_{x,B}]\big)^k.$$

Thus, variability in backbone-specific error can increase the conjunctive error when the backbone is shared. If the fine-tuning sample is also shared, the same argument conditions jointly on B and S. A backbone that creates a shared blind spot cannot generally be repaired by rerunning only the fine-tuning stage. In contrast, if the conditional learner remains accurate and pointwise replicable, our same-sample result can be applied conditional on the backbone.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our abstract and introduction provide a summary of contributions that accurately reflect the new model and the results of this paper.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See discussion throughout the paper.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: We provided all the assumptions and complete proofs (some are in the appendix).

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [N/A]

   Justification: The paper does not include experiments. All results are carefully proven and do not require experiments to establish correctness.

   Guidelines:

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

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

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

Answer: [N/A]

Justification: This is a theoretical paper. All results are carefully proven and do not require experiments to establish correctness.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [N/A]

   Justification: This is a theoretical paper. All results are carefully proven and do not require experiments to establish correctness.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [N/A]

   Justification: This is a theoretical paper. All results are carefully proven and do not require experiments to establish correctness.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [N/A]

   Justification: This is a theoretical paper. All results are carefully proven and do not require experiments to establish correctness.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have carefully read and comply with the NeurIPS code of ethics. This work is theoretical and we do not foresee harmful consequences.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: This work focuses on theoretical Machine Learning. As such, it does not present the direct societal consequences or immediate ethical risks often associated with deployed applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The work is entirely theoretical; we do not release new data or models.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: We do not use existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: We used LLMs for editing.

    Guidelines:

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