How Well Do LLMs Predict Human Behavior? A Measure of their Pretrained Knowledge*

Wayne Gao† Sukjin Han‡ Annie Liang§

March 26, 2026

Abstract

Large language models (LLMs) are increasingly used to predict human behavior. We propose a measure for evaluating how much knowledge a pretrained LLM brings to such a prediction: its *equivalent sample size*, defined as the amount of task-specific data needed to match the predictive accuracy of the LLM. We estimate this measure by comparing the prediction error of a fixed LLM in a given domain to that of flexible machine learning models trained on increasing samples of domain-specific data. We further provide a statistical inference procedure by developing a new asymptotic theory for cross-validated prediction error. Finally, we apply this method to the Panel Study of Income Dynamics. We find that LLMs encode considerable predictive information for some economic variables but much less for others, suggesting that their value as substitutes for domain-specific data differs markedly across settings.

*For helpful discussions, we thank Tom Cunningham, Matt Gentzkow, Guido Imbens, and Pat Kline.
†Department of Economics, University of Pennsylvania
‡School of Economics, University of Bristol
§Department of Economics, Northwestern University

1 Introduction

Large language models (LLMs) are increasingly used in economics as predictive tools—both to generate synthetic responses in place of human subjects (Horton, 2023; Anthis et al., 2025), and to forecast economic outcomes directly (Hewitt et al., 2024; Faria-e Castro and Leibovici, 2024; Chan-Lau et al., 2025). Their appeal in these roles is obvious: A pretrained LLM embeds a vast amount of information and can be deployed at negligible cost, often in settings where collecting new, domain-specific human data would be expensive or infeasible.

What remains unclear is how to assess the *quality* of these predictions. This paper proposes a measure that quantifies the domain-specific value of LLMs in an interpretable unit: the amount of human data they substitute for. Specifically, we ask how much human data would be required for a conventional model trained on that data to match the predictive performance of the pretrained LLM in that domain. We define the LLM's *equivalent sample size* to be the smallest size of training data required for the model to match the LLM.

For some prediction tasks, LLMs may perform comparably to models trained on large datasets, making them a useful surrogate for data collection. For others, LLMs may be so poorly suited that even models trained on a small amount of domain-specific data quickly outperform them. Quantifying the LLM's equivalent sample size can thus inform the need for data collection and the interpretation of LLM-based empirical results across domains.

Beyond introducing this measure, the paper develops a method for estimating the equivalent sample size and for conducting statistical inference. In this part of the paper, we construct a confidence interval for our proposed measure via sequential hypothesis testing, whose validity is shown by developing a novel asymptotic theory for cross-validated estimators. Finally, we illustrate the measure by evaluating LLM predictions of variables in the Panel Study of Income Dynamics, where we find substantial heterogeneity in the extent to which LLMs can proxy for observed data across prediction tasks.

The paper proceeds as follows. Section 2 formalizes the setting and introduces our proposed measure. We consider general prediction problems in which an analyst seeks to predict

an outcome Y given covariates X. A *prediction rule* is a mapping from covariates to outcomes, and the error of any such rule is evaluated by its expected loss under a specified loss function. The standard approach for learning a prediction rule is to collect a dataset of (X, Y) observations and train a model or machine learning algorithm on that data.

Large language models (LLMs) offer an alternative. Rather than training on domain-specific data, the analyst can describe the prediction problem to a pretrained LLM and request predictions of Y given X. While the LLM is itself trained on data, we focus on settings in which the analyst deploys a fixed, pretrained model without further domain-specific fine-tuning, as is common in the literature (see Section 1.1). We refer to the resulting mapping from covariates to predicted outcomes as the *LLM prediction rule*.

We then compare the error of the LLM prediction rule to that of a fixed comparator trained on datasets of increasing size. This comparator may be an off-the-shelf machine-learning method or a structural economic model; for brevity, we refer to it simply as an algorithm. Training the algorithm on datasets of different sizes yields a sequence of prediction rules that trace out an error curve. We define the *equivalent sample size* to be the smallest training sample size at which the algorithm's error weakly improves upon that of the LLM prediction rule. This quantity can be interpreted as the amount of data required to train a model that is as informative as the pretrained LLM for the prediction task at hand.

Section 3 develops an estimator and inference procedure for the equivalent sample size. We estimate the LLM's prediction error using its performance on the full sample, and estimate the error curve of the comparator algorithm via block-out cross-validation. We finally introduce a sequential testing procedure that compares the LLM and algorithm across an ordered sequence of training sizes and yields a one-sided confidence interval for the equivalent sample size.

Section 4 provides theoretical foundations for this inference approach and establishes its validity under the mild assumption that the algorithm's error is weakly reducing in the size of its training data. Because the sequence of hypotheses comparing the algorithm and the LLM is nested, the sequential procedure controls coverage without requiring multiple-testing cor-

rections, thus avoiding the conservativeness that typically arises in multiple hypothesis testing. This feature is especially important in our setting, where we want to compare algorithm performance across a large range of training sample sizes. We finally derive a central limit theorem for the block-out cross-validated risk estimator under a fixed-training-size asymptotic regime, and construct a consistent estimator of its asymptotic variance. Together, these results establish that the proposed procedure delivers a valid one-sided confidence interval for the equivalent sample size with correct asymptotic coverage.

Section 5 applies our proposed measure to evaluate LLM prediction of variables in the Panel Study of Income Dynamics (PSID). The PSID is one of the longest-running longitudinal household surveys in the U.S., and is frequently used in empirical research. Using the 2021 wave of the Panel Study of Income Dynamics, we study a suite of prediction tasks in which respondents' demographic, labor-market, and household covariates are used to forecast hourly wages, homeownership, drinking, and smoking. We construct LLM prediction rules by translating each covariate vector into a natural-language persona description and querying the model. To limit concerns about data leakage, we emphasize post-cutoff evaluation by using static open-source models with documented training cutoffs. The estimated equivalent sample size varies substantially across outcomes—e.g., approximately 20 observations for predicting hourly wages to 600 observations for predicting homeownership—highlighting substantial heterogeneity in the extent to which pretrained knowledge substitutes for domain-specific data.

Finally, Section 6 extends our evaluation framework in two ways. Section 6.1 extends our measure to settings in which the analyst faces multiple prediction tasks. It introduces a cross-domain notion of ESS that asks how much task-specific data a comparator ML method would need to match the LLM across all potential domains, and derives bounds that clarify how this requirement depends on the extent of transfer across tasks. Section 6.2 extends our measure for assessing LLM performance when it is prompted to infer counterfactual outcomes or treatment effects. Under an assumption of unconfoundedness, we define equivalent sample sizes for both the average potential outcome conditional on covariates and the conditional

average treatment effect (CATE). Although the true CATE is unknown, a prediction framework with a transformed outcome enables us to calculate the risk for CATE for both the LLM and ML and thus the equivalent sample size.

1.1 Related Literature

LLMs as Human Surrogates. LLMs are increasingly used as *surrogate* experimental and survey subjects in place of human participants. Researchers prompt LLMs with experimental or survey scenarios and treat the resulting responses as proxies for human behavior. This approach has been applied in survey settings (Argyle et al., 2023; Jansen et al., 2023; Sun et al., 2024) and in experimental settings (Aher et al., 2023; Hewitt et al., 2024; Park et al., 2024; Bohren et al., 2025). Related work explores broader applications of LLMs as simulated economic or social agents, including their use in market simulations and policy analysis (Horton, 2023; Manning et al., 2024).

Despite this growing adoption, there is little consensus on how to assess the quality or value of LLMs as human surrogates. Much of the existing literature evaluates performance in task-specific or descriptive terms—asking whether LLM outputs resemble human responses in a given domain—without a unifying metric that quantifies what is gained or lost by substituting LLMs for human data. At the same time, several papers caution against uncritical reliance on LLMs, documenting systematic biases, domain failures, and sensitivity to prompting choices (Gao et al., 2025; Ludwig et al., 2025). This paper contributes a general measure for evaluating the appropriateness of LLM predictions: the amount of human data the LLM effectively substitutes for.

Model Evaluation. Our paper is closely related to Huang et al. (2025), who also interpret the value of LLM outputs in terms of an equivalent number of human observations. In their setting, the effective sample size arises from an uncertainty-quantification problem: how many synthetic LLM responses are required to construct confidence intervals with valid coverage for human population parameters. In contrast, we treat the LLM as a fixed prediction rule

and define the equivalent sample size as the amount of real, domain-specific data required for a supervised learning algorithm to match the LLM's predictive performance.

Additionally, our approach is in the spirit of Erev et al. (2007)'s notion of the *equivalent number of observations (ENO)*, which evaluates game-theoretic models by asking how many subject observations would be required for the sample average of observed behavior to predict as well as the theory. The objects of comparison are however different: In their setting, the object being evaluated—a behavioral or equilibrium model—is itself estimated from data, whereas we treat the LLM as a fixed prediction rule that is not trained or tuned on the domain sample. Additionally, their comparator is the sample average of observed behavior, while ours is the prediction of a supervised learning algorithm. Thus our measure applies to a substantially broader set of prediction tasks.

Finally, our work is related more broadly to work on evaluating economic models (Fudenberg et al., 2022, 2026; Andrews et al., 2025), where here the "model" is the pretrained LLM. In particular, our benchmarking of LLM performance against that of a comparator algorithm is similar to Fudenberg et al. (2022), although our focus on the equivalent *sample size* of the comparator algorithm is entirely new.

Asymptotic Theory with Cross-Validation. A growing literature develops asymptotic theory for cross-validated prediction error to address the dependence induced by overlapping training samples across folds (Austern and Zhou, 2020; Bates et al., 2024; Fava, 2025; Lei, 2025). Our results in Section 3.4 for block-out cross validation contribute to this literature, and are in particular related to Lei (2025) and Fava (2025). The key difference in our analysis is that we consider an asymptotic regime where the training sample size is held fixed. Lei (2025) and Fava (2025) instead work with a standard asymptotic regime where the training sample size grows large, which is natural for typical machine learning applications. Inference for our proposed measure demands a different asymptotic regime, since we implement hypothesis testing with respect to a given number of training observations.

2 Framework

Section 2.1 introduces our framework: we study prediction problems in which an analyst observes a vector of covariates X and seeks to predict an outcome Y. Section 2.2 defines a prediction rule obtained by providing the LLM with a text description of X and eliciting a prediction of Y. This approach contrasts with the standard machine learning paradigm, reviewed in Section 2.3, in which the analyst first collects a dataset of (x, y) observations and then trains a task-specific prediction algorithm on that data. Finally, Section 2.4 introduces our proposed measure: the smallest training sample size for which a domain-specific model or algorithm achieves predictive performance comparable to that of the LLM.

2.1 Setup

A *prediction problem* is a pair (X, Y) where $X \in \mathcal{X}$ is a covariate vector and $Y \in \mathcal{Y}$ is an outcome of interest.

Example 1. *The covariate vector X describes a worker's educational attainment, years of experience, occupation, industry, geographic location, and past earnings. The outcome $Y \in \mathbb{R}_+$ is the worker's wage in the following year.*

Example 2. *The covariate vector X describes an individual's age, educational attainment, marital status, and race. The outcome $Y \in [0, 1]$ is the probability with which the individual owns a house.*

A *prediction rule* is any map $f : \mathcal{X} \to \mathcal{Y}$ that predicts the outcome Y given X, where the LLM-based prediction rule is one special choice of f (see Section 2.2). As is standard, we measure the accuracy of a prediction $\hat{y}$ given true outcome y using a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, where $\ell(y, y')$ is the inaccuracy of predicting y' when the true outcome is y (e.g., $\ell(y, y') = (y' - y)^2$ for regression or $\ell(y, y') = 1\{y' \neq y\}$ for classification). Define the *error* of prediction rule f to be

$$e(f) \equiv \mathbb{E}_P[\ell(f(X), Y)]$$

i.e., the expected loss when predictions are made using f, where P denotes the true (unknown) joint distribution of (X, Y). This is also known as the *risk* of f.

2.2 LLM-Based Prediction

Our procedure and its guarcrosses generalize to any fixed prediction rule f, but we are in particular interested in the prediction rule f_{LLM} that corresponds to asking a pretrained LLM to predict the outcome. The covariate vector X and desired outcome Y are described in natural language and given to the LLM, and the LLM's response is taken as the predicted outcome.

Here is an example prompt:

Prompt 1. *Here is your persona: You are a 37-year-old male living in Tennessee in 2020. You work in architecture and engineering occupations in manufacturing industry. You have 4 years of work experience in the current job. You have completed 14 years of education. You are married and have 1 child. Your father has 12 years of education. Your mother has 12 years of education. Your race is white. Your health is excellent. You are a Protestant.*

Given your persona, do you own a house or apartment or do you rent? If you own, say 1; if you rent, say 5; if you neither own nor rent, say 8.

The prompt text is fixed up to placeholders for individual covariate values, so the structure of the prompt is constant and varies only in the realized covariate values. Let

$$e_{\text{LLM}} \equiv e(f_{\text{LLM}})$$

denote the error of this prediction rule.

Throughout we treat the LLM-based rule f_{LLM} as fixed, and do not model the background data on which it is trained. In practice, researchers often do not retrain foundation models, but instead deploy a given pretrained system. Our object of interest is therefore the predictive performance of this fixed system.

2.3 Domain-Specific Learning

We compare the above approach to prediction rules that are obtained by training a model or algorithm on domain-specific data. This approach first specifies a set of prediction rules $\mathcal{F}$, and then uses data to select among them. Formally, an *algorithm* a is a map

$$a : \mathcal{D} \to \mathcal{F}$$

from the set of all finite datasets $\mathcal{D} \equiv \cup_{N \geq 1} (\mathcal{X} \times \mathcal{Y})^N$ to the set of prediction rules $\mathcal{F}$.[1]

A typical choice of a would be an economic model or a machine-learning procedure. In the first case, $\mathcal{F}$ represents an interpretable class of economically motivated prediction rules, and $a(D)$ denotes the rule obtained by estimating the model's parameters on the dataset D using a specified estimation method. In the second case, $\mathcal{F}$ is a potentially rich class of predictors—such as ensembles of decision trees—and $a(D)$ is the fitted predictor produced by training the algorithm on D, including any regularization or tuning steps. In both cases, the algorithm maps data into a concrete prediction rule whose out-of-sample performance typically improves with the size of the training dataset, providing a natural benchmark against which to evaluate the predictive value of a fixed LLM.

For any sample size $N \in \mathbb{N}_+$, let

$$f_N \equiv a(D_N)$$

be the prediction rule obtained by training the algorithm on a random sample $D_N \equiv \{(x_i, y_i)\}_{i=1}^N$ of N observations (x_i, y_i) drawn i.i.d. from P. Since D_N is random, the prediction rule f_N is as well. We define

$$e_N^a \equiv \mathbb{E}[e(f_N)] = \mathbb{E}_{D_N}\left[\mathbb{E}_{(x,y)}[\ell(a(D_N), (x, y))]\right]$$

[1] All of our procedures and results extend if we instead model the algorithm as a map $a : \mathcal{D} \times [0, 1] \to \mathcal{F}$, where the additional argument in $[0, 1]$ represents an exogenous source of randomness, allowing the algorithm's output to be stochastic—for example, due to random initialization, subsampling, or internal randomization.

to be the expected prediction error of the algorithm trained on N datapoints. That is, e_N^a integrates both over the randomness of the sample D_N used to train f_N and the randomness of the new observation $(X, Y) \sim P$ on which f_N is evaluated.

2.4 Proposed Measure

Our proposed measure of the LLM's *equivalent sample size* is the minimum amount of domain-specific data an experimenter would need to collect so that the algorithm a, when trained on that data, produces predictions at least as accurate as the LLM's.

Definition 2.1. *Fix a prediction problem* (X, Y)*. The* a-equivalent sample size *(a-ESS) of* f_{LLM} *is*

$$N^* \equiv N_a^* = \min \left\{ N \in \mathbb{N}_+ : e_N^a \leq e_{\text{LLM}} \right\}.$$

i.e., the smallest training sample size N^* *such that the predictive performance of* f_{N^*} *weakly improves upon that of the LLM. If no such* N *exists, we set* $N^* = \infty$*.*

The equivalent sample size depends on the benchmark algorithm a, as well as on the choice of LLM, its prompting protocol, and the prediction problem. Thus it should be interpreted as measuring the LLM's predictive ability in a specific prediction problem and relative to the specified algorithm, rather than as an absolute measure of the LLM's predictive content. We encourage the use of algorithms a that are well-suited to the prediction problem and thus serve as more demanding benchmarks.

At one extreme, $N^* = 1$ means that the LLM provides no predictive advantage relative to the algorithm a in this prediction problem, even when the algorithm is trained on a single data point. This case corresponds (for example) to an LLM that is very poorly suited to the prediction task, or which relies on spurious correlations in its original training data.

At the other extreme, the equivalent sample size can also be infinite, meaning that no amount of domain-specific data allows the benchmark algorithm to match the LLM's predictive performance. This could occur if a is too restrictive to approximate the LLM's mapping

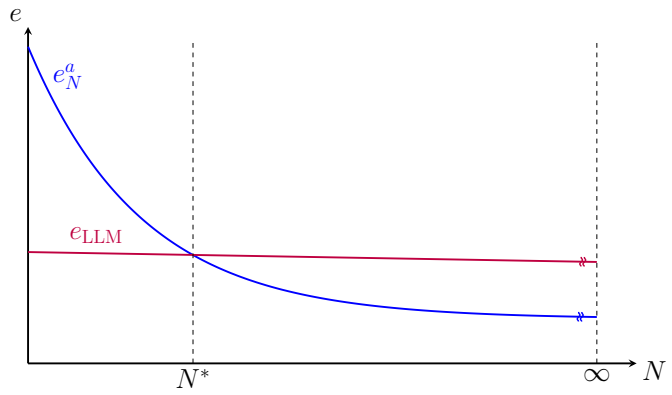


Figure 1: a-Equivalent Sample Size (N^*)

even asymptotically—for example, if a is an economic model that imposes strong structural restrictions.

In general, we expect the LLM's equivalent sample size to be finite. That is, while the LLM's pretraining enables it to make useful predictions even in the absence of any new data, a sufficiently flexible algorithm trained on data from the relevant prediction task eventually outperforms it.

In our application we will report figures such as Figure 1, which plots the constant error e_{LLM} against e_N^a as N varies. The intersection point defines the a-equivalent sample size N^*.

3 Estimation and Inference: Procedure

This section describes our procedure for estimating and conducting inference for the a-equivalent sample size N^*. Recall that N^* is defined as the smallest training sample size N given which the machine learning algorithm weakly outperforms the LLM-based prediction rule:

$$N^* \equiv \min\{N : e_N^a \leq e_{\text{LLM}}\}.$$

Section 3.1 describes our approach to estimating the error of the LLM-based prediction rule, e_{LLM}. Section 3.2 describes a block-out cross-validation procedure for estimating the error curve of the machine learning algorithm, $\{e_N^a\}_{N\geq 1}$. These can be combined to compute

a plugin estimator for N^*, as we describe in Section 3.3. Finally, Section 3.4 develops a sequential hypothesis-testing procedure that delivers a one-sided confidence interval (CI) for N^*. To aid readability, this section focuses on the procedure itself, while Section 4 provides theoretical guarcrosses for these procedures.

3.1 Estimating the LLM Benchmark

We estimate the error of the LLM-based prediction rule, e_{LLM}, as its performance on the full data sample $\{(X_i, Y_i)\}_{i=1}^n$. For each observation X_i, obtain a prediction $\hat{Y}_i^{\text{LLM}} = f_{\text{LLM}}(X_i)$ by querying the LLM using the fixed prompt template described in Section 2.2. The empirical error of the LLM is then given by

$$\hat{e}_{\text{LLM}} \equiv \frac{1}{n} \sum_{i=1}^{n} \ell\left(\hat{Y}_i^{\text{LLM}}, Y_i\right). \tag{3.1}$$

3.2 Estimating the Error Curve of the ML Algorithm

We next estimate the algorithm's error curve by training it on disjoint training blocks of a given size and evaluating each fitted predictor on the held-out complement. Averaging the resulting out-of-sample losses produces an estimate of the algorithm's expected error as a function of the size of the training sample. Throughout this section, we drop dependence of e_N^a on the algorithm a to ease notation.

Given a dataset of size n, fix a candidate training size N_k for $k \in \{1, ..., K\}$. Set $B_k \equiv \lfloor n/N_k \rfloor$ to be the number of blocks and $M_n \equiv n - N_k$ to be the size of the test set. For simplicity, assume $n = B_k N_k$; otherwise, discard the last $n - B_k N_k = O(1)$ observations.[2]

Partition the index set $\{1, \ldots, n\}$ into B_k disjoint blocks $S_{k,1}, \ldots, S_{k,B_k}$, each of size N_k. For concreteness, take $S_{k,b} = \{(b-1)N_k + 1, \ldots, bN_k\}$ for $b = 1, \ldots, B_k$. For each block

[2]Discarding these $B_k N_k$ observations does not affect the later asymptotic theory.

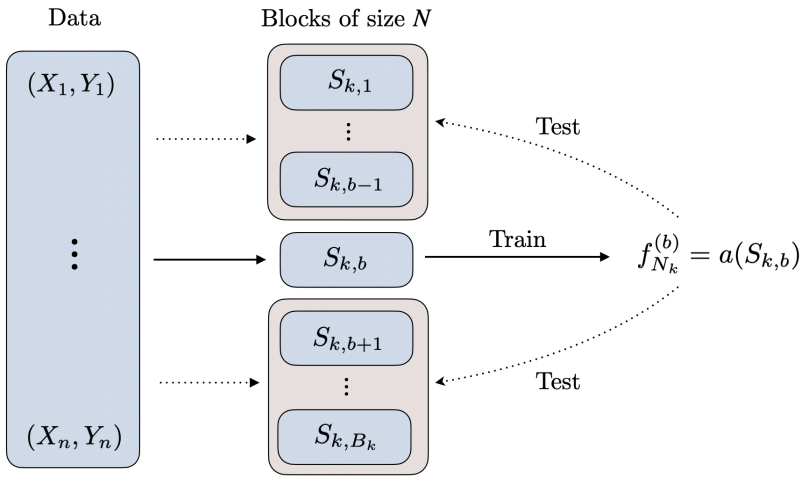


Figure 2: Depiction of block-out cross-validation: The data is split into B_k blocks each of size N_k. In each round, one block is used for training and the remaining are used for testing.

$b \in \{1, \ldots, B_k\}$, define the *block b training set* to be

$$\mathcal{D}_{\text{train}}^{(k,b)} \equiv \{(X_i, Y_i) : i \in S_{k,b}\}$$

and the *block b test set* to be

$$\mathcal{D}_{\text{test}}^{(k,b)} \equiv \{(X_i, Y_i) : i \notin S_{k,b}\}$$

as depicted in Figure 2. Because the observations (X_i, Y_i) are i.i.d. and the sets $S_{k,b}$ form a partition of $\{1, \ldots, n\}$, the training blocks $\mathcal{D}_{\text{train}}^{(k,1)}, \ldots, \mathcal{D}_{\text{train}}^{(k,B_k)}$ are independent and each has distribution P^{N_k}.

Recall that each training set is of size N_k while each test set is of size $M_n \equiv n - N_k$. In each round b, we fit the learning algorithm on the block b training set to obtain the block-wise predictor

$$f_{N_k}^{(b)} \equiv a\big(\mathcal{D}_{\text{train}}^{(k,b)}\big),$$

and evaluate its empirical error on the test complement:

$$\hat{e}_{k,b} \equiv \frac{1}{M_n} \sum_{i \notin S_{k,b}} \ell\big(f_{N_k}^{(b)}(X_i), Y_i\big).$$

The block-out CV estimator of e_{N_k} is finally

$$\hat{e}_{N_k}^{\mathrm{CV}} \equiv \frac{1}{B_k} \sum_{b=1}^{B_k} \hat{e}_{k,b}. \tag{3.2}$$

i.e., the average test error across the choice of training block.

3.3 Plugin Estimator for the ESS

Given the estimates (3.1) and (3.2), the plug-in estimator of the a-equivalent sample size is defined as

$$\hat{N}^* \equiv \min\big\{N_k : \hat{e}_{N_k}^{\mathrm{CV}} \leq \hat{e}_{\mathrm{LLM}},\ k \in \{1, ..., K\}\big\}$$

i.e., the smallest sample size at which the estimated LLM error $\hat{e}_{\mathrm{LLM}}$ exceeds the estimated error $\hat{e}_N^{\mathrm{CV}}$.

The next section defines a sequential estimator $\hat{N}_\alpha^*$ that incorporates sampling uncertainty and delivers a lower confidence bound for N^* with confidence level $1 - \alpha$. Under standard regularity conditions, both estimators are asymptotically equivalent, but $\hat{N}_\alpha^*$ is conservatively biased in finite samples to ensure correct coverage.

3.4 One-Sided Confidence Interval for the ESS

To develop a one-sided CI for N^*, we compare the predictive performance of the machine learning algorithm and the LLM across an increasing sequence of training sample sizes, and identify the smallest training size at which we can no longer conclude that the LLM outperforms the machine learning algorithm at the desired confidence level.

Let $\{N_k\}_{k=1}^K$ be an increasing sequence of candidate training sizes, with $N_1 < N_2 < \cdots <$

N_K. For each $k \in \{1, \ldots, K\}$, consider the one-sided hypotheses

$$H_{0,k} : e_{N_k} \leq e_{\text{LLM}}, \quad H_{1,k} : e_{N_k} > e_{\text{LLM}}.$$

Each hypothesis $H_{0,k}$ compares the predictive performance of the machine learning algorithm trained on N_k observations to that of the LLM, with the null asserting that the machine learning algorithm weakly outperforms the LLM at that training size. To test $H_{0,k}$, we use the test statistic

$$T_{N_k,n} \equiv \frac{\hat{e}^{\text{CV}}_{N_k} - \hat{e}_{\text{LLM}}}{\hat{\sigma}_k/\sqrt{n}}, \tag{3.3}$$

where $\hat{\sigma}_k^2$ is an estimator of the variance of $\hat{e}^{\text{CV}}_{N_k} - \hat{e}_{\text{LLM}}$, as detailed in (4.6) or (4.7) in Section 4.3 below. We reject $H_{0,k}$ at level α when $T_{N_k,n} > c_\alpha$ where c_α is the critical value.

We will be interested in the error curve $\{e_{N_k}\}_{k \geq 1}$, and our subsequent results are based on the following assumption on the monotonicity of the error curve on the grid $\{N_k\}$.

Assumption 3.1 (Monotonicity on Grid)**.** *e_{N_k} is nonincreasing in k.*

Assumption 3.1 can be viewed as a practical tuning requirement for the learning algorithm. Both theoretically and practically, it is possible that e_N is non-monotone in N, so increasing the training size may *decrease* expected performance: for example, the learning algorithm a may suffer from over-fitting when N is small, while under-fitting when N is large if the learning algorithm is implemented with ill-tuned hyperparameters. We emphasize that Assumption 3.1 is imposed only on the (possibly coarse) grid N_k, making it weaker than requiring monotonicity of e_N in N. While practical implementations of various learning algorithms may not always satisfy Assumption 3.1, such violations should be considered conceptually pathological given that, under the theoretical design of standard machine learning algorithms, more data should generally help with, but not hurt, a well-defined learning task. For these reasons, we maintain Assumption 3.1 in our subsequent analysis.

Given a monotone error curve, we can conduct inference on N^* using the following sequential testing procedure at overall level α:

Algorithm 3.1. *Proceed as follows:*

Step 1. *Test $H_{0,1}$ at level α. If $H_{0,1}$ is rejected, proceed to Step 2. Otherwise, stop and set $\hat{N}^*_\alpha = 1$.*

$\vdots$

Step k. *Test $H_{0,k}$ at level α. If $H_{0,k}$ is rejected, proceed to Step $k+1$. Otherwise, stop and set $\hat{N}^*_\alpha = N_{k-1} + 1$.*

$\vdots$

Step K. *Test $H_{0,K}$ at level α. If $H_{0,K}$ is rejected, conclude $N^* > N_K$ (and set $\hat{N}^*_\alpha = N_K + 1$). Otherwise, set $\hat{N}^*_\alpha = N_{K-1} + 1$.*

The procedure sequentially increases the training sample size of the machine-learning algorithm and tests, at each step, whether its predictive performance weakly exceeds that of the LLM. As long as the null hypothesis is rejected, the procedure continues to larger training sizes. At the first step k at which the corresponding null $H_{0,k}$ cannot be rejected, the procedure stops and sets the estimated equivalent sample size to $N_{k-1} + 1$, the smallest sample size exceeding the largest training size for which rejection occurred.[3] If all null hypotheses up to $H_{0,K}$ are rejected, the procedure concludes that the equivalent sample size exceeds N_K.

This stopping rule yields a valid lower confidence bound for the equivalent sample size N^*. In particular, suppose Assumption 3.1 holds and the level α test in each step of Algorithm 3.1 is valid. Then we show below that

$$\text{CI}_{N^*} \equiv \left[\hat{N}^*_\alpha, \infty\right) \tag{3.4}$$

is a one-sided $(1-\alpha)$-confidence interval for N^*. That is, $\Pr\left(\hat{N}^*_\alpha \leq N^*\right) \geq 1-\alpha$, i.e., with probability at least $1-\alpha$, one can conclude that the true equivalent sample size is no smaller than $\hat{N}^*_\alpha$. In practice, this allows one to say "N^* is at least as large as $\hat{N}^*_{.05}$ with 95%

[3] We set $\hat{N}^*_\alpha$ to $N_{k-1}+1$ rather than N_k to account for the possible coarseness of the grid (i.e., $\{N_1, ..., N_K\} \subsetneq \{1, ..., K\}$). When one employ the finest grid (i.e, $N_k = k$), then simply $N_{k-1} + 1 = N_k$.

confidence," so that we have a minimal guarcrosse on the value of the LLM prediction rule. The validity of the tests in Algorithm 3.1 and the CI (3.4) are established in Section 4.

Remark 3.1 (Alternative CI Representation)**.** *The CIs for N^* can be expressed in terms of CIs for the* excess risk *of the ML algorithm relative to the LLM (i.e., $e^a_{N_k} - e_{\text{LLM}}$). Recall that the algorithm stops at the first N_k where the null hypothesis $H_{0,k} : e^a_{N_k} \leq e_{\text{LLM}}$ is* not *rejected. By the duality between hypothesis tests and CIs, non-rejection at level α corresponds to the value 0 being contained in the one-sided CI for the difference $e^a_{N_k} - e_{\text{LLM}}$.*

Formally, let LB_k denote the lower bound of the asymptotic one-sided $(1-\alpha)$ CI for the excess risk, constructed from the test statistic (3.3).

$$LB_k \equiv \left(\hat{e}^{\text{CV}}_{N_k} - \hat{e}_{\text{LLM}}\right) - z_{1-\alpha}\frac{\hat{\sigma}_k}{\sqrt{n}}.$$

Assume the finest grid (i.e., $N_k = k$) for simplicity. The sequential estimator $\hat{N}^*_\alpha$ is simply the smallest sample size where this lower bound is non-positive (indicating that we cannot rule out that the ML model has matched the LLM):

$$\hat{N}^*_\alpha = \min\left\{N_k : LB_k \leq 0\right\} = \min\left\{N_k : \hat{e}^{\text{CV}}_{N_k} - z_{1-\alpha}\hat{\sigma}_k/\sqrt{n} \leq \hat{e}_{\text{LLM}}\right\}.^{4} \tag{3.5}$$

This expression holds regardless of whether the LLM's error is treated as fixed or estimated, provided the variance estimator $\hat{\sigma}_k$ correctly accounts for the variability of both terms.

4 Estimation and Inference: Theory

This section presents theoretical results justifying our proposed procedure in Section 3. It is organized as follows: First, Section 4.1 proves the validity of the sequential testing procedure described in Section 3.4, taking as given the validity of the individual tests of the hypotheses $H_{0,k} : e^a_{N_k} \leq e_{\text{LLM}}$. Section 4.2 then establishes a central limit theorem for the block-out CV

[4] For a generic coarser grid, $\hat{N}^*_\alpha = \max\left\{N_k : LB_k > 0\right\} + 1$.

error estimator of each $e^a_{N_k}$, which yields pointwise confidence intervals for the error curve $\{e^a_{N_k}\}_{k\geq 1}$. Finally, Section 4.3 uses this central limit theorem to establish the validity of individual test of each $H_{0,k}$.

4.1 Validity of Sequential Hypothesis Testing

The validity of Algorithm 3.1 in terms of overall size control is established as follows.

Theorem 4.1 (Validity of Sequential ESS Inference)**.** *Suppose that Assumption 3.1 holds. Assume that for each k, the test of $H_{0,k}$ employed in Algorithm 3.1 has asymptotic size controlled by α, i.e.,*

$$\limsup_{n\to\infty} \Pr(\textit{reject } H_{0,k} \mid H_{0,k} \textit{ is true}) \leq \alpha. \tag{4.1}$$

Then, the sequential testing procedure is valid at overall level α and the resulting CI for N^* has a correct coverage:

$$\liminf_{n\to\infty} \Pr(N^* \geq \hat{N}^*_\alpha) \geq 1-\alpha.$$

The proof of Theorem 4.1 is short but reveals a structural feature of the sequential test that avoids the size distortions typical of multiple hypothesis testing. We therefore present it in the main text.

Proof. Define $k^* \equiv \min\{k : e_{N_k} \leq e_{\text{LLM}}\}$ to be the index of the equivalent sample size in our sequence $\{N_1, ..., N_K\}$. Note that $N_{k^*-1}+1 \leq N^* \leq N_{k^*}$ by the coarseness of the grid $\{N_k\}$.

Write $\underline{N}_k \equiv N_{k-1}+1$ with $N_0 \equiv 0$. We will show that $[\hat{N}^*_\alpha, \infty)$ is a valid $(1-\alpha)$-CI for $\underline{N}_{k^*}$, i.e.,

$$\liminf_{n\to\infty} \Pr(\underline{N}_{k^*} \geq \hat{N}^*_\alpha) \geq 1-\alpha. \tag{4.2}$$

Then, since $\underline{N}_{k^*} \leq N^*$ and consequently $\{\underline{N}_{k^*} \geq \hat{N}^*_\alpha\} \subseteq \{N^* \geq \hat{N}^*_\alpha\}$, we can conclude that $[\hat{N}^*_\alpha, \infty)$ is also valid for N^* based on

$$\liminf_{n\to\infty} \Pr(N^* \geq \hat{N}^*_\alpha) \geq \liminf_{n\to\infty} \Pr(\underline{N}_{k^*} \geq \hat{N}^*_\alpha) \geq 1-\alpha. \tag{4.3}$$

We now prove (4.2). By Assumption 3.1, the sequence of null hypotheses $H_{0,k} : e_{N_k} \leq e_{\text{LLM}}$ has a nested truth structure:

(S1) For all $k < k^*$, we have $e_{N_k} > e_{\text{LLM}}$, so the null $H_{0,k}$ is *false*.

(S2) For all $k \geq k^*$, we have $e_{N_k} \leq e_{\text{LLM}}$, so the null $H_{0,k}$ is *true*.

Thus, H_{0,k^*-1} is the *last false null hypothesis* in the sequence, while H_{0,k^*} is the *first true null hypothesis* in the sequence.

Now, let $\hat{k}$ denote the random index at which Algorithm 3.1 terminates. The CI lower bound is defined as $\hat{N}^*_\alpha = \underline{N}_{\hat{k}}$. The algorithm stops at step k if it fails to reject $H_{0,k}$. Therefore, the event $\{\hat{k} = k\}$ implies that $H_{0,k}$ was not rejected, while all preceding hypotheses $H_{0,1}, \ldots, H_{0,k-1}$ were rejected. Consequently, the event $\{\hat{N}^*_\alpha > \underline{N}_{k^*}\} = \{\underline{N}_{\hat{k}} > \underline{N}_{k^*}\}$ is equivalent to $\{\hat{k} > k^*\}$.

For the algorithm to go beyond step k^* (i.e., for $\hat{k} > k^*$), it must have rejected the hypothesis H_{0,k^*}. Conversely, if H_{0,k^*} is rejected, then $\hat{k} > k^*$. Therefore,

$$\Pr(\hat{N}^*_\alpha > \underline{N}_{k^*}) = \Pr\left(\text{reject } H_{0,k^*}\right).$$

Since H_{0,k^*} is true by (S2), the rejection of H_{0,k^*} constitutes a Type I error (a false rejection). By the condition of the theorem, the test for each individual step controls the asymptotic size at level α. Specifically, for the step k^*, we have $\limsup_{n\to\infty} \Pr\left(\text{reject } H_{0,k^*}\right) \leq \alpha$.

Combining these results, we obtain the bound on the coverage probability:

$$\begin{aligned}\liminf_{n\to\infty} \Pr\left(N^* \geq \hat{N}^*_\alpha\right) &= 1 - \limsup_{n\to\infty} \Pr\left(\hat{N}^*_\alpha > N^*\right) \\ &\geq 1 - \limsup_{n\to\infty} \Pr\left(\text{reject } H_{0,k^*}\right) \geq 1 - \alpha.\end{aligned}$$

which implies the desired conclusion in view of (4.3). □

An important feature of this procedure is that it controls the family-wise error rate (FWER) (i.e., the probability of rejecting any true nulls) without requiring Bonferroni-

type corrections (e.g., Holm, 1979). This is because the sequence of hypotheses are ordered (nested) and the sequential testing procedure stops at the first non-rejection. Thus we only test true null hypotheses if we have correctly rejected all preceding false nulls, so the probability of making *any* false rejection is bounded simply by the probability of rejecting the *first* true null H_{0,k^*}. This property is particularly valuable given that we will typically want to evaluate a wide range of sample sizes.

Remark 4.1 (Impact of the Coarse Grid $\{N_k\}_{k=1}^K$)**.** *Let $\hat{N}^*_{\alpha,\text{fine}}$ denote the cutoff $\hat{N}^*_\alpha$ obtained under the "finest grid" $N_k = k$. Then $\hat{N}^*_{\alpha,\text{fine}}$ coincides with the smallest integer k at which we reject the null in Algorithm 3.1, which would produce a "sharp" $(1-\alpha)$ CI for N^* in the form of $[\hat{N}^*_{\alpha,\text{fine}}, \infty)$. For a generic coarse grid $\{N_k\}$, the cutoff $\hat{N}^*_\alpha$ is conservative against LLM in the sense of $\hat{N}^*_\alpha \leq \hat{N}^*_{\alpha,\text{fine}}$. Equivalently, we may view $[\hat{N}^*_{\alpha,\text{fine}}, \infty)$ as a sharp CI for $\underline{N}_{k^*} \leq N^*$, which results in conservative one-sided coverage for N^*.*

The validity of our sequential test, as established in Theorem 4.1, requires a valid test statistic at each step. Recall in Section 3.4 that we use the studentized test statistic (3.3) for the boundary of the null hypothesis, $\tilde{H}_{0,k} : e_{N_k} = e_{\text{LLM}}$, i.e., the null where the machine learning algorithm and the LLM have equal expected error. Since this is the least favorable configuration within the null—i.e., the case where false rejection is most likely—a level-α test of $\tilde{H}_{0,k}$ is also a valid level-α test of the original null hypothesis $H_{0,k}$. An asymptotic one-sided test rejects $H_{0,k}$ at level α if $T_{N_k,n} > z_{1-\alpha}$, where $z_{1-\alpha}$ is the $(1-\alpha)$-quantile of the standard normal distribution.[5] The validity of this individual test follows from Proposition 4.1 in Section 4.3, which is based on the Central Limit Theorem and consistent variance estimation in Section 4.2.

4.2 Central Limit Theorem under Block-Out Cross Validation

This section develops a Central Limit Theorem for risk estimators using the block-out CV procedure described in Section 3.2. Our established CLT in Theorem 4.2, and the correspond-

[5] We reject the null (that ML is as good as LLM) when the ML error is significantly *higher* than the LLM error. Thus, the rejection region is the right tail.

ing consistent variance estimator in Theorem 4.3 are established under very mild conditions without the requirement of stability conditions (Lei, 2025), and may be of independent interest beyond the scope of this paper.

Specifically, we consider a setting where researchers have access to a single large data set with sample size n. For each candidate training size N_k, we estimate the out-of-sample error $e^a_{N_k}$ of any learning algorithm a via block-out CV. We work with an asymptotic framework where $n \to \infty$ while N_k remains fixed.[6] These CIs in turn produce CIs for N^* as discussed in Remark 3.1.

We work under the following sampling framework and regularity conditions. Let $Z \equiv (X, Y) \sim P$ denote the distribution of observed data.

Assumption 4.1 (Random Sampling)**.** *A random sample of data $D_n \equiv (Z_1, ..., Z_n)$ is observed, with $Z_i \equiv (X_i, Y_i) \sim_{i.i.d.} Z \equiv (X, Y) \sim P$.*

For each given $N \leq n$ and a sample D_N of size N, we define a generic learning algorithm as a measurable map $a : \mathcal{Z}^N \to \mathcal{F}$, where $\mathcal{F}$ is a class of measurable functions $f : \mathcal{X} \to \mathcal{Y}$. For any training sample $D_N \in \mathcal{Z}^N$, recall that we write $f_N = a(D_N)$.

We consider a generic loss function ℓ that satisfies the following conditions:

Assumption 4.2 (Loss and Moments)**.** *The loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is measurable and, for $D_N \sim P^N$ and an independent $Z = (X, Y) \sim P$,*

$$\mathbb{E}\big[\,|\ell(f_N(X), Y)|^{2+\delta}\big] < \infty \quad \textit{for some } \delta > 0.$$

Recall that $e(f_N) \equiv \mathbb{E}_Z\big[\ell(f_N(X), Y) \mid D_N, \big]$ and $e^a_N \equiv \mathbb{E}_{D_N}[e(f_N)]$. In addition, for each realization $z = (x, y) \in \mathcal{Z}$, define

$$e^a_N(z) \equiv \mathbb{E}_{D_N}\big[\ell(f_{D_N}(x), y)\big],$$

[6] In Appendix B, we consider an alternative asymptotic framework where $N_k \to \infty$ proportionally to n (i.e., B is fixed), and establish the validity of our approach under an alternative set of assumptions.

where the expectation is over an independent copy $D_N \sim P^N$ with z fixed. Note that $e_N^a = \mathbb{E}_Z[e_N^a(Z)]$. Under Assumption 4.2, $e(f_N)$, $e_N^a(z)$, and e_N^a are well-defined with $\mathbb{E}\big[e(f_N)^2\big]$, $\mathbb{E}\big[e_N^a(Z)^2\big] < \infty$. We now present our main result.

Theorem 4.2 (CLT for block-out CV Risk Estimator)**.** *Suppose Assumptions 4.1-4.2 hold and N_k is fixed while $n \to \infty$. Then*

$$\sqrt{n}\big(\hat{e}_{N_k}^{\mathrm{CV}} - e_{N_k}^a\big) \xrightarrow{d} \mathcal{N}\big(0, \sigma_{N_k}^2\big),$$

where the asymptotic variance $\sigma_{N_k}^2$ is given by

$$\sigma_{N_k}^2 \equiv N_k V_{N_k,\mathrm{train}} + V_{N_k,\mathrm{test}} + 2N_k C_{N_k}, \tag{4.4}$$

with $V_{N_k,\mathrm{train}} \equiv \mathrm{Var}\big(e(f_{N_k})\big)$, $V_{N_k,\mathrm{test}} \equiv \mathrm{Var}\big(e_{N_k}^a(Z)\big)$, and $C_{N_k} \equiv \mathrm{Cov}\big(e(f_{N_k}), e_{N_k}^a(Z_1)\big)$, where $Z_1 \in \mathcal{D}_{N_k}$.

Importantly, we note that the asymptotic variance of the block-out CV estimator features three components: $V_{N_k,\mathrm{train}}$ that captures the training-sample randomness contribution, $V_{N_k,\mathrm{test}}$ that captures the testing-sample randomness contribution, and C_{N_k} that reflects the fact that each observation appears both in one training block and in the test sets of all other blocks, a key feature of cross-validation approaches.

4.2.1 Asymptotic Variance Estimation

We now construct an estimator $\hat{\sigma}_{N_k}^2$ of the asymptotic variance $\sigma_{N_k}^2$ by plugging in estimators for each component of the three-term decomposition in (4.4), as described below.

Estimator of $V_{N_k,\mathrm{train}}$ For each block b, define the block-wise population risk $e_{b,\mathrm{train}} \equiv \mathbb{E}\big[\ell(f_{N_k}^{(b)}(X), Y) \mid \mathcal{D}_{\mathrm{train}}^{(k,b)}\big]$. The $(e_{b,\mathrm{train}})_{b=1}^{B_k}$ are i.i.d. with mean $e_{N_k}^a$ and variance $V_{N_k,\mathrm{train}}$. We

estimate $V_{N_k,\text{train}}$ by the sample variance of the block-wise CV risks:

$$\hat{V}_{N_k,\text{train}} \equiv \frac{1}{B_k - 1} \sum_{b=1}^{B_k} \big(\hat{e}_{k,b} - \hat{e}_{N_k}^{\text{CV}}\big)^2.$$

Estimator of $V_{N_k,\text{test}}$ Each i belongs to exactly one training block and appears in the test sets of all other $B_k - 1$ blocks. Define the block-averaged test loss for observation i:

$$\hat{\mu}_i \equiv \frac{1}{B_k - 1} \sum_{b: i \notin S_{k,b}} \ell\big(f_{N_k}^{(b)}(X_i), Y_i\big), \quad i = 1, \ldots, n.$$

As $B_k \to \infty$, $\hat{\mu}_i \xrightarrow{p} e_{N_k}^a(Z_i)$. We estimate $V_{N_k,\text{test}}$ by:

$$\hat{V}_{N_k,\text{test}} \equiv \frac{1}{n-1} \sum_{i=1}^{n} \big(\hat{\mu}_i - \bar{\mu}\big)^2, \quad \bar{\mu} \equiv \frac{1}{n} \sum_{i=1}^{n} \hat{\mu}_i.$$

Estimator of C_{N_k} For each block b, define the block-wise summaries

$$\hat{e}_b \equiv \hat{e}_{k,b}, \quad \hat{m}_b \equiv \frac{1}{N_k} \sum_{i \in S_{k,b}} \hat{\mu}_i, \quad b = 1, \ldots, B_k.$$

Let $\bar{r} \equiv B_k^{-1} \sum_{b=1}^{B_k} \hat{e}_b$ and $\bar{m} \equiv B_k^{-1} \sum_{b=1}^{B_k} \hat{m}_b$. We estimate the cross term by:

$$\hat{C}_{N_k} \equiv \frac{1}{B_k - 1} \sum_{b=1}^{B_k} (\hat{e}_b - \bar{r})(\hat{m}_b - \bar{m}).$$

Combining the above, we construct the asymptotic variance estimator as

$$\hat{\sigma}_{N_k}^2 \equiv N_k \hat{V}_{N_k,\text{train}} + \hat{V}_{N_k,\text{test}} + 2N_k \hat{C}_{N_k}, \tag{4.5}$$

whose validity is established in the following theorem.

Theorem 4.3 (Studentized CLT)**.** *Suppose Assumptions 4.1 and 4.2 hold and N_k is fixed while $n \to \infty$. Let $\hat{\sigma}_{N_k}^2$ be defined by (4.5). Then:*

1. $\hat{\sigma}^2_{N_k} \xrightarrow{p} \sigma^2_{N_k}$ *as* $n \to \infty$.

2. *If* $\sigma^2_{N_k} > 0$, *then the studentized statistic satisfies*

$$\frac{\sqrt{n}\big(\hat{e}^{\mathrm{CV}}_{N_k} - e^a_{N_k}\big)}{\hat{\sigma}_{N_k}} \xrightarrow{d} \mathcal{N}(0,1).$$

4.3 A Valid Test of $H_{0,k} : e^a_{N_k} \leq e_{\mathrm{LLM}}$

We finally show how to construct a valid hypothesis test of $H_{0,k} : e^a_{N_k} \leq e_{\mathrm{LLM}}$ based on Theorem 4.3, which can then be used to implement the sequential testing procedure in Algorithm 3.1.

We base our test on the difference between the estimated risks:

$$\hat{\Delta}_k \equiv \hat{e}^{\mathrm{CV}}_{N_k} - \hat{e}_{\mathrm{LLM}}.$$

Since both terms are computed on the same dataset, they are correlated. To account for this correlation within the fixed-N_k asymptotic framework, we apply the general block-out CV theory directly to the *loss difference*. Define the loss difference for a given predictor f and observation $Z = (X, Y)$ as

$$\delta(f, Z) \equiv \ell(f(X), Y) - \ell(f_{\mathrm{LLM}}(X), Y).$$

Note that the estimator $\hat{\Delta}_k$ is algebraically equivalent to the block-out CV estimator applied to this difference loss function.[7] Applying Theorem 4.3 to the loss difference δ, we obtain the following result.

Proposition 4.1 (Test Validity)**.** *Define the test statistic:*

$$T_{\mathrm{diff},N_k,n} \equiv \frac{\hat{e}^{\mathrm{CV}}_{N_k} - \hat{e}_{\mathrm{LLM}}}{\hat{\sigma}_{\mathrm{diff},k}/\sqrt{n}}. \tag{4.6}$$

[7]Note that $\hat{e}_{\mathrm{LLM}}$ is the sample mean of the LLM loss. Since the LLM prediction rule is fixed, this is identical to the block-out CV estimator for a fixed function f_{LLM}.

where $\hat{\sigma}^2_{\text{diff},k}$ *estimates the asymptotic variance of the difference. Under Assumptions 4.1 and 4.2, under the boundary of the null hypothesis* ($e^a_{N_k} = e_{\text{LLM}}$)*:*

$$T_{\text{diff},N_k,n} \xrightarrow{d} \mathcal{N}(0,1) \quad \text{as } n \to \infty.$$

provided that $\sigma^2_{\text{diff},k} > 0$*. Consequently, the one-sided test that rejects* $H_{0,k}$ *when* $T_{N_k,n} > z_{1-\alpha}$ *has asymptotic size* α*.*

Remark 4.2 (Conservative Alternative)**.** *In practice, implementing the exact difference variance estimator* $\hat{\sigma}^2_{\text{diff},k}$ *requires computing covariances between ML and LLM errors. A simpler alternative is to use the sum of their individual variances, i.e.,*

$$T_{\text{con},N_k,n} \equiv \frac{\hat{e}^{\text{CV}}_{N_k} - \hat{e}_{\text{LLM}}}{\sqrt{\left(\hat{\sigma}^2_{N_k} + \hat{\sigma}^2_{\text{LLM}}\right)/n}}. \tag{4.7}$$

Note that $\sigma^2_{\text{diff},k} = \sigma^2_{N_k} + \sigma^2_{\text{LLM}} - 2\text{Cov}(\sqrt{n}\hat{e}^{\text{CV}}_{N_k}, \sqrt{n}\hat{e}_{\text{LLM}})$*. Assuming that the predictions of the ML model and the LLM are positively correlated (which is typical, as "hard" observations tend to yield high losses for both models), the covariance term is positive. In this case, the sum of variances overestimates the true variance* ($\hat{\sigma}^2_{N_k} + \hat{\sigma}^2_{\text{LLM}} > \hat{\sigma}^2_{\text{diff},k}$)*, making the test statistic* $T_{N_k,n}$ *smaller and the test conservative (size* $< \alpha$*).*

5 Applications

5.1 Data and Prediction Problems

The Panel Study of Income Dynamics (PSID) is one of the longest-running longitudinal household surveys in the United States, following individuals and families since 1968. It collects detailed information on economic, social, and health outcomes and is widely used in empirical research and policy analysis. We use the 2021 PSID wave and evaluate LLM prediction of four outcomes Y: hourly wage ($Y \in \mathbb{R}_+$), homeownership ($Y \in \{\text{own}, \text{rent}\}$),

smoking ($Y \in \{0,1\}$), and drinking ($Y \in \{0,1\}$). In each prediction problem, the covariate vector X denotes the respondent's characteristics, including age, sex, race, education, occupation, hours worked, labor market experience, self-reported health, state of residence, marital status, number of children, parents' education, religion, and the other outcomes listed above except for the one being predicted.

For the continuous-valued outcome (hourly wage), we set the loss function to be squared-error loss $\ell(y, y') = (y' - y)^2$, although for readability we report the *root mean-squared error*

$$\text{RMSE}(f) = \sqrt{e(f)}$$

rather than the error $e(f)$. For the other discrete outcomes (homeownership, drinking, and smoking) we set the usual 0-1 loss $\ell(y, y') = 1\{y' \neq y\}$, so error is the misclassification rate.

5.2 LLM Prediction Rules

We generate LLM predictions as follows. For each test observation $i = 1, \ldots, n$, we transform the covariate vector X_i into a natural-language *persona* description and then prompt the LLM to predict Y_i. Importantly, we do not fine-tune the LLM on PSID data and do not use few-shot examples drawn from PSID. Recall Prompt 1.

A natural concern is potential data leakage (Ludwig et al., 2025), namely that PSID data could have been included—directly or indirectly—in LLMs' training corpora. While access to PSID data requires registration and human verification through the PSID Data Center, we nonetheless implement two safeguards: (i) we evaluate a static LLM with a documented training data descriptions and cutoffs: OpenAI's ChatGPT-4 (cutoff: April 2023),[8] and (ii) we use the 2021 wave of PSID released on October 2024,[9] which postdates the reported knowledge cutoffs of the LLMs. (See Appendix E for supplementary results considering

[8] We access these models via an API. Based on interviews with current and former OpenAI employees, we expect that when ChatGPT is accessed via the API it is not subject to further fine-tuning (e.g., through reinforcement learning from human feedback), and that the reported knowledge cutoff is reliable.

[9] https://www.icpsr.umich.edu/web/sbeccc/studies/39190/versions/V1

additional LLMs whose cutoffs also predate the 2021 PSID release.)

5.3 Domain-Specific Learning

We compare the LLM's predictions to those of supervised ML models trained directly on PSID data. We consider two popular ML algorithms, which are briefly reviewed here.

Lasso regression and Logit L1. Lasso is a regularized linear prediction method. Different from standard OLS, it seeks to maximize fit to the data subject to a penalty for model complexity, evaluated as the sum of the absolute value of coefficients. Lasso regression thus encourages small coefficients, and often leads to sparse linear models in which many coefficients are set equal to zero. Logit L1 applies the same ℓ_1-regularization principle in a logistic regression framework for binary outcomes.

Random forests. Random forests are a flexible prediction method built by averaging many decision trees. Each component tree is trained on a different subsample of the data and uses a different subset of covariates, and the final prediction aggregates across them. This procedure allows the algorithm to capture nonlinear relationships and interactions without requiring these features to be specified in advance.

For each ML model, we compute the error curve, namely, the test error as a function of the training sample size N. We tune hyperparameters (e.g., the lasso penalty parameter, the depth and leaf size for RF) for every N using k-fold CV; see Section C in the Appendix for the implementation details. These error curve estimates are subsequently compared to the LLM benchmark to compute the ESS.

5.4 Results

We estimate error curves for each prediction problem and compute the equivalent sample size of the LLM. Figures 9–12 present the full estimated error curves, the estimated equivalent

sample size $\hat{N}^*$, and confidence intervals for all estimated quantities. The reported confidence intervals for the block-out CV errors are calculated as hybrid CIs: they are based on the fixed-N asymptotic theory (Theorem 4.2) when $N \leq 400$ and the fixed-B asymptotic theory (Corollary B.1 in Section B) otherwise.[10] The confidence intervals for N^* are calculated based on (3.5) in Remark 3.1. Table 1 more succinctly presents our confidence intervals for the equivalent sample size.

Table 1: Equivalent sample size across prediction tasks

Outcome	LLM Error	Algorithm	$\hat{N}^*$	One-Sided CI
Hourly Wage	12.50	Lasso	62	$[20, \infty)$
		RF	49	$[17, \infty)$
Home Ownership	0.24	Logit L1	NA	$[676, \infty)$
		RF	NA	$[590, \infty)$
Drinking	0.4	Logit L1	196	$[59, \infty)$
		RF	147	$[41, \infty)$
Smoking	0.23	Logit L1	1	$[1, \infty)$
		RF	1	$[1, \infty)$

We find substantial heterogeneity in the LLM's predictive performance across outcomes. The LLM performs extremely well in predicting homeownership: at the 95% confidence level, its equivalent sample size is at least 590 observations when benchmarked against a random forest and at least 676 observations when benchmarked against L1-penalized logit.

The LLM's performance is weaker, but still meaningful, for predicting drinking behavior. At the same confidence level, the LLM's equivalent sample size is at least 41 observations relative to the random forest and at least 59 observations relative to L1-penalized logit.

Predictive performance deteriorates further for hourly wages. Here, we can only conclude that the LLM's equivalent sample size is at least 17 observations when compared to the random forest and at least 20 observations when compared to L1-penalized logit.

Finally, the LLM provides essentially no predictive value for smoking behavior: we fail to reject the null hypothesis that conventional machine-learning algorithms outperform the

[10] This choice is made based on calibrated simulation to ensure desirable coverage performance.

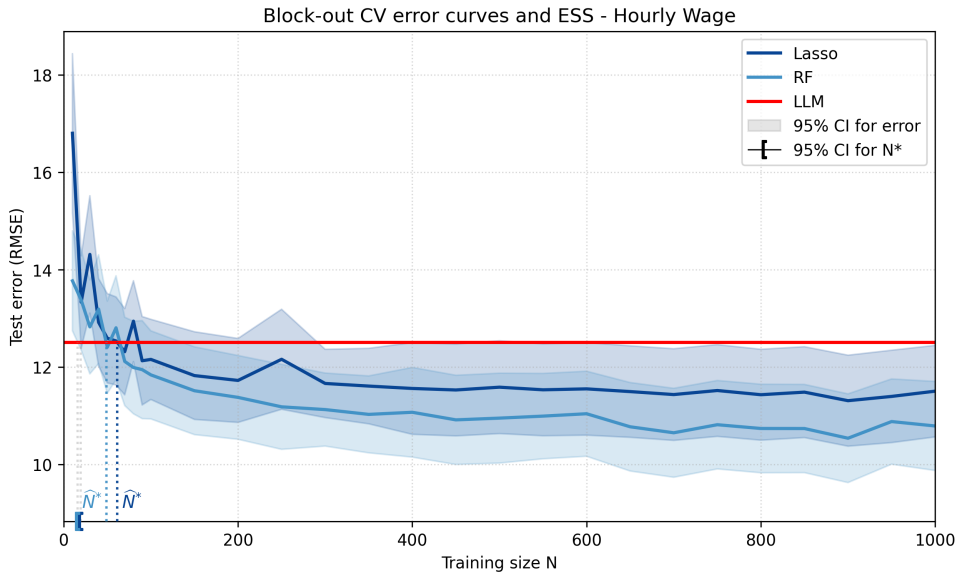


Figure 3: RMSE across training sample sizes (Y: Hourly wage)

LLM even when trained on a single observation.

Taken together, these results illustrate that the value of LLM predictions is highly task-specific—ranging from a close substitute for hundreds of observations in some settings to providing no meaningful predictive content in others. This underscores the importance of evaluating the LLM's predictive performance across domains.

6 Extensions

6.1 ESS with Multiple Domains

Consider a bank that faces many different prediction problems, e.g., predicting default risk for a new personal-loan product or fraud risk for a new account. For each prediction task, the bank can acquire data to train a specialized ML model. But if the number of such tasks is large, then a general-purpose LLM that can be deployed out of the box across tasks may be more valuable. This suggests that our baseline ESS measure—which is developed for a single, known prediction problem—may understate the predictive value of LLMs in settings with multiple tasks. This section extends ESS to such a setting.

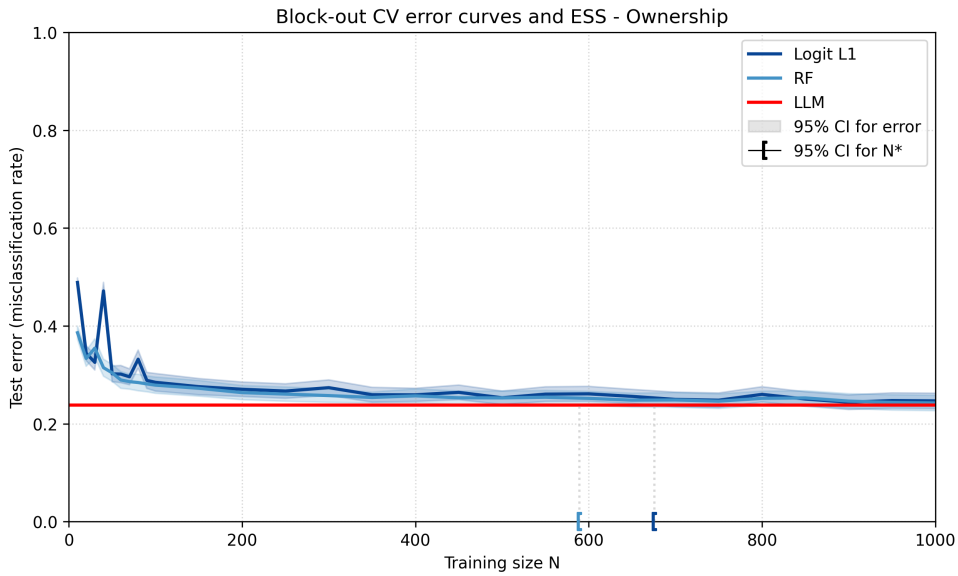


Figure 4: Misclassification rate across training sample sizes (Y: Homeownership)

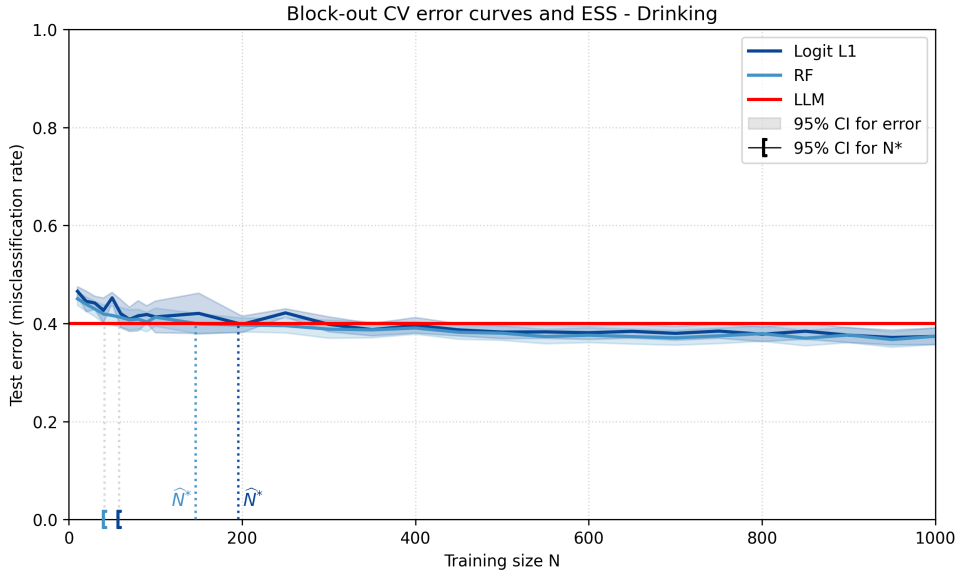


Figure 5: Misclassification rate across training sample sizes (Y: Drinking)

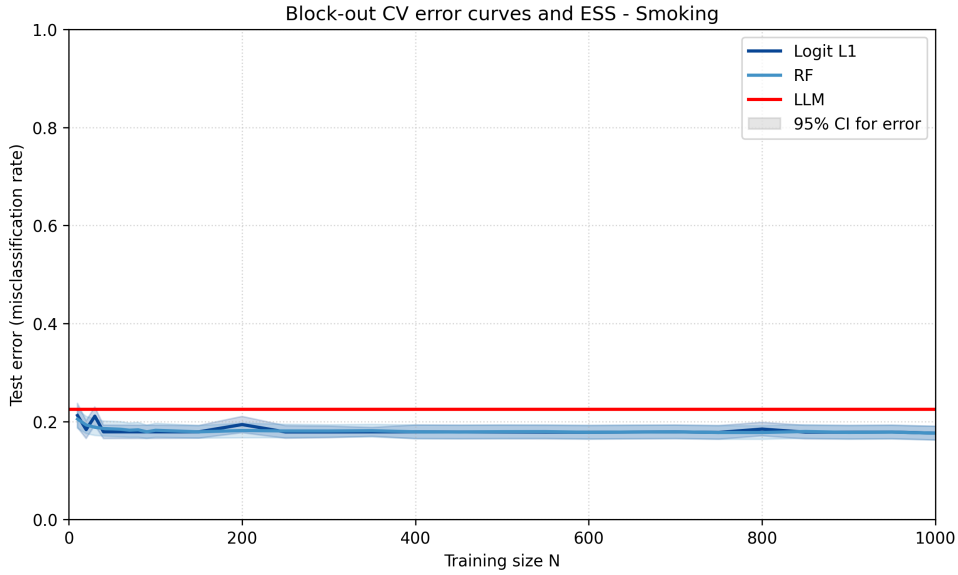


Figure 6: Misclassification rate across training sample sizes (Y: Smoking)

6.1.1 Setting

Prediction tasks, henceforth *domains*, are indexed by $i = 1, \ldots, m$, with associated covariate space $\mathcal{X}_i$, outcome space $\mathcal{Y}_i$, and domain-specific joint distribution $(X^{(i)}, Y^{(i)}) \sim P_i$, where $X^{(i)} \in \mathcal{X}_i$ and $Y^{(i)} \in \mathcal{Y}_i$. In each domain i, performance is evaluated using a domain-specific loss function $\ell_i : \mathcal{Y}_i \times \mathcal{Y}_i \to \mathbb{R}$.[11]

LLM prediction. The LLM induces a predictor $f^i_{\text{LLM}} : \mathcal{X}_i \to \mathcal{Y}_i$ in each domain i. Its expected error in domain i is

$$e^i_{\text{LLM}} := \mathbb{E}_{(X^{(i)}, Y^{(i)}) \sim P_i} \Big[\ell_i \big(f^i_{\text{LLM}}(X^{(i)}), Y^{(i)} \big) \Big].$$

Comparator. The analyst can instead collect a vector of domain-specific training sets

$$D_{\mathbf{N}} := (D^1_{N_1}, \ldots, D^m_{N_m}),$$

[11] More generally, one could begin with a loss function $L : \prod_{i=1}^m (\mathcal{Y}_i \times \mathcal{Y}_i) \to \mathbb{R}$ that maps a vector of predictions and realized outcomes into a scalar loss. The formulation in the text corresponds to the additively separable case $L(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i=1}^m p_i \, \ell_i(\hat{y}^{(i)}, y^{(i)})$, with $\hat{\mathbf{y}} = (\hat{y}^{(1)}, \ldots, \hat{y}^{(m)})$ and $\mathbf{y} = (y^{(1)}, \ldots, y^{(m)})$.

where each $D^i_{N_i}$ consists of N_i i.i.d. draws from P_i. Given this data, an algorithm

$$a : \prod_{i=1}^{m} \mathcal{D}_i \to \prod_{i=1}^{m} \mathcal{F}_i$$

produces predictors

$$(f^1_{\mathbf{N}}, \ldots, f^m_{\mathbf{N}}) \equiv a(D^1_{N_1}, \ldots, D^m_{N_m}),$$

where $\mathbf{N} = (N_1, \ldots, N_m)$ and each $f^i_{\mathbf{N}} : \mathcal{X}_i \to \mathcal{Y}_i$. The expected error of the resulting predictor in domain i is

$$e^i_{\mathbf{N}} := \mathbb{E}_{D_{\mathbf{N}}} \left[\mathbb{E}_{(X^{(i)}, Y^{(i)}) \sim P_i} \left[\ell_i \big(f^i_{\mathbf{N}}(X^{(i)}), Y^{(i)} \big) \right] \right].$$

6.1.2 Measure

For each domain i, define the stand-alone ESS

$$N^*_i := \min \left\{ N \in \mathbb{N}_+ : e^i_{(0,\ldots,0,N,0,\ldots,0)} \leq e^i_{\text{LLM}} \right\}.$$

This is the amount of domain-i training data needed for the ML algorithm to match the LLM's performance in domain i when trained only on data from that domain.

Our cross-domain extension instead requires the ML algorithm to match the LLM's performance in each domain i.

Definition 6.1 (Cross-Domain ESS)**.** *The* cross-domain ESS *is*

$$N^*_{\text{cross}} := \min \left\{ N \in \mathbb{N}_+ : \exists \mathbf{N} \textit{ such that } \sum_{i=1}^{m} N_i = N \textit{ and } e^i_{\mathbf{N}} \leq e^i_{\text{LLM}} \ \forall i \right\}.$$

6.1.3 Bounds on N^*_{cross}

We impose two assumptions. The first extends our earlier monotonicity condition to the multi-domain setting.

Assumption 6.1 (Monotonicity)**.** *For each domain i, the error $e^i_{\mathbf{N}}$ is weakly decreasing in each coordinate of $\mathbf{N}$.*

The second assumption says that, for improving performance in domain i, additional data from domain i itself is weakly more useful than any other way of allocating the same number of additional observations across domains.

Assumption 6.2 (Own-domain dominance)**.** *For any baseline allocation $\mathbf{N} = (N_1, \ldots, N_m)$, any increment $r \in \mathbb{N}_+$, and any domain i, define*

$$\mathbf{N}^{\,i+r} := (N_1, \ldots, N_{i-1}, N_i + r, N_{i+1}, \ldots, N_m).$$

Then for any alternative allocation $\mathbf{N}'$ satisfying

$$\sum_{j=1}^{m} N'_j = \sum_{j=1}^{m} N_j + r \qquad \text{and} \qquad N'_j \geq N_j \;\; \text{for all } j,$$

we have

$$e^i_{\mathbf{N}^{\,i+r}} \leq e^i_{\mathbf{N}'}.$$

Assumption 6.2 rules out "super-transfer" across domains. In particular, when the goal is to improve performance in domain i, it says that among all ways of allocating a given number of additional observations, assigning them all to domain i weakly minimizes domain-i error.

Proposition 6.1 (Bounds)**.** *Suppose Assumptions 6.1 and 6.2 hold, and each N^*_i is finite. Then*

$$\max_i N^*_i \leq N^*_{\text{cross}} \leq \textstyle\sum_{i=1}^{m} N^*_i$$

Moreover if $N^*_1 = \cdots = N^*_m =: N^*$, then

$$N^* \leq N^*_{\text{cross}} \leq mN^*.$$

These bounds are tight.

Proposition 6.1 shows that the cross-domain ESS lies between two natural extremes. In the most favorable case for transfer, the same training data is useful across all domains, so that matching the LLM simultaneously across domains requires no more data than the hardest stand-alone task, $\max_i N_i^*$. In the least favorable case, data is useful only within its own domain, so the analyst must effectively meet each domain-specific benchmark separately, yielding the upper bound $\sum_i N_i^*$. Thus the gap between the lower and upper bounds captures the extent to which training data generalizes across domains: the more domain-specific the learning problem, the closer N_{cross}^* is to the sum; the more transferable the information contained in the data, the closer it is to the maximum.

6.2 ESS for Treatment Effects

LLMs are considered useful tools for causal inference (Kiciman et al., 2023; Wang et al., 2024; Han, 2025; Liu et al., 2025). In light of this line of work, we extend our evaluation framework for LLMs that are employed for treatment effect estimation.

Let $\mu(X) \equiv \mathbb{E}_P[Y \mid X]$. Under squared loss $\ell(y, \hat{y}) = (y - \hat{y})^2$, the population risk admits the standard bias–variance-type decomposition

$$\begin{aligned} e(f) \equiv \mathbb{E}_P\big[(Y - f(X))^2\big] &= \mathbb{E}_P\big[(Y - \mu(X))^2\big] + \mathbb{E}_P\big[(\mu(X) - f(X))^2\big] \\ &\equiv e_{\text{irreducible}} + e_\mu(f), \end{aligned}$$

where the equality holds as the cross term is zero by the law of iterated expectations and the irreducible error is achieved as $\mu(X)$ uniquely minimizes $\mathbb{E}_P[(Y - f(X))^2]$. Therefore, any ESS defined with respect to $e(f)$ would be equivalent to an ESS defined with respect to $e_\mu(f)$.

Now consider the estimation of the conditional average treatment effect (CATE) with data (X, T, Y) distributed as P, where T is an indicator for an additional treatment. Assume unconfoundedness: $T \perp (Y_0, Y_1) \mid X$ where Y_t is the counterfactual outcome for $t \in \{0, 1\}$. Then, the conditional mean of Y_t is identified as $\mu_t(X) \equiv \mathbb{E}_P[Y_t \mid X] = \mathbb{E}_P[Y \mid X, T = t]$.

Within treatment arm d and under squared loss $\ell(y, \hat{y}) = (y - \hat{y})^2$, the conditional risk admits the standard bias–variance-type decomposition:

$$\begin{aligned} e_t(f) &\equiv \mathbb{E}_P\big[(Y_t - f(X))^2\big] \\ &= \mathbb{E}_P\big[(Y_t - \mu_t(X))^2\big] + \mathbb{E}_P\big[(\mu_t(X) - f(X))^2\big] \\ &\equiv e_{\text{irreducible},t} + e_{\mu_t}(f), \end{aligned}$$

by an argument analogous to above. Therefore, any ESS defined with respect to $e_t(f)$ would be equivalently an ESS defined with respect to $e_{\mu_t}(f)$. Moreover, $e_t(f)$ can be computed from observed data for each $T = t$, as $\mathbb{E}_P[(Y_t - f(X))^2] = \mathbb{E}_P\left[\mathbb{E}_P[(Y - f(X))^2 \mid X, T = t]\right]$ by unconfoundedness.

For the LLM, we can estimate $e_t(f_{\text{LLM}})$ by applying the prompt strategy in Section 2.2 to the test sample restricted to $T = t$ (i.e., to the treated or control test sample).[12] For ML trained on each sample of size N, define

$$e^a_{N,t} \equiv \mathbb{E}_{D_N}\, e_t(f_N) = \mathbb{E}_{D_N}\, e_t(a(D_N)), \quad D_N \equiv \{(x_i, t_i, y_i)\}_{i=1}^N,$$

where $e^a_{N,t}$ can be evaluated via block-out cross-validation within the $T = t$ subsample. Then, for $t \in \{0, 1\}$ the treatment-specific ESS is defined as

$$N^*_t \equiv \min\{N : \; e^a_{N,t} \leq e_t(f_{\text{LLM}})\}.$$

Define the CATE $\tau(X) \equiv \mathbb{E}_P[Y_1 - Y_0 \mid X]$ and the CATE risk

$$e_\tau(g) \equiv \mathbb{E}_P[(\tau(X) - g(X))^2].$$

For both the LLM and ML, using prediction rules (f_1, f_0) corresponding to $(e_{\mu_1}(f_1), e_{\mu_0}(f_0))$ only yields the upper bound on $e_\tau(f_1 - f_0)$ due to the nonlinearity of the loss function, and

[12] This approach is used to prompt LLMs to predict treatment effects for social interventions (Hewitt et al., 2024) and relates to counterfactual generation using LLMs in clinical trials (Wang et al., 2024).

thus defining ESS for τ through $f_1 - f_0$ is elusive.[13] We thus proceed as follows. Instead of prompting the LLM to predict Y_t, we prompt it to directly "predict" $\tau(X)$. An example prompt can be as follows:

Prompt 2. *Here is your persona: You are a 37-year-old male living in Tennessee in 2020. You work in architecture and engineering occupations in manufacturing industry. You have 4 years of work experience in the current job. You have completed 14 years of education. You are married and have 1 child. Your father has 12 years of education. Your mother has 12 years of education. Your race is white. Your health is excellent. You are Protestant.*

Finally, you are a subject in a randomized controlled trial in which the treatment is the provision of health insurance, specifically Medicaid.

Given your persona, what would be the effect of the randomized provision of health insurance on the number of your primary care visits?

Let $g_{\text{LLM}}(x)$ denote such a prediction. For the benchmark algorithm, let $\tilde{a} : \mathcal{D} \to \mathcal{G}$ map data to a CATE function $g_N \in \mathcal{G}$ and define

$$e^{\tilde{a}}_{N,\tau} \equiv \mathbb{E}[e_\tau(g_N)] = \mathbb{E}_{D_N} e_\tau(\tilde{a}(D_N)).$$

Then define the CATE-ESS as

$$N^*_\tau \equiv \min\{N : \ e^{\tilde{a}}_{N,\tau} \leq e_\tau(g_{\text{LLM}})\}.$$

In practice, because τ is unknown, we proceed via the transformed outcome. Under uncon-

[13] Specifically, we only have $e_\tau(f_1 - f_0) = \mathbb{E}_P\big[(\tau(X) - (f_1(X) - f_0(X)))^2\big] \leq 2\{e_{\mu_1}(f_1) + e_{\mu_0}(f_0)\}$.

foundedness, it can be shown that[14]

$$\mathbb{E}_P[\tilde{Y} \mid X] = \tau(X),$$

where

$$\tilde{Y} \equiv Y \frac{T - \pi(X)}{\pi(X)(1 - \pi(X))}, \quad \pi(X) \equiv \mathbb{P}[T = 1 \mid X].$$

Thus, for prediction g by either the LLM or ML model,

$$\begin{aligned} e_\tau(g) \equiv \mathbb{E}_P[(\tau(X) - g(X))^2] &= \mathbb{E}_P[(\tilde{Y} - g(X))^2] - \mathbb{E}_P[(\tilde{Y} - \tau(X))^2] \\ &\equiv \tilde{e}(g) - \tilde{e}_{\text{irreducible}}, \end{aligned}$$

where $\tilde{e}_{\text{irreducible}}$ does not depend on g. Consequently the ESS does not change if we replace e_τ by $\tilde{e}$ for the LLM and ML errors:

$$N_\tau^* = \tilde{N}^* \equiv \min\{N : \ \tilde{e}_N^{\tilde{a}} \leq \tilde{e}(g_{\text{LLM}})\}, \quad \tilde{e}_N^{\tilde{a}} \equiv \mathbb{E}_{D_N} \tilde{e}(\tilde{a}(D_N)).$$

In practice, the ML model can simply predict $\tilde{Y}$ based on X to obtain g_N. For both the LLM and ML model, $\tilde{e}(g)$ can be estimated by computing $\tilde{Y}$ (e.g., with known $\pi(X)$ in a randomized control trial) and contrasting it with g.

7 Conclusions

LLMs have become popular prediction tools in empirical social science. We measure the value of LLMs' pretrained knowledge using the *equivalent sample size* (ESS): the training

[14] Observe that

$$\begin{aligned} \mathbb{E}[\tilde{Y}|X] &= P(T=1|X)\mathbb{E}[\tilde{Y}|X,T=1] + P(T=0|X)\mathbb{E}[\tilde{Y}|X,T=0] \\ &= \pi(X)\left[\frac{\mathbb{E}[Y_1|X]}{\pi(X)}\right] + (1-\pi(X))\left[\frac{-\mathbb{E}[Y_0|X]}{1-\pi(X)}\right] = \mathbb{E}[Y_1|X] - \mathbb{E}[Y_0|X], \end{aligned}$$

where unconfoundedness is applied in the second equality.

sample size required for an algorithm trained on domain-specific data to match the LLM benchmark performance. We propose an inference framework for ESS based on sequential hypothesis tests and a new asymptotic theory for cross-validation error under block-out CV. The framework can help researchers quantify and communicate the capability (or limitation) of LLMs in domain-specific prediction tasks.

References

Aher, G. V., R. I. Arriaga, and A. T. Kalai (2023): "Using large language models to simulate multiple humans and replicate human subject studies," in *International conference on machine learning*, PMLR, 337–371.

Andrews, I., D. Fudenberg, L. Lei, A. Liang, and C. Wu (2025): "The Transfer Performance of Economic Models," Working paper.

Anthis, J. R., R. Liu, S. M. Richardson, A. C. Kozlowski, B. Koch, E. Brynjolfsson, J. Evans, and M. S. Bernstein (2025): "Position: LLM Social Simulations Are a Promising Research Method," in *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, position Paper Track.

Argyle, L. P., E. C. Busby, N. Fulda, J. R. Gubler, C. Rytting, and D. Wingate (2023): "Out of one, many: Using language models to simulate human samples," *Political Analysis*, 31, 337–351.

Austern, M. and W. Zhou (2020): "Asymptotics of Cross-Validation," *arXiv preprint arXiv:2001.11111*.

Bates, S., T. Hastie, and R. Tibshirani (2024): "Cross-validation: what does it estimate and how well does it do it?" *Journal of the American Statistical Association*, 119, 1434–1445.

Bohren, J. A., P. Hull, and A. Imas (2025): "Systemic Discrimination: Theory and Measurement," *The Quarterly Journal of Economics*, 140, 1743–1799.

Chan-Lau, J. A., T. L. Quach, and A. R. Shi (2025): "Forecasting Federal Funds Rates with AI: LSTM, GRU, and Large Language Model Approaches," Working paper (wp/25-10), ASEAN+3 Macroeconomic Research Office (AMRO), september 15, 2025.

Erev, I., A. E. Roth, R. L. Slonim, and G. Barron (2007): "Learning and Equilibrium as Useful Approximations: Accuracy of Prediction on Randomly Selected Constant Sum Games," *Economic Theory*, 33, 29–51.

Faria-e Castro, M. and F. Leibovici (2024): "Artificial Intelligence and Inflation Forecasts," *Federal Reserve Bank of St. Louis Review*, 106, 1–14.

Fava, B. (2025): "Training and Testing with Multiple Splits: A Central Limit Theorem for Split-Sample Estimators," *arXiv preprint arXiv:2511.04957*.

Fudenberg, D., W. Gao, and A. Liang (2026): "How Flexible Is That Functional Form? Measuring the Restrictiveness of Theories," *The Review of Economics and Statistics*, forthcoming.

Fudenberg, D., J. Kleinberg, A. Liang, and S. Mullainathan (2022): "Measuring the Completeness of Economic Models," *Journal of Political Economy*, 130, 956–990.

Gao, Y., D. Lee, G. Burtch, and S. Fazelpour (2025): "Take caution in using LLMs as human surrogates," *Proceedings of the National Academy of Sciences*, 122, e2501660122.

Han, S. (2025): "Mining causality: Ai-assisted search for instrumental variables," *arXiv preprint arXiv:2409.14202*.

Hewitt, L., A. Ashokkumar, I. Ghezae, and R. Willer (2024): "Predicting Results of Social Science Experiments Using Large Language Models," Working paper.

Holm, S. (1979): "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, 65–70.

Horton, J. J. (2023): "Large language models as simulated economic agents: What can we learn from homo silicus?" Tech. rep., National Bureau of Economic Research.

Huang, C., Y. Wu, and K. Wang (2025): "How Many Human Survey Respondents is a Large Language Model Worth? An Uncertainty Quantification Perspective," *arXiv preprint arXiv:2502.17773*.

Jansen, B. J., S.-g. Jung, and J. Salminen (2023): "Employing large language models in survey research," *Natural Language Processing Journal*, 4, 100020.

Kiciman, E., R. Ness, A. Sharma, and C. Tan (2023): "Causal reasoning and large language models: Opening a new frontier for causality," *Transactions on Machine Learning Research*.

Lei, J. (2025): "A Modern Theory of Cross-Validation through the Lens of Stability," *arXiv preprint arXiv:2505.23592*.

Liu, X., P. Xu, J. Wu, J. Yuan, Y. Yang, Y. Zhou, F. Liu, T. Guan, H. Wang, T. Yu, et al. (2025): "Large language models and causal inference in collaboration: A comprehensive survey," *Findings of the Association for Computational Linguistics: NAACL 2025*, 7668–7684.

Ludwig, J., S. Mullainathan, and A. Rambachan (2025): "Large language models: An applied econometric framework," Tech. rep., National Bureau of Economic Research.

Manning, B. S., K. Zhu, and J. J. Horton (2024): "Automated Social Science: Language Models as Scientist and Subjects," *National Bureau of Economic Research*.

Park, J. S., C. Q. Zou, A. Shaw, B. M. Hill, C. Cai, M. R. Morris, R. Willer, P. Liang, and M. S. Bernstein (2024): "Generative agent simulations of 1,000 people," *arXiv preprint arXiv:2411.10109*.

Sun, S., E. Lee, D. Nan, X. Zhao, W. Lee, B. J. Jansen, and J. H. Kim (2024): "Random silicon sampling: Simulating human sub-population opinion using a large language model based on group-level demographic information," *arXiv preprint arXiv:2402.18144*.

Van der Vaart, A. W. (2000): *Asymptotic statistics*, vol. 3, Cambridge university press.

Wang, Y., T. Fu, Y. Xu, Z. Ma, H. Xu, B. Du, Y. Lu, H. Gao, J. Wu, and J. Chen (2024): "TWIN-GPT: digital twins for clinical trials via large language model," *ACM Transactions on Multimedia Computing, Communications and Applications*.

A Block-Out CV under Fixed-N Asymptotics

This appendix contains proofs of the main asymptotic results stated in Section 4.2. Throughout, we fix the training size N_k and let $n \to \infty$. For notational simplicity, we suppress the subscript k and write N for N_k, B for B_k, S_b for $S_{k,b}$, etc.

Recall the following definitions. The block-out CV estimator is

$$\hat{e}_N^{\mathrm{CV}} = \frac{1}{B}\sum_{b=1}^{B} \hat{e}_b, \quad \hat{e}_b = \frac{1}{M_n}\sum_{i \notin S_b} \ell\big(f_N^{(b)}(X_i), Y_i\big),$$

where $M_n = n - N$ and $f_N^{(b)} = a(\mathcal{D}_{\mathrm{train}}^{(b)})$. For each block b, define the block-wise error

$$e_{b,\mathrm{train}} \equiv \mathbb{E}\big[\ell(f_N^{(b)}(X), Y) \mid \mathcal{D}_{\mathrm{train}}^{(b)}\big].$$

By construction, $(e_{b,\mathrm{train}})_{b=1}^{B}$ are i.i.d. with $\mathbb{E}[e_{b,\mathrm{train}}] = e_N^a$ and $\mathrm{Var}(e_{b,\mathrm{train}}) = V_{N,\mathrm{train}}$.

Proof of Theorem 4.2. Decompose the CV estimator as

$$\hat{e}^{\mathrm{CV}} \equiv A_n + B_n, \quad \text{with} \quad A_n \equiv \frac{1}{B}\sum_{b=1}^{B} e_{b,\mathrm{train}}, \; B_n \equiv \frac{1}{B}\sum_{b=1}^{B}(\hat{e}_b - e_{b,\mathrm{train}}).$$

First, we consider the training contribution A_n. The training blocks $\mathcal{D}_{\text{train}}^{(b)} = (Z_{(b-1)N+1}, \ldots, Z_{bN})$ are i.i.d across b and thus $e_{b,\text{train}}$ are i.i.d. with $\mathbb{E}[e_{b,\text{train}}] = e_N^a$ and $\text{Var}(e_{b,\text{train}}) = V_{N,\text{train}}$. By the classical CLT,

$$\sqrt{B}(A_n - e_N^a) \xrightarrow{d} \mathcal{N}(0, V_{N,\text{train}}).$$

Since $B = n/N$, we have

$$\sqrt{n}(A_n - e_N^a) = \sqrt{N} \cdot \sqrt{B}(A_n - e_N^a) \xrightarrow{d} \mathcal{N}(0, NV_{N,\text{train}}).$$

Second, we consider the testing contribution B_n. Writing $\psi(D_N, z) \equiv \ell(f_{D_N}(x), y) - e(f_N)$. Then for each block b and each $i \notin S_b$, $\ell(f_N^{(b)}(X_i), Y_i) - e_{b,\text{train}} = \psi(\mathcal{D}_{\text{train}}^{(b)}, Z_i)$ and

$$B_n = \frac{1}{BM_n} \sum_{b=1}^{B} \sum_{i \notin S_b} \psi(\mathcal{D}_{\text{train}}^{(b)}, Z_i).$$

For each pair (b, i) with $i \notin S_b$, the training set $\mathcal{D}_{\text{train}}^{(b)}$ and the test point Z_i are independent, and thus

$$\mathbb{E}\big[\psi(\mathcal{D}_{\text{train}}^{(b)}, Z_i) \mid Z_i\big] = e_N^a(Z_i) - e_N^a.$$

In the meanwhile, note that

$$\mathbb{E}[\psi(\mathcal{D}_{\text{train}}^{(b)}, Z_i) \mid \mathcal{D}_{\text{train}}^{(b)}] = 0.$$

We now define T_n as the first-order Hájek projection of B_n:

$$T_n \equiv \sum_{j=1}^{n} \mathbb{E}[B_n \mid Z_j].$$

Since each Z_j serves as a test point for exactly $B - 1$ blocks, we obtain:

$$T_n = \sum_{j=1}^{n} \frac{B-1}{BM_n}\big(e_N^a(Z_j) - e_N^a\big) = \frac{B-1}{BM_n} \sum_{i=1}^{n} \big(e_N^a(Z_i) - e_N^a\big) = \frac{1}{n} \sum_{i=1}^{n} \big(e_N^a(Z_i) - e_N^a\big).$$

since $\frac{B-1}{BM_n} = \frac{B-1}{B(BN-N)} = \frac{1}{n}$.

Define $\mathcal{R}_n \equiv B_n - T_n$, where

$$\mathcal{R}_n = \frac{1}{BM_n} \sum_{b=1}^{B} \sum_{i \notin S_b} \eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i),$$

for a function η satisfying the degeneracy conditions

$$\mathbb{E}[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i) \mid Z_i] = 0, \quad \mathbb{E}[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i) \mid \mathcal{D}_{\text{train}}^{(b)}] = 0.$$

Expanding the second moment as

$$\mathbb{E}[\mathcal{R}_n^2] = \frac{1}{(BM_n)^2} \sum_{b,b'=1}^{B} \sum_{\substack{i \notin S_b \\ j \notin S_{b'}}} \mathbb{E}\big[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i)\, \eta(\mathcal{D}_{\text{train}}^{(b')}, Z_j)\big],$$

we observe that:

- For $b = b'$ and $i \neq j$, the second degeneracy condition yields

$$\mathbb{E}\big[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i)\, \eta(\mathcal{D}_{\text{train}}^{(b)}, Z_j)\big] = \mathbb{E}\Big[\mathbb{E}\big[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i) \mid \mathcal{D}_{\text{train}}^{(b)}, Z_j\big]\, \eta(\mathcal{D}_{\text{train}}^{(b)}, Z_j)\Big] = 0.$$

- For $b \neq b'$ with $S_b \cap S_{b'} = \emptyset$ and $i \neq j$, independence gives

$$\mathbb{E}\big[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i)\, \eta(\mathcal{D}_{\text{train}}^{(b')}, Z_j)\big] = \mathbb{E}\big[\eta(\mathcal{D}_{\text{train}}^{(b)}, Z_i)\big]\, \mathbb{E}\big[\eta(\mathcal{D}_{\text{train}}^{(b')}, Z_j)\big] = 0.$$

The only non-vanishing components are the diagonal terms ($b = b'$, $i = j$) and cross-block terms where training/test indices overlap. The number of such terms is $O(n^2)$, while each term is $O(1)$. Combined with the normalization $(BM_n)^{-2} = O(n^{-2})$, we have

$$\mathbb{E}[\mathcal{R}_n^2] = O(n^{-2}).$$

Then, by Chebyshev's inequality,

$$\mathbb{P}(|\sqrt{n}\mathcal{R}_n| > \epsilon) \leq \frac{n\,\mathbb{E}[\mathcal{R}_n^2]}{\epsilon^2} = O(n^{-1}) \to 0,$$

hence $\sqrt{n}\mathcal{R}_n \xrightarrow{p} 0$.

Third, we consider the covariance between A_n and B_n. Since $\sqrt{n}\mathcal{R}_n \xrightarrow{p} 0$ and $\sqrt{n}(A_n - e_N^a) = O_p(1)$, we have

$$\mathrm{Cov}\big(\sqrt{n}(A_n - e_N^a), \sqrt{n}\mathcal{R}_n\big) \to 0.$$

Hence, it suffices to study the covariance between A_n and T_n. Using the definition of T_n,

$$\mathrm{Cov}(A_n, T_n) = \frac{B-1}{B^2 M_n} \sum_{b=1}^{B} \sum_{i=1}^{n} \mathrm{Cov}\big(e_{b,\mathrm{train}}, e_N^a(Z_i)\big).$$

If $i \notin S_b$, then $e_{b,\mathrm{train}}$ depends only on $\{Z_j : j \in S_b\}$ whereas $e_N^a(Z_i)$ depends only on Z_i and an independent training sample, so $e_{b,\mathrm{train}}$ and $e_N^a(Z_i)$ are independent and their covariance is 0. If $i \in S_b$, then the pair $(e_{b,\mathrm{train}}, e_N^a(Z_i))$ has the same distribution as $(e(f_N), e_N^a(Z_1))$ with $D_N = (Z_1, \ldots, Z_N) \sim P^N$. Hence,

$$\mathrm{Cov}\big(e_{b,\mathrm{train}}, e_N^a(Z_i)\big) = C_N \quad \text{for all } b \text{ and } i \in S_b.$$

Observing that there are exactly $BN = n$ pairs (b, i) with $i \in S_b$, we deduce

$$\mathrm{Cov}(A_n, T_n) = \frac{B-1}{B^2 M_n} \cdot BN \cdot C_N = \frac{B-1}{B} \cdot \frac{N}{M_n} \cdot C_N = \frac{N}{n} C_N.$$

and consequently

$$\mathrm{Cov}\big(\sqrt{n}(A_n - e_N^a), \sqrt{n}T_n\big) = n\,\mathrm{Cov}(A_n, T_n) \to N C_N.$$

Third, we consider the covariance between A_n and B_n. Since $\sqrt{n}\mathcal{R}_n \xrightarrow{p} 0$ and $\sqrt{n}(A_n -$

$e_N^a) = O_p(1)$, we have

$$\mathrm{Cov}\big(\sqrt{n}(A_n - e_N^a), \sqrt{n}\mathcal{R}_n\big) \to 0.$$

Hence, it suffices to study the covariance between A_n and T_n. Using the definition of T_n,

$$\mathrm{Cov}(A_n, T_n) = \frac{B-1}{B^2 M_n} \sum_{b=1}^{B} \sum_{i=1}^{n} \mathrm{Cov}\big(e_{b,\mathrm{train}}, e_N^a(Z_i)\big).$$

If $i \notin S_b$, then the independence of $\{Z_j : j \in S_b\}$ and Z_i implies $\mathrm{Cov}(e_{b,\mathrm{train}}, e_N^a(Z_i)) = 0$. If $i \in S_b$, then the pair $(e_{b,\mathrm{train}}, e_N^a(Z_i))$ has the same distribution as $(e(f_N), e_N^a(Z_1))$ with $D_N = (Z_1, \ldots, Z_N) \sim P^N$, and hence $\mathrm{Cov}\big(e_{b,\mathrm{train}}, e_N^a(Z_i)\big) = C_N$. Observing that there are exactly $BN = n$ pairs (b, i) with $i \in S_b$, we deduce

$$\mathrm{Cov}(A_n, T_n) = \frac{B-1}{B^2 M_n} \cdot BN \cdot C_N = \frac{B-1}{B} \cdot \frac{N}{M_n} \cdot C_N = \frac{N}{n} C_N.$$

and consequently

$$\mathrm{Cov}\big(\sqrt{n}(A_n - e_N^a), \sqrt{n}T_n\big) = n\,\mathrm{Cov}(A_n, T_n) \to N C_N.$$

Finally, combining the analysis above, we deduce

$$\sqrt{n}(A_n - e_N^a) \xrightarrow{d} \mathcal{N}(0, N V_{N,\mathrm{train}}), \quad \sqrt{n}B_n \xrightarrow{d} \mathcal{N}(0, V_{N,\mathrm{test}}),$$

and the joint limit is bivariate normal with covariance $\lim_n \mathrm{Cov}(\sqrt{n}(A_n - e_N^a), \sqrt{n}B_n) = NC_N$. Thus

$$\sqrt{n}(\hat{e}^{\mathrm{CV}} - e_N^a) = \sqrt{n}(A_n - e_N^a) + \sqrt{n}B_n \xrightarrow{d} \mathcal{N}(0, \sigma_N^2),$$

with $\sigma_N^2 = N V_{N,\mathrm{train}} + V_{N,\mathrm{test}} + 2NC_N$. □

The proof of Theorem 4.3 relies on showing that each component of the plug-in variance estimator is consistent. We establish this through three lemmas.

Lemma A.1. *Under Assumptions 4.1, 4.2, and $n = BN$,*

$$\hat{V}_{N,\text{train}} \xrightarrow{p} V_{N,\text{train}}.$$

Proof of Lemma A.1. Write $\hat{e}_b = e_{b,\text{train}} + U_b$ where

$$U_b \equiv \frac{1}{M_n} \sum_{i \notin S_b} W_i, \quad W_i \equiv \ell(f_N^{(b)}(X_i), Y_i) - e_{b,\text{train}}.$$

Conditional on $\mathcal{D}_{\text{train}}^{(b)}$, the variables $\{W_i\}_{i \notin S_b}$ are i.i.d. with mean zero. By Assumption 4.2, $\mathbb{E}[|W_i|^{2+\delta}] < \infty$ for some $\delta > 0$.

By the conditional Marcinkiewicz-Zygmund inequality and $M_n = n - N = O(n)$,

$$\mathbb{E}\big[|U_b|^{2+\delta}\big] = O\big(M_n^{-(1+\delta/2)}\big) = O\big(n^{-(1+\delta/2)}\big).$$

Since $B = n/N = O(n)$, we have by Markov's inequality

$$\Pr\left(\max_{1 \le b \le B} |U_b| > \epsilon\right) \le B \cdot \frac{\mathbb{E}[|U_b|^{2+\delta}]}{\epsilon^{2+\delta}} = O(n) \cdot O(n^{-(1+\delta/2)}) = O(n^{-\delta/2}) \to 0.$$

Thus $\max_b |U_b| \xrightarrow{p} 0$.

Since $(e_{b,\text{train}})_{b=1}^B$ are i.i.d. with variance $V_{N,\text{train}}$, we have $S_B^2(e_{1,\text{train}}, \ldots, e_{B,\text{train}}) \xrightarrow{p} V_{N,\text{train}}$ by the LLN. By the continuity of S_B^2 and $\max_b |\hat{e}_b - e_{b,\text{train}}| \xrightarrow{p} 0$, we have

$$S_B^2(\hat{e}_1, \ldots, \hat{e}_B) \xrightarrow{p} V_{N,\text{train}}. \qquad \square$$

Lemma A.2. *Under Assumptions 4.1, 4.2, and $n = BN$,*

$$\hat{V}_{N,\text{test}} \xrightarrow{p} V_{N,\text{test}}.$$

Proof of Lemma A.2. Recall $\hat{\mu}_i = \frac{1}{B-1} \sum_{b: i \notin S_b} \ell(f_N^{(b)}(X_i), Y_i)$. Conditional on Z_i, the

summands are i.i.d. with mean $e_N^a(Z_i)$ and variance $v_N(Z_i) \equiv \mathrm{Var}(\ell(f_N(X), Y) \mid Z = Z_i)$. Then, since $\mathbb{E}[v_N(Z)] < \infty$ by Assumption 4.2, we have

$$\mathbb{E}\big[(\hat{\mu}_i - e_N^a(Z_i))^2\big] = \frac{\mathbb{E}[v_N(Z)]}{B-1} = O(n^{-1}).$$

Also, note that

$$\frac{1}{n}\sum_{i=1}^n \hat{\mu}_i^2 = \frac{1}{n}\sum_{i=1}^n e_N^a(Z_i)^2 + \frac{1}{n}\sum_{i=1}^n \big(\hat{\mu}_i^2 - e_N^a(Z_i)^2\big).$$

The first term converges to $\mathbb{E}[e_N^a(Z)^2]$ by the LLN. For the second term, we have

$$\mathbb{E}\big|\hat{\mu}_i^2 - e_N^a(Z_i)^2\big| \le \mathbb{E}\big[(\hat{\mu}_i - e_N^a(Z_i))^2\big]^{1/2} \cdot \mathbb{E}\big[(\hat{\mu}_i + e_N^a(Z_i))^2\big]^{1/2} = O(n^{-1/2}) \cdot O(1) = O(n^{-1/2}),$$

since

$$\mathbb{E}\big[(\hat{\mu}_i + e_N^a(Z_i))^2\big]^{1/2} \le \mathbb{E}\big[(\hat{\mu}_i - e_N^a(Z_i))^2\big]^{1/2} + 2\mathbb{E}\big[e_N^a(Z)^2\big]^{1/2} = O(1).$$

Similarly, $\frac{1}{n}\sum_{i=1}^n \hat{\mu}_i \xrightarrow{p} e_N^a$, and together we have

$$\hat{V}_{N,\mathrm{test}} = \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^n \hat{\mu}_i^2 - \bar{\mu}^2\right) \xrightarrow{p} \mathbb{E}[e_N^a(Z)^2] - (e_N^a)^2 = V_{N,\mathrm{test}}. \qquad \square$$

Lemma A.3. *Under Assumptions 4.1, 4.2, and $n = BN$,*

$$\hat{C}_N \xrightarrow{p} C_N.$$

Proof of Lemma A.3. Define the errors $\Delta e_b \equiv \hat{e}_b - e_{b,\mathrm{train}}$ and $\Delta m_b \equiv \hat{m}_b - \tilde{m}_b$. From the proof of Lemma A.1, $\max_b |\Delta e_b| \xrightarrow{p} 0$.

Now, note that $\Delta m_b = \frac{1}{N}\sum_{i \in S_b}(\hat{\mu}_i - e_N^a(Z_i))$, and

$$\frac{1}{B}\sum_{b=1}^B (\Delta m_b)^2 \le \frac{1}{BN}\sum_{b=1}^B \sum_{i \in S_b} (\hat{\mu}_i - e_N^a(Z_i))^2 = \frac{1}{n}\sum_{i=1}^n (\hat{\mu}_i - e_N^a(Z_i))^2.$$

Since $\mathbb{E}[(\hat{\mu}_i - e_N^a(Z_i))^2] = O(n^{-1})$ in the proof of Lemma A.2), we have by Markov's inequality

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_i - e_N^a(Z_i))^2 \xrightarrow{p} 0.$$

Now, consider

$$\hat{C}_N - \tilde{C}_N = \frac{1}{B-1}\sum_{b=1}^{B}\big(\hat{e}_b - \bar{\hat{e}}\big)\big(\hat{m}_b - \bar{\hat{m}}\big) - \frac{1}{B-1}\sum_{b=1}^{B}\big(e_{b,\text{train}} - \bar{e}_{\text{train}}\big)\big(\tilde{m}_b - \bar{\tilde{m}}\big).$$

Note that, for any sequences $(a_b), (a'_b), (c_b), (c'_b)$, we have

$$\begin{aligned}
&\frac{1}{B-1}\sum_{b=1}^{B}(a_b - \bar{a})(c_b - \bar{c}) - \frac{1}{B-1}\sum_{b=1}^{B}(a'_b - \bar{a}')(c'_b - \bar{c}') \\
&\quad = \frac{1}{B-1}\sum_{b=1}^{B} a_b c_b - \frac{B}{B-1}\bar{a}\bar{c} - \frac{1}{B-1}\sum_{b=1}^{B} a'_b c'_b + \frac{B}{B-1}\bar{a}'\bar{c}'.
\end{aligned}$$

Applying this with $(a_b, c_b) = (\hat{e}_b, \hat{m}_b)$ and $(a'_b, c'_b) = (e_{b,\text{train}}, \tilde{m}_b)$, we obtain

$$\hat{C}_N - \tilde{C}_N = \frac{1}{B-1}\sum_{b=1}^{B}\big(\hat{e}_b\hat{m}_b - e_{b,\text{train}}\tilde{m}_b\big) - \frac{B}{B-1}\big(\bar{\hat{e}}\,\bar{\hat{m}} - \bar{e}_{\text{train}}\bar{\tilde{m}}\big).$$

Substituting $\hat{e}_b = e_{b,\text{train}} + \Delta e_b$ and $\hat{m}_b = \tilde{m}_b + \Delta m_b$, we have

$$\hat{e}_b\hat{m}_b - e_{b,\text{train}}\tilde{m}_b = e_{b,\text{train}}\Delta m_b + \tilde{m}_b\Delta e_b + \Delta e_b\Delta m_b,$$

and

$$\bar{\hat{e}}\,\bar{\hat{m}} - \bar{e}_{\text{train}}\bar{\tilde{m}} = \bar{e}_{\text{train}}\overline{\Delta m} + \bar{\tilde{m}}\overline{\Delta e} + \overline{\Delta e}\,\overline{\Delta m} + \overline{\Delta e}\bar{\tilde{m}} + \bar{e}_{\text{train}}\overline{\Delta m},$$

where $\overline{\Delta e} = \frac{1}{B}\sum_b \Delta e_b$ and $\overline{\Delta m} = \frac{1}{B}\sum_b \Delta m_b$. Together, we have

$$\begin{aligned}
\hat{C}_N - \tilde{C}_N = &\frac{1}{B-1}\sum_{b=1}^{B} e_{b,\text{train}}\Delta m_b + \frac{1}{B-1}\sum_{b=1}^{B}\tilde{m}_b\Delta e_b + \frac{1}{B-1}\sum_{b=1}^{B}\Delta e_b\Delta m_b \\
&- \frac{B}{B-1}\big(\bar{e}_{\text{train}}\overline{\Delta m} + \bar{\tilde{m}}\overline{\Delta e} + \overline{\Delta e}\,\overline{\Delta m}\big).
\end{aligned}$$

in which each term vanishes in probability:

For the first term,

$$\left|\frac{1}{B}\sum_{b=1}^{B} e_{b,\text{train}}\Delta m_b\right| \le \left(\frac{1}{B}\sum_{b=1}^{B} e_{b,\text{train}}^2\right)^{1/2}\left(\frac{1}{B}\sum_{b=1}^{B}(\Delta m_b)^2\right)^{1/2} = O_p(1)\cdot o_p(1).$$

For the second term,

$$\left|\frac{1}{B}\sum_{b=1}^{B} \tilde{m}_b\Delta e_b\right| \le \left(\frac{1}{B}\sum_{b=1}^{B} \tilde{m}_b^2\right)^{1/2}\left(\frac{1}{B}\sum_{b=1}^{B}(\Delta e_b)^2\right)^{1/2} \le O_p(1)\cdot \max_b|\Delta e_b| = o_p(1).$$

For the third term,

$$\left|\frac{1}{B}\sum_{b=1}^{B} \Delta e_b\Delta m_b\right| \le \max_b|\Delta e_b|\cdot\frac{1}{B}\sum_{b=1}^{B}|\Delta m_b| = o_p(1)\cdot O_p(1).$$

For the remaining terms, they are bounded similarly using $|\overline{\Delta e}| \le \max_b|\Delta e_b|$ and $|\overline{\Delta m}|^2 \le \frac{1}{B}\sum_b(\Delta m_b)^2$. Hence

$$\hat{C}_N - \tilde{C}_N \xrightarrow{p} 0.$$

Finally, $(e_{b,\text{train}}, \tilde{m}_b)_{b=1}^B$ are i.i.d., so $\tilde{C}_N \xrightarrow{p} C_N$ by the LLN. Thus $\hat{C}_N \xrightarrow{p} C_N$. □

Proof of Theorem 4.3. By Lemmas A.1, A.2, and A.3,

$$N\hat{V}_{N,\text{train}} \xrightarrow{p} NV_{N,\text{train}}, \quad \hat{V}_{N,\text{test}} \xrightarrow{p} V_{N,\text{test}}, \quad N\hat{C}_N \xrightarrow{p} NC_N.$$

From the variance decomposition (4.4), we obtain

$$\hat{\sigma}_N^2 = N\hat{V}_{N,\text{train}} + \hat{V}_{N,\text{test}} + 2N\hat{C}_N \xrightarrow{p} NV_{N,\text{train}} + V_{N,\text{test}} + 2NC_N = \sigma_N^2,$$

establishing part (1).

For part (2), Theorem 4.2 gives $\sqrt{n}(\hat{e}^{\text{CV}} - e_N^a) \xrightarrow{d} \mathcal{N}(0, \sigma_N^2)$. Since $\hat{\sigma}_N^2 \xrightarrow{p} \sigma_N^2 > 0$ by

part (1), by the Slutsky's Theorem, we have

$$\frac{\sqrt{n}(\hat{e}^{\mathrm{CV}} - e_N^a)}{\hat{\sigma}_N} \xrightarrow{d} \mathcal{N}(0,1).$$

□

B Block-Out CV under Fixed-B Asymptotics

The asymptotic results in Section 4.2 rely on the regime where the training size N is fixed while the number of blocks $B = n/N \to \infty$ (the fixed-N asymptotic regime). Here, we consider an alternative setting where the number of folds B is small and fixed (e.g., $B = 5$ or 10), while the sample size N within each block is large.

We define the *fixed-B asymptotic regime* as the asymptotic framework where B is fixed and $N \to \infty$ (implying $n \to \infty$). Under this regime, the estimator $\hat{e}^{\mathrm{CV}}$ aggregates losses from B highly stable predictors. The variance estimator $\hat{\sigma}^2_{N_k}$ based on the fixed-N asymptotic regime can perform poorly here because it relies on estimating the between-block covariance structure from very few (B) observations.

In this section, we derive a Central Limit Theorem (CLT) for the fixed-B asymptotic regime by exploiting the algorithmic stability of the learner as $N \to \infty$. This approach leverages recent results on cross-validation stability (Lei, 2025), specifically the behavior of risk gradients for deterministic centering.

Recall that the block-out cross-validation estimator is given by:

$$\hat{e}^{\mathrm{CV}} = \frac{1}{n}\sum_{i=1}^{n} \tilde{\ell}_i, \quad \text{where} \quad \tilde{\ell}_i = \frac{1}{B-1}\sum_{b \neq b(i)} \ell\big(f_N^{(b)}(X_i), Y_i\big),$$

where $b(i)$ denotes the block containing observation i. Note that $\tilde{\ell}_i$ is the average loss of observation i evaluated by the $B-1$ models that did not include i in their training set. Let $e_N^a = \mathbb{E}[\ell(f_{D_N}(X), Y)]$ be the population prediction error for a training set of size N.

To establish asymptotic normality centered at e_N^a, we require conditions on the stability of the learning algorithm. Following Lei (2025), we distinguish between the stability of the loss function and the stability of the risk (expected loss). Define the conditional risk of a model trained on dataset D_N as $R(D_N) = \mathbb{E}_{\tilde{Z}}[\ell(f_{D_N}(\tilde{X}), \tilde{Y})]$, where the expectation is taken wrt an independent copy $\tilde{Z} \equiv (\tilde{X}, \tilde{Y})$.

Assumption B.1 (Algorithm Stability)**.** *Let $D_N^{(1)} \equiv (Z_1', Z_2, \ldots, Z_N)$ denote the dataset with the first observation replaced by an independent copy $Z_1' \sim P$, and let $\nabla_1 R(D_N) \equiv R(D_N) - R(D_N^{(1)})$ denote the Perturb-One Stability, i.e., the change in conditional risk when one training point is replaced by an independent copy.*[15] *We assume that the algorithm satisfies*

$$\|\nabla_1 R\|_{L_2} = o(N^{-1}) \text{ as } N \to \infty. \tag{B.1}$$

Assumption B.1 corresponds to the condition in Proposition 4.11.(1) of Lei (2025), where the variability induced by training data vanishes relative to the evaluation noise. Lei (2025) also provides arguments[16] that in machine learning contexts, the risk function usually enjoys much better stability than the loss function, making risk stability requirement in our Assumption B.1 plausible.

Theorem B.1 (CLT under Fixed-B Asymptotics)**.** *Under Assumptions 4.1, 4.2, and B.1, as $N \to \infty$ with B fixed:*

$$\sqrt{n}\big(\hat{e}^{\mathrm{CV}} - e_N^a\big) \xrightarrow{d} \mathcal{N}(0, \tau^2),$$

where $\tau^2 = \mathrm{Var}\big(\mathbb{E}[\ell(f_{D_N}(X), Y) \mid Z]\big)$ is the variance of the conditional expected loss given a test point $Z = (X, Y)$.

Proof of Theorem B.1. We adapt Proposition 4.11(1) of Lei (2025) to our block-out CV design using the Hájek projection technique.

[15] See Definition 2.3 in Lei (2025) for details.
[16] See the arguments after Theorem 4.7 (pp.54-55) in Lei (2025).

Define $L_n \equiv \sum_{i=1}^n \tilde{\ell}_i = n\hat{e}^{\mathrm{CV}}$. The first-order Hájek projection of L_n is

$$\hat{L}_n \equiv \sum_{j=1}^n \big(\mathbb{E}[L_n \mid Z_j] - \mathbb{E}[L_n]\big).$$

We decompose $\mathbb{E}[L_n \mid Z_j]$ into

$$\mathbb{E}[L_n \mid Z_j] = \mathbb{E}[\tilde{\ell}_j \mid Z_j] + \sum_{i \neq j} \mathbb{E}[\tilde{\ell}_i \mid Z_j].$$

For the first term, Z_j serves as the test point. Since $\tilde{\ell}_j$ averages over models trained on blocks not containing j, we have

$$\mathbb{E}[\tilde{\ell}_j \mid Z_j] = \mathbb{E}[\ell(f_{D_N}(X_j), Y_j) \mid Z_j] \equiv l_N(Z_j).$$

For the second term in the form of a sum over $i \neq j$, the point Z_j appear in training sets used to evaluate $\tilde{\ell}_i$. Define

$$g_N(Z_j) \equiv \sum_{i \neq j} \big(\mathbb{E}[\tilde{\ell}_i \mid Z_j] - \mathbb{E}[\tilde{\ell}_i]\big),$$

which captures the influence of Z_j on all other observations. Together, the projection becomes

$$\hat{L}_n = \sum_{j=1}^n \big(l_N(Z_j) - \mathbb{E}[l_N(Z)] + g_N(Z_j)\big).$$

By the Efron-Stein inequality Lei (2025, Theorem 4.1), we have

$$\mathrm{Var}(g_N(Z_1)) \leq C\big(N\|\nabla_1 R\|_{L_2}\big)^2,$$

Under Assumption B.1, $N\|\nabla_1 R\|_{L_2} = o(1)$, so $\mathrm{Var}(g_N(Z_1)) \to 0$. Since $l_N(Z_j)$ and $g_N(Z_j)$

are functions of Z_j alone, $\hat{L}_n$ is a sum of i.i.d. mean-zero terms with variance

$$\mathrm{Var}(\hat{L}_n) = n\,\mathrm{Var}\big(l_N(Z_1) + g_N(Z_1)\big) = n\big(\tau^2 + o(1)\big),$$

where $\tau^2 \equiv \mathrm{Var}(l_N(Z_1))$.

For the remainder $R_n \equiv L_n - \hat{L}_n$, standard results on Hájek projections (Van der Vaart, 2000, Theorem 11.2) yield $\mathrm{Var}(R_n) = o(n)$, hence $R_n = o_p(\sqrt{n})$.

Applying the CLT to $\hat{L}_n/\sqrt{n}$ and using $L_n = \hat{L}_n + o_p(\sqrt{n})$, we have

$$\frac{1}{\sqrt{n}}\big(L_n - \mathbb{E}[L_n]\big) = \frac{1}{\sqrt{n}}\sum_{j=1}^{n}\big(l_N(Z_j) - \mathbb{E}[l_N(Z)]\big) + o_p(1) \xrightarrow{d} \mathcal{N}(0, \tau^2).$$

Since $\mathbb{E}[L_n] = n\,\mathbb{E}[\tilde{\ell}_1] = n\,e_N^a$, the result follows. □

Corollary B.1 (CLT with Variance Estimation)**.** *Under the conditions of Theorem B.1,* $\hat{\tau}_B^2 \xrightarrow{p} \tau^2$, *and*

$$\frac{\sqrt{n}(\hat{e}^{\mathrm{CV}} - e_N^a)}{\hat{\tau}_B} \xrightarrow{d} \mathcal{N}(0,1).$$

Proof of Corollary B.1. Write $\tilde{\ell}_i = l_N(Z_i) + \eta_i$, where $\eta_i \equiv \tilde{\ell}_i - l_N(Z_i)$. By the stability argument in the proof of Theorem B.1, $\mathbb{E}[\eta_i^2] \to 0$ as $N \to \infty$. In addition,

$$\hat{\tau}_B^2 = \frac{1}{n-1}\sum_{i=1}^{n}\big(\tilde{\ell}_i - \hat{e}^{\mathrm{CV}}\big)^2 = \frac{1}{n-1}\sum_{i=1}^{n}\big(l_N(Z_i) - \bar{l}_N\big)^2 + o_p(1),$$

where $\bar{l}_N = \frac{1}{n}\sum_{i=1}^{n} l_N(Z_i)$. By the LLN, the leading term converges to $\mathrm{Var}(l_N(Z_1)) = \tau^2$. Then Theorem B.1 and Slutsky's lemma imply the result. □

C Details of ML Implementation

Outcomes and feature construction. We apply the following pre-processing to all ML methods we use. For earnings as the target outcome, we winsorize it prior to analysis at the

1st and 99th percentiles to reduce sensitivity to extreme outliers.[17] The continuous predictors are standardized, and the categorical predictors are handled as follows: (i) rare categories are collapsed into a single "other" level when their frequency falls below a minimum-count threshold (e.g., 50), and (ii) the resulting categorical variables are converted to indicators (with one category dropped as a reference group). In addition, we augment X with other outcomes as predictors (continuous outcomes enter as standardized numeric features; categorical outcomes enter via indicators), excluding the current target and respecting the earnings/wage exclusion noted above.

Learning-curve design and block-out CV. Fix a training size N. After randomly permuting the analysis sample once for reproducibility, we partition the data into $B = \lfloor n/N \rfloor$ disjoint blocks of size N. For each block $b = 1, \ldots, B$, we fit the ML model using only the observations in block b as the training set, generate predictions for all observations, and evaluate loss on the complement of block b. Averaging the out-of-block losses across b yields the block-out CV estimate of predictive risk at training size N. Repeating this procedure over a grid of N values produces the error curve.

Model classes and per-N hyperparameter tuning. For each N, we tune hyperparameters on a tuning subset of size N drawn at random from the analysis sample, and then apply the selected hyperparameters in the block-out CV loop.[18] For regression (continuous Y), we consider lasso and random forest. The lasso penalty parameter is selected by `LassoCV` using 5-fold CV. The random forest uses 300 trees and is tuned by 3-fold CV over a small grid of structural parameters (maximum depth in $\{\texttt{None}, 10, 20\}$ and minimum leaf size in $\{1, 5\}$), with performance scored in terms of RMSE. For classification (discrete Y), we consider ℓ_1-penalized logistic regression (logit lasso) and random forest classification.

[17] We alternatively train the ML models on $\log(1+Y)$ while calculating the error on the original level scale.

[18] This approach is useful in stabilizing the error curve across N, although to faithfully follow the definition of e_N^a, we may want to tune hyperparameters for each training block in the block-out CV loop. Still, the current approach may be acceptable as the tuning is low-dimensional and held fixed across CV folds for a given N.

Logistic regression is fit using the `saga` solver and tuned over a grid of inverse-regularization parameters $C \in \{0.01, 0.1, 1, 10\}$ using stratified CV; random forest classification uses 300 trees and is tuned over the same depth/leaf grid as in regression using stratified CV.

For neural networks used in the preliminary analysis in Section E, we fit a shallow fully-connected feedforward network with two ReLU hidden layers (64 and 32 units) and an output layer that is linear for regression and softmax for classification (with the number of outputs set to the number of classes observed in the training data). Networks are trained with the Adam optimizer using MSE for regression (100 epochs) and categorical cross-entropy for classification (10 epochs, with one-hot encoded labels).

Small-N safeguards for classification. For very small N, stratified CV and/or block-level training sets can become degenerate due to severe class imbalance. We therefore implement explicit safeguards: (i) the number of folds in stratified CV is chosen adaptively and never exceeds the minimum class count (capped at 3 folds), and (ii) if a block-level training set contains only one class, we default to a majority-class prediction rule for that block.

Uncertainty quantification. We report uncertainty measures derived from the variance estimator introduced in Section 4: the cross-block variability of observation-level out-of-block losses yields an estimated variance component $\hat{\sigma}_k^2$, which is converted into standard errors of the CV risk via $\sqrt{\hat{\sigma}_k^2/n}$. For $\hat{\sigma}_k^2$, as mentioned in the main text, we use a hybrid approach based on the fixed-N asymptotic theory (Theorem 4.2) when $N \leq 400$ and the fixed-B asymptotic theory (Corollary B.1 in Section B) otherwise. For RMSE, we apply a delta-method transformation from MSE to RMSE to obtain confidence intervals.

D Proof of Proposition 6.1

We first derive the upper bound. Consider the allocation $\overline{\mathbf{N}} := (N_1^*, \ldots, N_m^*)$. In each domain i, monotonicity implies $e^i_{\overline{\mathbf{N}}} \leq e^i_{(0,\ldots,0,N_i^*,0,\ldots,0)}$. Moreover, by the definition of N_i^*, $e^i_{(0,\ldots,0,N_i^*,0,\ldots,0)} \leq e^i_{\text{LLM}}$. Hence $e^i_{\overline{\mathbf{N}}} \leq e^i_{\text{LLM}}$ for all i, implying $N^*_{\text{cross}} \leq \sum_{i=1}^m N_i^*$.

Next we prove the lower bound. Call $T \in \mathbb{Z}_+$ feasible if there exists an allocation $\mathbf{N} = (N_1, \ldots, N_m)$ such that

$$\sum_{j=1}^{m} N_j = T \qquad \text{and} \qquad e^i_{\mathbf{N}} \leq e^i_{\text{LLM}} \quad \forall i.$$

By definition, N^*_{cross} is the smallest feasible T.

Fix any domain i, and suppose towards contradiction that some feasible T satisfies $T < N^*_i$. By the definition of N^*_i,

$$e^i_{(0,\ldots,0,T,0,\ldots,0)} > e^i_{\text{LLM}}.$$

By Assumption 6.2, among all allocations of total size T, the allocation assigning all T observations to domain i weakly minimizes domain-i error. Therefore, for any allocation $\mathbf{N}$ satisfying $\sum_j N_j = T$,

$$e^i_{(0,\ldots,0,T,0,\ldots,0)} \leq e^i_{\mathbf{N}}.$$

Combining the two displays yields

$$e^i_{\mathbf{N}} > e^i_{\text{LLM}},$$

so no allocation of total size T is feasible, a contradiction. Hence every feasible T must satisfy $T \geq N^*_i$ for every i, and therefore

$$N^*_{\text{cross}} \geq \max_i N^*_i.$$

Finally, if $N^*_1 = \cdots = N^*_m = N^*$, then the lower bound becomes $N^*_{\text{cross}} \geq \max_i N^*_i = N^*$, and the upper bound becomes $N^*_{\text{cross}} \leq \sum_{i=1}^m N^*_i = mN^*$. This gives

$$N^* \leq N^*_{\text{cross}} \leq mN^*.$$

It remains to show that the bounds are attainable. For the upper bound, suppose that

only data gathered from domain i is useful for predicting in domain i, so that

$$e^i_{(N_1,\ldots,N_{i-1},N_i,N_{i+1},\ldots,N_m)} = e^i_{(0,\ldots,0,N_i,0,\ldots,0)} \quad \forall (N_j)_{j\neq i} \in \mathbb{Z}_+^{m-1}. \tag{D.1}$$

Suppose toward contradiction that some $T < \sum_i N_i^*$ is feasible. Then there exists an allocation $\mathbf{N} = (N_1, \ldots, N_m)$ such that $\sum_i N_i = T$ and $e^i_{\mathbf{N}} \leq e^i_{\text{LLM}}$ for every i. Since $T < \sum_i N_i^*$, there exists some domain j with $N_j < N_j^*$. Hence, by (D.1),

$$e^j_{\mathbf{N}} = e^j_{(0,\ldots,0,N_j,0,\ldots,0)}.$$

But since $N_j < N_j^*$, the definition of N_j^* implies

$$e^j_{(0,\ldots,0,N_j,0,\ldots,0)} > e^j_{\text{LLM}},$$

contradicting feasibility. Therefore $N^*_{\text{cross}} = \sum_{i=1}^m N_i^*$.

For the lower bound, let $k \in \arg\max_i N_i^*$, and consider an environment in which all domains depend only on the total sample size:

$$e^i_{\mathbf{N}} = h_i\left(\sum_{j=1}^m N_j\right),$$

where each h_i is weakly decreasing and satisfies

$$h_i(N_k^*) \leq e^i_{\text{LLM}} \qquad \text{and} \qquad h_i(n) > e^i_{\text{LLM}} \ \text{ for all } n < N_i^*.$$

Again Assumption 6.1 holds, and Assumption 6.2 holds with equality, since any two allocations with the same total sample size induce the same error in every domain. Under this construction, any allocation with total size $N_k^* = \max_i N_i^*$ is feasible, while no smaller total sample size can be feasible because domain k would remain above the LLM benchmark.

Therefore

$$N^*_{\text{cross}} = \max_i N^*_i,$$

so the lower bound is attained.

E Additional Results for the Applications

E.1 Other Prediction Methods

Here we compare LLM performance to ML performance at a fixed training size for the ML models: 800 observations. In addition to ChatGPT-4, we report performance for Meta's Llama 3 models (Llama3-8B cutoff: March 2023; Llama3-70B cutoff: December 2023). We find that predictive performance is broadly similar across LLMs and, separately, across machine-learning algorithms, while varying systematically across outcomes. This pattern suggests that our main findings are not driven by the choice of a particular LLM or comparator algorithm.

Table 2: Equivalent sample size across prediction tasks

Outcome	LLM Error	Algorithm	$\hat{N}^*$	One-Sided CI
Hourly Wage	15	Lasso	17	$[13, \infty)$
		RF	1	$[1, \infty)$
Ownership	0.26	Logit L1	357	$[150, \infty)$
		RF	268	$[122, \infty)$
Drinking	0.41	Logit L1	185	$[39, \infty)$
		RF	88	$[38, \infty)$
Smoking	0.24	Logit L1	1	$[1, \infty)$
		RF	1	$[1, \infty)$

E.2 Other Error Metrics

In the main text, prediction error for discrete outcomes is measured using the 0-1 loss function. Misclassification rate can be mechanically low for outcomes that are very skewed, which might

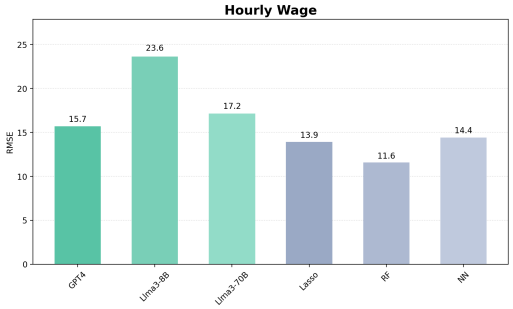


Figure 7: Root Mean Squared Error (lower is better) ($N = 800$)

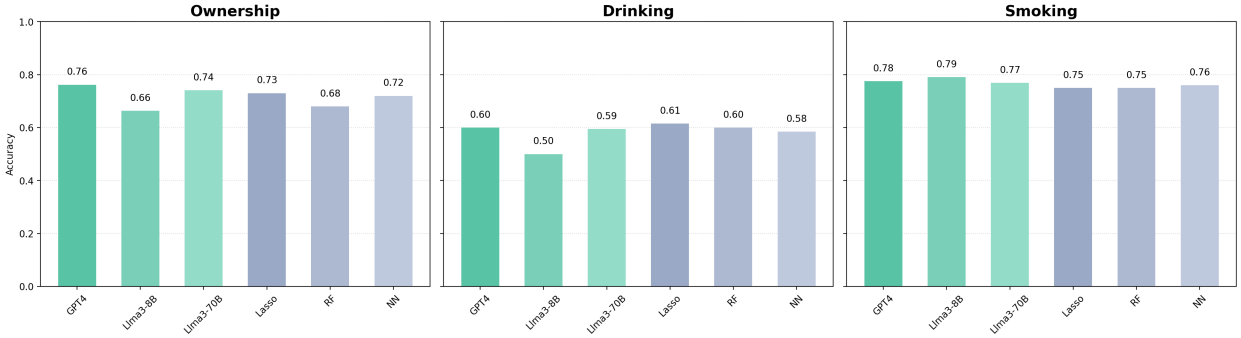


Figure 8: Accuracy (higher is better) ($N = 800$)

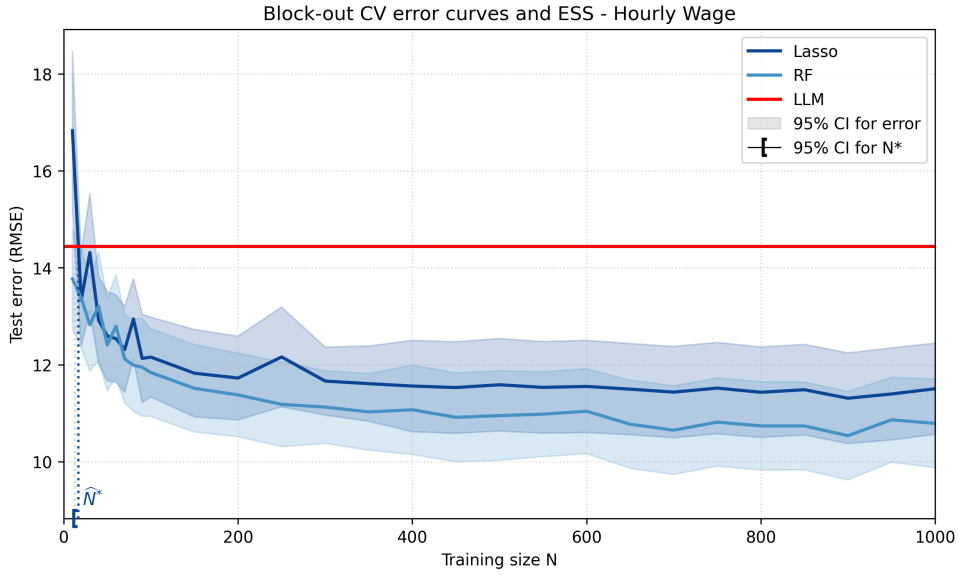


Figure 9: RMSE across training sample sizes (Y: Hourly wage)

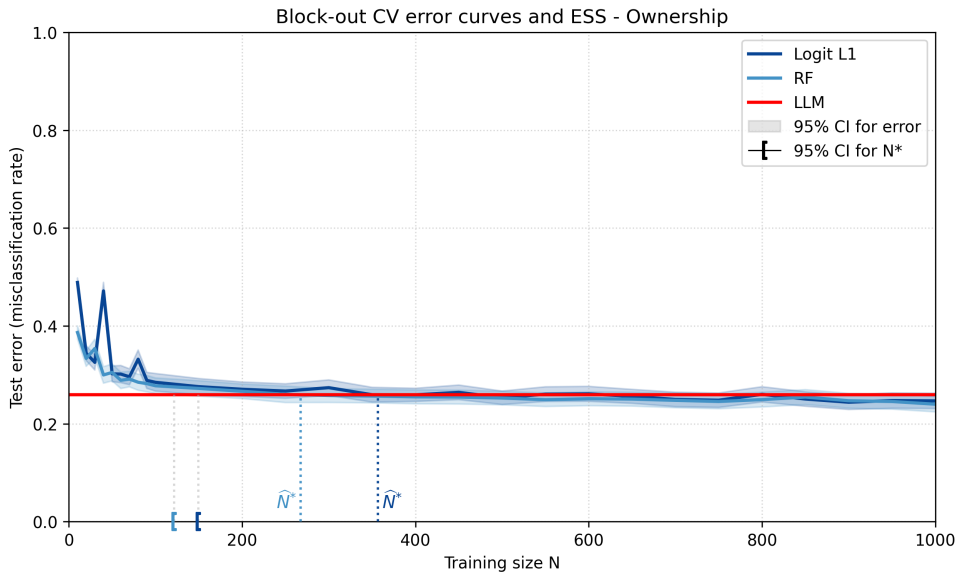


Figure 10: Misclassification rate across training sample sizes (Y: Homeownership)

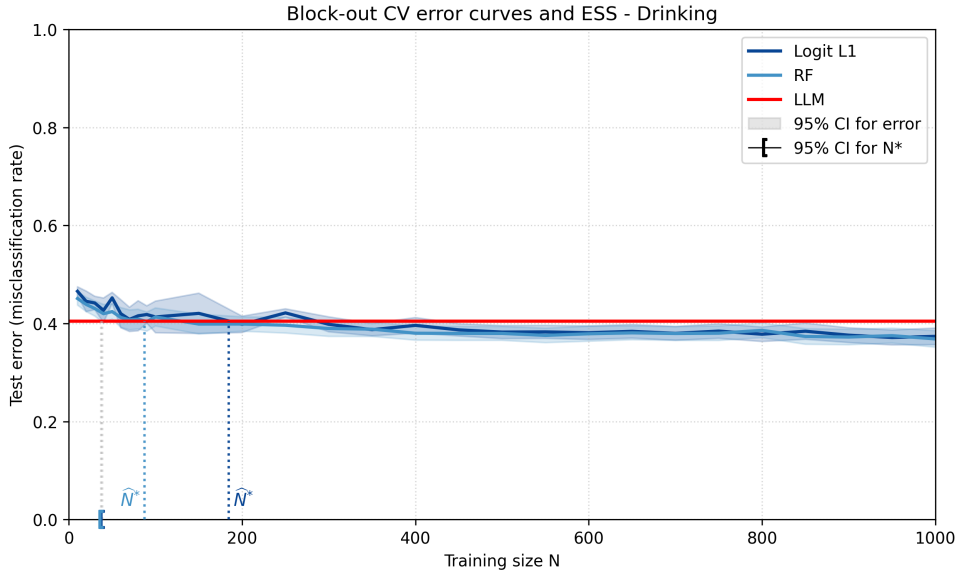


Figure 11: Misclassification rate across training sample sizes (Y: Drinking)

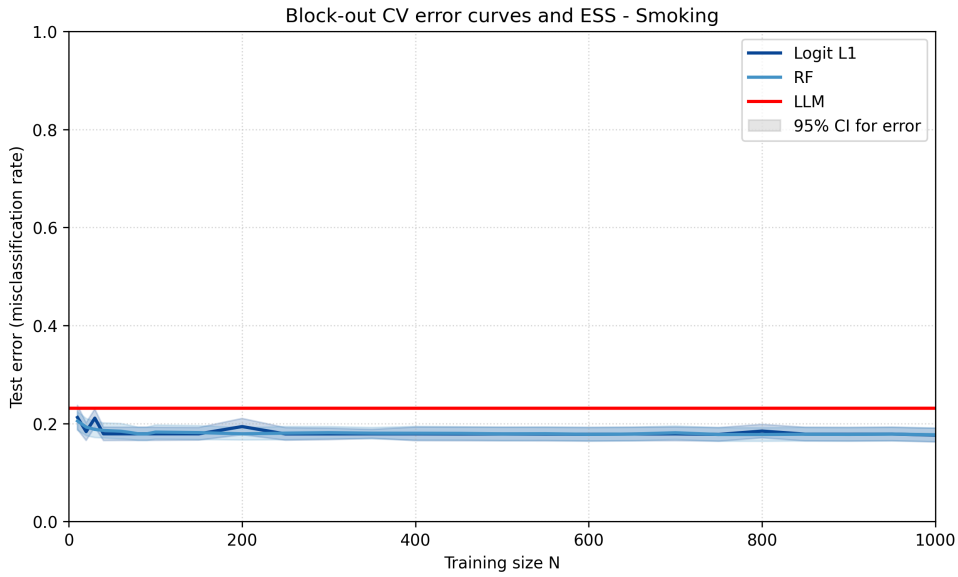


Figure 12: Misclassification rate across training sample sizes (Y: Smoking)

mask differences between the LLMs and ML algorithms that would otherwise emerge. As a robustness check, we thus compare performance in terms of *balanced accuracy* and *F1*, defined as follows.

For discrete (binary) outcomes, let $Y \in \{0, 1\}$ denote the true label and $f(X) \in \{0, 1\}$ the predicted label. Writing TP, TN, FP, and FN for the usual entries of the confusion matrix, the *balanced accuracy* is defined as the average of the class-wise recall,

$$\mathrm{BA}(f) = \tfrac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right).$$

The *F1 score* is defined as the harmonic mean of precision and recall,

$$\mathrm{F1}(f) = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} = \frac{2\,\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}},$$

and emphasizes performance on the positive class by jointly penalizing false positives and false negatives. Figures 13 and 14 replicate our classification results in Appendix E.1 using these error metrics. We find that the performances across domains are qualitatively similar to those in Figure 8.

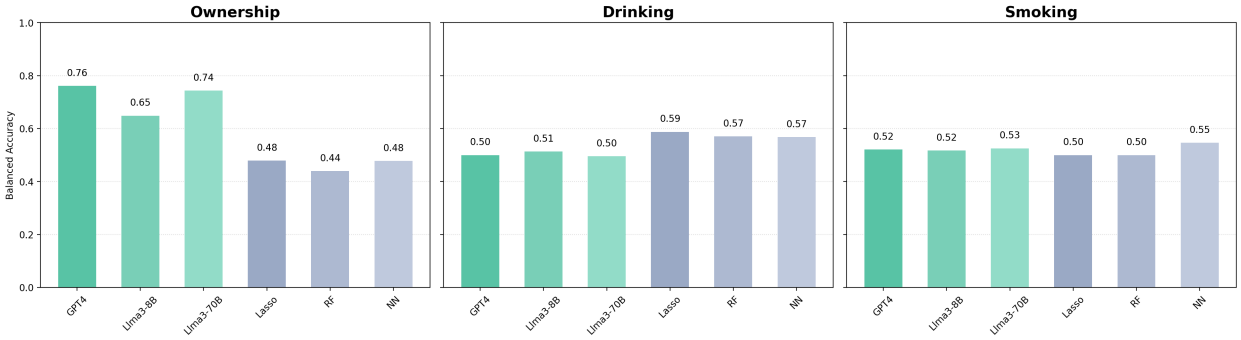


Figure 13: Balanced Accuracy (higher is better) ($N = 800$)

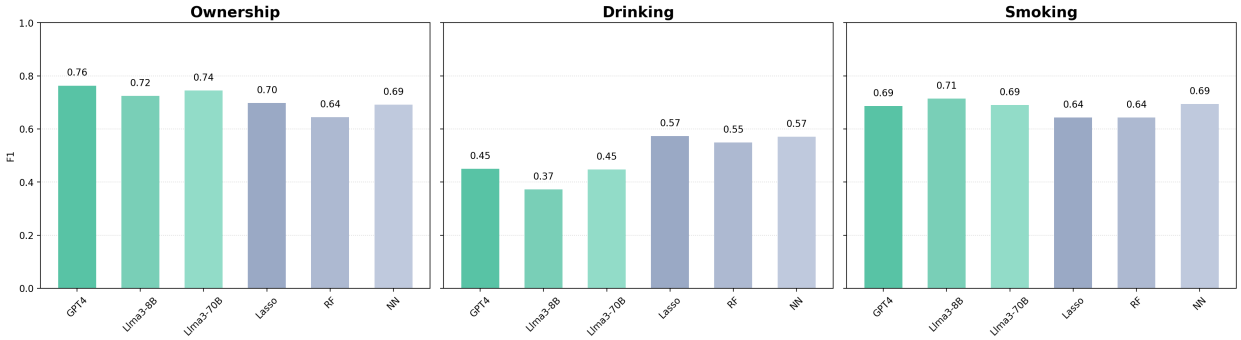


Figure 14: F1 (higher is better) ($N = 800$)