COLLABEVAL: STATISTICALLY EFFICIENT COLLABORATIVE EVALUATION VIA MATRIX COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating generative AI models is a routine, but resource-intensive, process that is conducted over and over again during the course of model development. In this work, we propose Collaborative Evaluation (CollabEval), a simple, effective, and principled method for exploiting dependencies between historical runs of different models on the same tasks to improve statistical efficiency. Specifically, our approach treats model evaluation as a matrix completion problem over an $M \times N$ matrix of evaluation scores, where M is the total number of models and N is the total number of evaluation prompts. We assume that a subset of these M models are targeted for evaluation. For these target models only a small fraction, p, of prompts has been annotated with evaluation scores. Leveraging recent results in prediction-powered statistical inference, we build a low-rank approximation of the score matrix, and use the reconstructed values as control variates in a manner that guarantees unbiased estimates of the true evaluation metric mean, in addition to statistically valid confidence intervals. Empirically, across a wide range of datasets, models, and sparsity levels p, we find that CollabEval substantially reduces the mean confidence interval size, and the mean squared error of the point estimate, compared to baseline methods at the same annotation budget.

1 INTRODUCTION

Evaluating large-scale generative AI models typically involves gathering prompts from an input distribution, generating model outputs, and scoring all outputs with a reliable rater (e.g., a verifiable reward, LLM-as-a-judge autorater, or human annotator). Though this is straightforward in principle, obtaining reliable results requires gathering a non-trivial number of data points, which can make evaluation costly. While recent work has explored the viability of using more efficient automatic metrics and autoraters as proxies for expensive human annotations, even the *inference* step for most large AI models carries substantial overhead (e.g., particularly for agents involving long, multi-step reasoning traces with tool calls), motivating methods that bypass generation entirely.

Evaluation costs compound when many evaluations are conducted simultaneously or continuously over long periods of time—for instance, when comparing fine-tuned model variants, or when monitoring for regressions against new checkpoints. The systems under comparison often share architectures, training data, or system prompts, so their performances are correlated. Standard methods, however, usually treat each of these evaluations as a distinct estimation problem, and fail to leverage this correlational structure. This leads to a suboptimal use of the available evaluation budget.

In this work, we propose *Collaborative Evaluation* (CollabEval): a lightweight approach that exploits dependencies between model output scores to increase the *effective sample size* of an evaluation, thereby *reducing the set of evaluated prompts* while preserving accuracy and confidence. Unlike methods that reduce *labeling costs* via cheaper autoraters but still incur *generation costs* (Boyeau et al., 2024; Fisch et al., 2024; Angelopoulos et al., 2025), skipped prompts in our setting incur nearly *zero cost*—requiring neither inference nor scoring. Our method has two parts. First, we reframe evaluation with partially observed scores as a collaborative filtering task: we represent the full set of models $\times$ prompts $\rightarrow$ scores as an $M \times N$ matrix S, where only a fraction p of entries are observed for the target models we want to evaluate (a subset of the M total models; other non-target models may be historical). We use matrix completion to try to impute the unobserved scores, transferring knowledge across related evaluations. Then we use these imputed scores as control variates to reduce

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

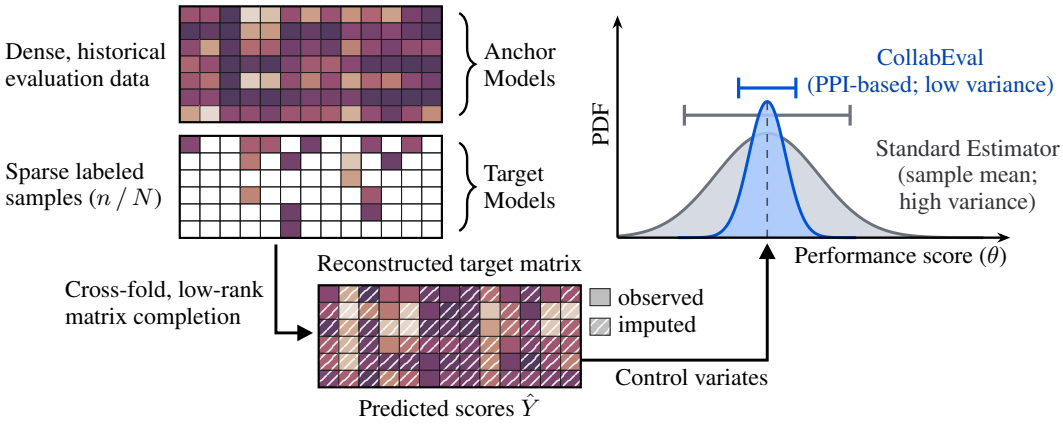

Figure 1: An illustration of the core concepts underlying CollabEval. CollabEval takes in an $M \times N$ matrix of evaluation scores, divided into dense historical data from "Anchor Models" and sparse labeled samples from new "Target Models". By applying a type of cross-fold matrix completion (Section 4.3), we reconstruct the matrix to predict missing scores. These predictions are then used as control variates within a framework based on cross-prediction-powered inference (CrossPPI; see Zrnic & Candès (2024)) to obtain more accurate estimates with tighter, valid confidence intervals.

variance, while guaranteeing unbiased estimates with asymptotically valid confidence intervals that are simple to compute. Importantly, the theoretical validity of these intervals relies entirely on the control variate framework; CollabEval maintains valid coverage even if the score matrix lacks low-rank structure and the attempted imputation is noisy.

We develop CollabEval by combining collaborative filtering techniques with prediction-powered inference (Angelopoulos et al., 2023a)—specifically cross-prediction-powered inference (Zrnic & Candès, 2024)—and introduce technical extensions for rigorously estimating both *pointwise* metrics (e.g., the absolute accuracy of model A) and *pairwise* differences (e.g., the accuracy delta between models A and B). Notably, unlike prior work such as item response theory (Section 2), our framework places few restrictions on the evaluation metric, the matrix completion algorithm, or the composition or size of the *anchor model* set (the models for which we have dense annotations). Furthermore, when combined with power tuning (Angelopoulos et al., 2023b), CollabEval is guaranteed to have an asymptotic variance less than or equal to classical estimation (Proposition A.9). This variance reduction scales with the squared covariance between the true target model scores and the imputed predictions from matrix completion; if imputed values are uncorrelated with true values, the estimator gracefully falls back to classical estimation. The magnitude of the variance reduction in realistic practical scenarios is an empirical question. To that end, we evaluate CollabEval on five diverse text generation benchmarks across a variety of real-world experimental settings, and show that it can **reduce confidence interval widths by up to 30%** compared to classical inference.

Summary of main contributions

Model evaluation suffers from high variance when the sample size is small. However, different models are often correlated. CollabEval mitigates evaluation variance by reframing model evaluation as a matrix completion problem—using results from other, historical, models to impute scores for skipped evaluation prompts from a (potentially) much larger evaluation set. We derive control variates from these predictions via cross-prediction-powered inference (Zrnic & Candès, 2024), yielding lower mean squared evaluation error and tighter confidence intervals.

2 Related work

CollabEval builds on recent work in prediction-powered inference (Angelopoulos et al., 2023a;b; Zrnic & Candès, 2024), which can reduce the cost of model evaluation by combining cheap-and-noisy "weak" labelers, such as LLM-as-judge, with more expensive and accurate annotations, such as human ratings (Boyeau et al., 2024; Gligorić et al., 2024; Fisch et al., 2024; Angelopoulos et al., 2025). This

line of work has focused on reducing the cost of *labeling* model outputs, but in all cases still requires paying the computational price to *generate* the outputs. Furthermore, at least in framing, these approaches typically require an explicit, standalone label predictor. Instead, the key insight in our work is to use matrix completion to produce noisy annotations without the cost of generation—and effectively use the historical scores of other models as the only "features" needed to predict a target model's performance (though we note that additional covariates for each prompt / model pair could also be readily incorporated into our framework as part of a *contextual* matrix completion algorithm). On a technical level, CollabEval uses cross-prediction-powered inference (Zrnic & Candès, 2024) to do the statistical inference, with some minor modifications to adapt the underlying theory for the matrix case, and pairwise comparisons with power tuning. The augmented inverse propensity weighting (AIPW) estimator of Zhang et al. (2025) applies the same idea with a logistic regression on other models' scores, though without cross-fitting (which is needed for valid confidence intervals) or power tuning.

The use of matrix reconstruction is motivated by growing evidence that model evaluation scores exhibit strong low-rank structure—not only across benchmarks, where a handful of evaluations can predict the rest (Ye et al., 2023; Zhang et al., 2024; Papailiopoulos, 2026), but also at the *item level*, where individual prompt scores are highly correlated across models (Zhou et al., 2025). CollabEval exploits this finer-grained redundancy to reduce the number of items evaluated *within* each benchmark, while still providing rigorous statistical guarantees—and puts this general collaborative filtering style of approach to evaluation on solid footing. Prior work on evaluation has also leveraged such cross-model correlations, specifically through Item Response Theory (IRT; Martínez-Plumed et al., 2019; Rodriguez et al., 2021; Maia Polo et al., 2024; Kipnis et al., 2025; Hofmann et al., 2025; Maia Polo et al., 2026), which factors a matrix of items and scores to estimate an "ability" parameter per model. CollabEval is based on the same intuitions, but rather than estimating a logit-space ability parameter, it enables accurate and precise estimates of any evaluation characteristic of interest. Importantly, matrix completion can be seen as a generalization of IRT, at least in terms of predictive power: while a valid IRT model implies a very low rank score (log odds) matrix, the converse does not necessarily hold, allowing our framework to capture model-item interactions that IRT might miss. Of course, IRT models can also be used as drop-in substitutes for the matrix completion algorithm in CollabEval (and we report results for 2PL IRT models in our experiments for binary tasks).

Finally, we note that another class of related work is based on coresets (see, e.g., Anchor Points; Vivek et al., 2024), in which evaluation data is downsampled to a few key datapoints. In contrast, CollabEval leverages the unlabeled data rather than throwing it away—and furthermore does so in a manner that yields *unbiased* estimates and *statistically valid confidence intervals* without strong assumptions about the underlying data. Empirically, CollabEval also exhibits greater stability across tasks.

3 Background

Collaborative Evaluation is built upon the statistical framework of prediction-powered inference (PPI; Angelopoulos et al., 2023a;b). PPI is designed to efficiently estimate population parameters (in our case, the mean model performance, $\theta = \mathbb{E}[Y]$) in settings where acquiring the true labels Y is expensive or difficult. The standard approach of calculating a sample mean from a small set of n annotated data points is unbiased, but often suffers from high variance, yielding confidence intervals that are too wide for meaningful decision-making. PPI provides a principled way to reduce this variance by leveraging a much larger, cheaply-obtained set of $N + n$ "predicted" labels $\hat{Y}$ on both the original labeled set of n examples, and an additional set of N unlabeled examples—though these predictions may be systematically biased. PPI then combines the high-cost, unbiased data with the low-cost, biased data using a "rectified" mean, as shown in Equation (1).

$$\hat{\theta}_{\text{ppi}} = \frac{1}{N}\sum_{i=1}^{N}\hat{Y}_i \; + \; \frac{1}{n}\sum_{i=N+1}^{N+n} Y_i - \hat{Y}_i, \tag{1}$$

where the random variables $\{\hat{Y}_i\}_{i=1}^{N+n}$ and $\{Y_i\}_{i=N+1}^{N+n}$ are respectively assumed to be independent and identically distributed (iid). Notice that this construction then has two useful properties. First, the PPI estimator, $\hat{\theta}_{\text{ppi}}$, is unbiased (i.e., $\mathbb{E}[\hat{\theta}_{\text{ppi}}] = \theta$), regardless of the accuracy of $\hat{Y}$ with respect to the true Y. In expectation, the rectifier term cancels out the bias of the predicted mean. Second, when $N \gg n$, the variance of $\hat{\theta}_{\text{ppi}}$ no longer primarily depends on the variance of the true scores, $\text{Var}(Y)$, but rather on

the variance of the prediction *residuals*, $\text{Var}(Y - \hat{Y})$. If the predicted scores $\hat{Y}$ are strongly correlated with the true scores Y, the residual variance will be small. This will then result in a more precise estimate of θ, and significantly tighter confidence intervals, for the same small labeling budget n.

Why bias correction is important. To take a moment to understand why the rectification step in PPI is important, consider the danger of a naive combined estimator. If one were to simply use the predictions $\hat{Y}$ as drop-in replacements for the unobserved labels, the resulting estimator

$$\hat{\theta}_{naive} = \frac{1}{N+n}\left(\sum_{i=1}^{N} \hat{Y}_i + \sum_{i=N+1}^{N+n} Y_i\right)$$

could dramatically reduce variance. However, this comes at the cost of introducing a potentially severe bias whenever the predictor systematically under- or over-estimates the true performance, and cause the mean squared error to be dominated by $(\frac{N}{N+n})^2(\mathbb{E}[\hat{Y}]-\mathbb{E}[Y])^2$. PPI sidesteps this by instead using the predictions as a control variate (Ripley, 1987) to drive down the variance, without incurring bias.

3.1 Cross-prediction-powered inference

Conventional PPI relies on the availability of an independent, pre-existing model to generate the predicted values $\hat{Y}$. If this is not the case, the simplest thing to do is to split the n labeled data points into a proper training and test set, where the training set is used to train a predictor that yields $\hat{Y}$ (e.g., given features X for label Y), and the test set is used for estimating θ per Equation (1). This is not, however, the most *efficient* way to use the labeled data points, when n is small.

Cross-prediction-powered inference (CrossPPI; Zrnic & Candès, 2024) adapts PPI to use the entire labeled dataset *both* for training the predictor that gives $\hat{Y}$ *and* for estimating θ. CrossPPI achieves this through a variation of cross-validation. First, the labeled data points are randomly partitioned into K disjoint, exchangeable folds, $\{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$. Let $\hat{Y}^{(k)}$ denote the predictions generated from a predictor that is trained on all folds $j \neq k$. These "cross-predictions" are then aggregated across all folds:

$$\hat{\theta}_{\text{crossppi}} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K} \hat{Y}_i^{(k)} + \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in\mathcal{I}_k} Y_i - \hat{Y}_i^{(k)}.$$

Under mild assumptions, Zrnic & Candès (2024) show that as n and N grow, $\hat{\theta}_{\text{crossppi}}$ is unbiased, low variance, and normally distributed: a fact useful for deriving confidence intervals.

In the next section, we adapt CrossPPI to matrix completion using an $M \times N$ matrix of partially observed scores for M models across N inputs. This yields statistically efficient estimates with asymptotically valid confidence intervals for both individual model means and pairwise differences.

4 Collaborative Model Evaluation via Matrix Completion

The main practical challenge of PPI-based methods, and *the core problem that CollabEval focuses on addressing*, is how to efficiently acquire a high-quality set of predictions $\hat{Y}$ in the first place. At a high-level, CollabEval is then fairly simple: we use matrix completion to impute missing scores, and then correct for the bias that the noisy predictions may have. The following sections carefully setup this up. In Section 4.2 we formally define the setting. Then, in Section 4.3, we describe the specifics of our cross-fold matrix completion algorithm, and resulting estimator. Finally, in Section 4.4 we derive confidence intervals with rigorous statistical guarantees (however, we defer the formal statements and proofs of the guarantees to Appendices A and E). See also Appendix D for implementation details.

4.1 A motivating example

To start, consider a scenario where we want to evaluate a target model T across a diverse set of prompts, while leveraging historical evaluations from two anchor models, A_1 and A_2. We can assume that T's performance correlates differently with that of A_1 and A_2, depending on the type of input. For example, T might be comparable to A_1 on "Type 1" prompts (e.g., math reasoning), but then more like A_2 on "Type 2" prompts (e.g., creative writing). Relying on either A_1 or A_2 alone as a

global control variate would be suboptimal. Ideally, an oracle control variate would dynamically route predictions (i.e., use A_1 for Type 1 prompts and A_2 for Type 2 prompts).

The general motivation for using matrix completion techniques here is to naturally approximate this sort of "control variate routing" behavior, or potentially something better. For example, in the low-rank matrix factorization approach we take (see Section 4.3), observing just a few scores from model T allows us to project it into a shared latent space alongside the other anchor models and the prompts. The predicted score $\hat{Y}$ for model T for any unobserved prompt is a linear combination of the anchor models' scores on that same prompt, where the weighting depends on the learned latent factors.

4.2 Setting

Let $S \in \mathbb{R}^{M \times N}$ be a matrix of evaluation scores, where M is the number of models, N is the total number of input prompts, and S_{ij} is the score of model i's response to prompt j. Assuming the N prompts are sampled i.i.d. from a distribution of interest, each S_{ij} is a random variable. We use S_i to represent the marginal score distribution for model i, meaning $S_i \overset{d}{=} S_{ij}$ for all $j \in \{1, \ldots, N\}$. We then partition the models into two disjoint sets: *anchor models* $\mathcal{A} \subset \{1, \ldots, M\}$, for which we have historical data (having evaluated many entries in an anchor row $i \in \mathcal{A}$), and *target models* $\mathcal{T} \subseteq \{1, \ldots, M\}$, which are new models with initially unobserved scores.[1]

For each target model i, our goal is to estimate $\theta_i = \mathbb{E}[S_i]$. Since model evaluations are ultimately often used to do model comparisons, we also estimate the mean performance *difference* between the target model i and any other model $j \neq i$, defined as $\Delta_{ij} = \mathbb{E}[S_i - S_j]$. The comparison model j can be either a target or anchor model. To reduce the cost of evaluating all N prompts for the target models, we observe only a small subset of n_i independently selected indices from $\{1, \ldots, N\}$ for each target model $i \in \mathcal{T}$.[2] We use $\Omega \in \{0, 1\}^{M \times N}$ to denote the binary matrix of observed indices:

$$\Omega_{ij} = \begin{cases} 1 & \text{if model } i\text{'s evaluation score for prompt } j \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases}$$

We assume that Ω is independent from the prompts. For the entries (i, j) where $\Omega_{ij} = 0$, S_{ij} is imputed via a procedure that we call *cross-fold matrix completion*, which we describe next.

4.3 Cross-fold matrix completion

Let $\mathcal{M} \colon \mathbb{R}^{M \times N} \times \{0, 1\}^{M \times N} \to \mathbb{R}^{M \times N}$ be a matrix completion algorithm that accepts a partially observed matrix S and a binary mask Ω, and returns a matrix $\hat{S}$ where missing entries (i, j) for which $\Omega_{ij} = 0$ have been imputed. We use an iterative algorithm based on SVD (see, e.g., Troyanskaya et al., 2001; Mazumder et al., 2010) that recursively updates missing entries in the matrix S by using low-rank approximations for $\hat{S}$ (see Appendix D for details). It is also worth noting that the computational overhead of this matrix completion is small for the sizes of evaluation data matrices that we consider, even when repeated multiple times. As the imputed scores in $\hat{S}$ depend on the observed scores in the original S, we use a K-fold cross-completion strategy based on CrossPPI in Section 3.1. This will allow us to estimate the bias in our imputed evaluation scores, and correct for it.

First, we identify the set of active prompt indices $\mathcal{U} = \bigcup_{i \in \mathcal{T}} \mathcal{J}_i$, which is the set of prompts for which at least one target model has an observation. We randomly partition $\mathcal{U}$ into K disjoint folds, $\{\mathcal{U}_1, \ldots, \mathcal{U}_K\}$. For each target model i, this splits its observed indices $\mathcal{J}_i$ into K folds, $\{\mathcal{J}_i^{(1)}, \ldots, \mathcal{J}_i^{(K)}\}$, where $\mathcal{J}_i^{(k)} = \mathcal{J}_i \cap \mathcal{U}_k$. We then construct K cross-prediction masks, $\{\Omega^{(1)}, \ldots, \Omega^{(K)}\}$, where each mask $\Omega^{(k)}$ holds out the k-th fold of labels across all target models.

Applying the completion algorithm to each mask yields a corresponding set of prediction matrices, $\hat{S}^{(k)} = \mathcal{M}(S, \Omega^{(k)})$. We then aggregate these matrices into a single prediction matrix $\hat{Y}$ using the

[1] In a real-world evaluation pipeline, we might assume that the anchor model set is periodically refreshed.

[2] We slightly abuse the prior notation from Section 3: $n_i \leq N$ now denotes a *subset* of the full dataset.

held-out prediction for observed entries and the cross-fold average for unobserved entries:

$$\hat{Y}_{ij} = \begin{cases} \hat{S}_{ij}^{(k)} & \text{if } j \in \mathcal{J}_i^{(k)} \text{ and } i \in \mathcal{T} \quad \text{(cross-fold prediction independent of } S_{ij}\text{)} \\ \dfrac{1}{K}\displaystyle\sum_{k=1}^{K} \hat{S}_{ij}^{(k)} & \text{otherwise} \quad \text{(average cross-fold prediction)} \end{cases} \tag{2}$$

Finally, we estimate the performance $\hat{\theta}_i$ using a slightly rewritten form of CrossPPI,

$$\hat{\theta}_i = \frac{1}{N}\sum_{j=1}^{N} \hat{Y}_{ij} + \frac{1}{|\mathcal{J}_i|}\sum_{j \in \mathcal{J}_i} (S_{ij} - \hat{Y}_{ij}) = \frac{1}{|\mathcal{J}_i|}\sum_{j \in \mathcal{J}_i} S_{ij} - \left(\frac{1}{|\mathcal{J}_i|}\sum_{j \in \mathcal{J}_i} \hat{Y}_{ij} - \frac{1}{N}\sum_{j=1}^{N} \hat{Y}_{ij} \right), \tag{3}$$

and the difference in performance between models i and j as $\hat{\Delta}_{ij} = \hat{\theta}_i - \hat{\theta}_j$.

Note that Equation (3) calculates the mean prediction using all N items (versus only the $N - n$ unlabeled items where $\Omega_{ij} = 0$ as standard PPI would do), and rewrites the estimator in a classical control variate form (Ripley, 1987). We use this form for convenience (it can be shown that the two versions have equivalent asymptotic variance when $\hat{Y}_{ij}$ is scaled by an optimal constant of choice).

4.4 Confidence interval estimation

Let $n_i = |\mathcal{J}_i|$ be the number of observed labels for model i, and $n_{ij} = |\mathcal{J}_i \cap \mathcal{J}_j|$ be the number of overlapping observed labels between models i and j. We then estimate the covariance matrix $\hat{\Sigma}$ for the joint vector of CollabEval predictions $\sqrt{N}\hat{\theta} \in \mathbb{R}^M$ (which, in Appendix A, we show converges to the true asymptotic covariance matrix Σ under mild assumptions) using:

$$\hat{\Sigma}_{ij} = \frac{N n_{ij}}{n_i n_j}\widehat{\mathrm{Cov}}(S_i, S_j) + \left(\frac{N n_{ij}}{n_i n_j} - 1\right)\left[\widehat{\mathrm{Cov}}(\hat{Y}_i, \hat{Y}_j) - \widehat{\mathrm{Cov}}(\hat{Y}_i, S_j) - \widehat{\mathrm{Cov}}(S_i, \hat{Y}_j)\right],$$

where the empirical covariances $\widehat{\mathrm{Cov}}(\cdot, \cdot)$ involving true scores S are calculated over the intersecting observed sets (e.g., $j \in \mathcal{J}_i \cap \mathcal{J}_j$), and the empirical covariances of the predicted scores $\hat{Y}$ are calculated over the all N prompts. We can then construct confidence intervals with $(1 - \alpha)$ coverage for both the individual model means and the pairwise differences as follows:

$$\mathcal{C}_\alpha(\hat{\theta}_i) = \left[\hat{\theta}_i \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\Sigma}_{ii}}{N}}\right] \quad \text{and} \quad \mathcal{C}_\alpha(\hat{\Delta}_{ij}) = \left[\hat{\Delta}_{ij} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\Sigma}_{ii} + \hat{\Sigma}_{jj} - 2\hat{\Sigma}_{ij}}{N}}\right],$$

where $z_{1-\alpha/2}$ is the standard normal quantile. Crucially, as long as the matrix completion algorithm satisfies a basic stability condition (i.e., its predictions converge as data grows), these intervals are statistically valid. We summarize this in the following informal theorem.

Theorem 1 (CI Coverage of CollabEval: Informal)**.** *Under mild stability assumptions on the completion algorithm $\mathcal{M}$, as the total number of prompts N and the labeled subsets n_i grow large, the estimators $\hat{\theta}_i$ and $\hat{\Delta}_{ij}$ are asymptotically normally distributed. Consequently, the confidence intervals $\mathcal{C}_\alpha(\hat{\theta}_i)$ and $\mathcal{C}_\alpha(\hat{\Delta}_{ij})$ converge to valid $(1 - \alpha)$ marginal coverage of the true parameters, θ_i and Δ_{ij}.*

A formal statement of Theorem 1 and related results are provided in Appendix A. We note that the empirical covariance estimates above can also be used to compute optimal "power tuning" weights (Angelopoulos et al., 2023b), a technique that scales $\hat{Y}_{ij}$ by a scalar $\lambda_i \in \mathbb{R}$ to ensure that the asymptotic variance is never worse than that of the classical mean. We provide a derivation of the optimal weights $\lambda^* \in \mathbb{R}^M$ to use for estimating both means and differences in means (they can differ) in Appendix A.2. In the remainder of the paper, we use these scaled estimators by default.

5 Experiments

Datasets. The datasets we use cover a diverse range of settings and metrics. Specifically, we assess instruction-following capabilities using **AlpacaEval 2.0** (Li et al., 2023), multiple-choice multitask

language understanding via **MMLU** (Hendrycks et al., 2020), grounded generation and evidence-supported correctness with **Attributed Question Answering (AQA)** (Bohnet et al., 2023), machine translation across 55 languages using **WMT24++** (Deutsch et al., 2025), and repository-level patch generation via **SWE-bench** (Jimenez et al., 2024).[3] For AlpacaEval 2.0 and MMLU, we designate specific target models based on recent leaderboard submissions while using older or remaining models as anchors; conversely, for AQA, WMT24++, and SWE-bench, we dynamically construct our target and anchor sets using leave-one-out or leave-two-out evaluation protocols over the available systems. See Appendix B for additional details. In the main text, we report performance metrics averaged across all datasets, with error bars denoting the **min** and **max** results. Individual task results are in Appendix C.

Sampling strategies. We consider two straightforward strategies for sampling observations:

1. **Independent sampling (IID):** For each target model, the observed indices are drawn uniformly at random. This can be helpful for the matrix completion stage: observing a more diverse set of prompts may lead to better estimates of the underlying low-rank structure, especially in settings where the set of anchor models is small, or does not span the subspace of the new target models.
2. **Paired sampling (Paired):** Here we observe the exact same subset of prompts for each target model ($\mathcal{J}_i = \mathcal{J}_j$). This can be very advantageous when estimating Δ_{ij}, the difference in model performance, if S_i and S_j are correlated in ways that the predictions $\hat{Y}_i$ and $\hat{Y}_j$ do not fully capture. While this is good for variance reduction, it does limit the diversity of the observed data in the matrix completion, and makes the anchor models critical for being able to reconstruct the target rows.

Baselines. We compare CollabEval against the **classic sample mean** baseline, which computes the empirical mean over the observed samples. We apply this baseline using both the independent (IID) and paired sampling strategies described in Section 5. We then also consider three baselines (the results of which are deferred to Appendix C, along with several other additional experimental results):

- **Naive Imputation**: Here we directly construct a dense evaluation matrix by imputing all missing entries with one step of matrix completion. We then naively calculate the sample mean and variance without any corrections. While this aggressively reduces variance, it is vulnerable to systematic bias, and, in general, will not produce valid confidence intervals.
- **Anchor Points (Vivek et al., 2024)**: A coreset-based approach that selects a small subset of prompts to evaluate based on historical anchor model scores. We compute the mean performance scores for new models over these selected points only, and skip the rest. This yields a point estimate, but does not natively provide valid confidence intervals (and like naive imputation, can also be biased).
- **PPI (Anchor Mean)**: A baseline that replaces our matrix completion step with a simple, untrained control variate. Specifically, we compute the mean score of all available anchor models for prompt j and use it as the drop-in prediction for any target model's score (i.e., it is not model specific).

Appendix D.2 also compares CollabEval across four alternative, drop-in, matrix completion algorithms: IterativeSVD (default), a neural network, nuclear norm minimization, and a 2-PL IRT model.

5.1 Metrics

We compute all metrics over $10k$ bootstrap trials, where we resample N base examples from each dataset with replacement, and set the target ground-truth parameters θ and Δ to be the dataset average.

- **CI Coverage:** The empirical probability that the nominal $(1-\alpha)$ confidence interval produced by a given method contains the true full-sample parameter (θ_i or Δ_{ij}).
- **% Reduction in CI Size (or MSE):** The relative decrease in average confidence interval width achieved by a given method compared to the classic independent sampling baseline. When comparing methods that do not produce valid confidence intervals (e.g., the naive imputation estimator), we instead report the % reduction in mean squared error (MSE).
- **Effective Sample Fraction:** The labeling budget the classic independent baseline would require to achieve the exact same MSE as a given method evaluated at fraction p. This is plotted as a fraction of the total dataset N; values strictly greater than p indicate that the evaluated method yields the statistical power of a larger independent labeled sample.

[3] In Appendix C.1 we show that the evaluation matrices for these tasks indeed exhibit low effective rank.

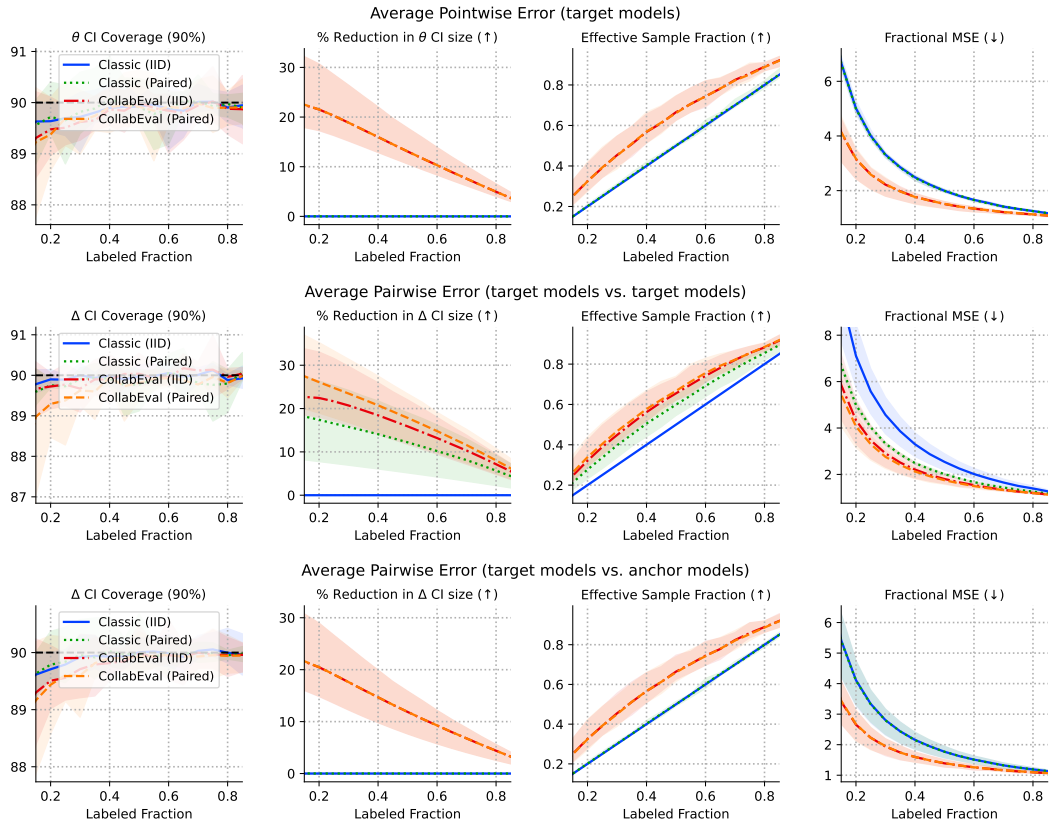

Figure 2: **Performance metrics averaged across all five datasets.** Rows show pointwise estimation of target models ($\hat{\theta}_i$, top), pairwise differences between target models ($\hat{\Delta}_{ij}$ for $i, j \in \mathcal{T}$, middle), and pairwise differences between target and anchor models ($\hat{\Delta}_{ij}$ for $i \in \mathcal{T}, j \in \mathcal{A}$, bottom). CollabEval consistently improves over classical baselines (and the additional baselines in Appendix C.2). IID and paired sampling perform equivalently for the top and bottom rows, but for target-vs-target comparisons (middle) paired sampling with CollabEval achieves the highest efficiency, particularly for small p.

- **Fractional MSE:** The MSE of a given method at fraction p, divided by the MSE at $p = 1$ (recall that we evaluate MSE using bootstrap resampling, so the MSE at $p = 1$ is > 0). This quantifies how much of the precision of the full evaluation we can recover with a smaller budget. Lower values are better, with 1 representing parity with the full size evaluation.

5.2 Results

Confidence interval coverage. CollabEval consistently approaches a coverage rate of 90%. This empirically validates our theoretical results that were used to derive the confidence intervals in Section 4. While our guarantees are asymptotic in the total number of labels, we note that the coverage empirically quickly approaches the desired coverage rate even in lower sample settings ($p < 0.2$). In contrast, the naive imputation baseline given in Appendix C.2 fails to maintain valid coverage.

Estimation efficiency (CI Size and MSE). CollabEval demonstrates a significant improvement in statistical efficiency compared to the classical empirical estimators. For individual mean estimation (θ_i), CollabEval yields substantially tighter confidence intervals than the baseline, showing an average reduction in CI width of over 20% when the labeled fraction p is small. This effect is even more pronounced on specific datasets; for example, in the AQA evaluation (see Appendix C.3), we observe a reduction in CI width of over 30% at lower fractions. Similar trends are observed for pairwise differences (Δ_{ij}), where CollabEval reduces the average CI width by over 30% in the target-vs-target setting, driven by particularly strong performance on the WMT and AQA benchmarks (Appendix C.3). Furthermore, the "Effective Sample Fraction" curves (Figure 2, third column) consistently stay above the $y = x$ parity line, indicating that CollabEval requires fewer ground-truth labels to achieve the same

MSE as the baseline. For instance, on average across datasets, when the classic estimator requires 60% of the labeled data to achieve a certain precision for θ, CollabEval can achieve the same precision with approximately 45% of the labeled data. Finally, our additional results in Appendix C.2 show that CollabEval is significantly better at reducing the MSE than both the naive estimator (which can even sometimes *hurt MSE*) and the PPI (Anchor Mean) baseline—highlighting both the importance of explicit bias correction, as well as the use of the cross-fold matrix completion approach over an untrained alternative. Similarly, Appendix C.2 shows that CollabEval consistently yields lower MSE than the Anchor Points baseline, and exhibits greater empirical stability across tasks and labeling fractions.

Effect of the labeled fraction size. The gains of CollabEval are generally most pronounced when the labeled fraction p is small. As seen in the "% Reduction in CI Size" plots, the relative advantage of CollabEval peaks when the labeled data is scarce, but in sufficient quantity to learn from (e.g., $p \approx 0.1 - 0.2$), and gradually diminishes as p grows. This behavior is expected: as the labeled set becomes larger, two things happen: (1) the variance of the classical estimator naturally decreases, and (2) the relative amount of extra unlabeled data CollabEval can leverage diminishes. However, even at moderate labeled fractions (e.g., $p = 0.5$), CollabEval often retains a substantial efficiency advantage—reducing the average CI width by $\sim$10-20% depending on the specific estimand, and $> 20\%$ on tasks like AQA and WMT (Appendix C.3). As p gets closer to 1.0, all methods converge.

Effects of the sampling strategy. For individual means (θ_i), paired and independent sampling perform virtually identically across our experiments, as seen in the top row of Figure 2. However, for difference estimates between target models only (Figure 2, middle row), the benefits of paired sampling are pronounced. As evaluation scores on the same prompts are often correlated across models, the classical paired estimator is competitive, and outperforms the classical IID estimator (and suggests that paired sampling should generally be the natural approach for these types of comparative evaluation tasks). Building upon this, CollabEval with paired sampling further improves upon the classical paired estimator, yielding the tightest confidence intervals and lowest error overall. Note that these effects disappear when looking at the difference estimates between target models and anchor models (Figure 2, bottom row), since here paired and IID sampling are effectively the same (as every target model observation is automatically paired with an anchor model observation).

Effects of the anchor model set. CollabEval relies on correlations between anchor and target model evaluations. In Figure 10 in Appendix C.4, we ablate the anchor set's size and composition at a fixed 50% labeled data fraction. We compare three selection strategies for an anchor set of size k: highest-scoring (Top-k), lowest-scoring (Bottom-k), and randomly selected models (Random-k). Confidence interval coverage remains stable across all sizes and strategies, consistent with our theoretical guarantees. Efficiency improves as the anchor set grows and becomes more predictive of target performance, though CollabEval maintains positive variance reduction even when the anchor model set is small (for all selection strategies). With respect to differences in anchor selection strategies, random selection (Random-k) yields the highest variance reduction, while selecting only the lowest-scoring models (Bottom-k) is the least effective strategy (though it still reduces variance relative to classical estimation). These results indicate that while the *degree* of variance reduction that can be achieved by CollabEval naturally varies depending on the anchor models, its performance is quite robust in general.

6 Conclusion

This work introduces Collaborative Evaluation (CollabEval), a practical and rigorous framework for reducing model evaluation cost. Formalizing the widely reported phenomenon in recent literature that different model outputs and evaluation scores are often highly correlated, CollabEval casts evaluation as a matrix completion problem, and uses historical data to impute scores that are then used as control variates within a cross-prediction-powered inference framework. We theoretically establish that CollabEval yields unbiased estimates with asymptotically valid confidence intervals for both individual model performance and pairwise model comparisons, regardless of the accuracy of the underlying completion (and with only mild assumptions on the algorithm used), or the data distribution. Empirically, we demonstrate across a range of benchmarks that CollabEval significantly outperforms standard estimation techniques. By effectively converting historical evaluations into increased statistical power, CollabEval substantially increases evaluation precision at nearly no extra cost to the evaluator, and serves as a simple but principled resource for practitioners tasked with the continuous evaluation and monitoring of rapidly evolving generative AI systems.

AI USE

We used generative AI to refine mathematical claims and develop proof strategies. All proofs are written and checked by the authors. AI assistance was also used to produce the code for the experiments and the figures, and to help draft or edit parts of the paper for readability. All code and text was rigorously checked, revised, refined, and validated by the authors. We used an AI automated reviewer to evaluate an early draft of the manuscript and to suggest improvements. We take complete responsibility for the final content of this work, including text, claims or artifacts produced with the aid of AI.

REPRODUCIBILITY

The empirical part of the paper is based on public datasets, which are described in Section 5 and Section B. Section D is dedicated to describing the implementation of CollabEval, including all sub-algorithms such as the iterative SVD technique used for matrix completion. Other aspects of the experiments are described in Section 5 in the main paper, and in the appendix in Section C.

REFERENCES

Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 382(6671):669–674, 2023a.

Anastasios N Angelopoulos, John C Duchi, and Tijana Zrnic. PPI++: Efficient prediction-powered inference. *arXiv preprint arXiv:2311.01453*, 2023b.

Anastasios N. Angelopoulos, Jacob Eisenstein, Jonathan Berant, Alekh Agarwal, and Adam Fisch. Cost-optimal active ai model evaluation. *arXiv preprint arXiv:2506.07949*, 2025. URL https://arxiv.org/abs/2506.07949.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint: 2212.08037*, 2023.

Pierre Boyeau, Anastasios N Angelopoulos, Nir Yosef, Jitendra Malik, and Michael I Jordan. AutoEval done right: Using synthetic data for model evaluation. *arXiv preprint arXiv:2403.07008*, 2024.

Ivi Chatzi, Eleni Straitouri, Suhas Thejaswi, and Manuel Gomez Rodriguez. Prediction-powered ranking of large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 113096–113133. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cd47cd67caa87f5b1944e00f6781598f-Paper-Conference.pdf.

Daniel Deutsch, Eleftheria Briakou, Isaac Rayburn Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. WMT24++: Expanding the language coverage of WMT24 to 55 languages & dialects. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 12257–12284, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.634. URL https://aclanthology.org/2025.findings-acl.634/.

Adam Fisch, Joshua Maynez, R. Alex Hofer, Bhuwan Dhingra, Amir Globerson, and William W. Cohen. Stratified prediction-powered inference for effective hybrid evaluation of language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8CBcdDQFDQ.

Kristina Gligorić, Tijana Zrnic, Cinoo Lee, Emmanuel J Candès, and Dan Jurafsky. Can unconfident llm annotations be used for confident conclusions? *arXiv preprint arXiv:2408.15204*, 2024.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15789–15809, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Valentin Hofmann, David Heineman, Ian Magnusson, Kyle Lo, Jesse Dodge, Maarten Sap, Pang Wei Koh, Chun Wang, Hannaneh Hajishirzi, and Noah A. Smith. Fluid language model benchmarking. In *Second Conference on Language Modeling*, 2025.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In Barry Haddow, Tom Kocmi, Philipp Koehn, and Christof Monz (eds.), *Proceedings of the Ninth Conference on Machine Translation*, pp. 492–504, Miami, Florida, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.wmt-1.35.

Alex Kipnis, Konstantinos Voudouris, Luca M. Schulze Buschoff, and Eric Schulz. metabench - a sparse benchmark of reasoning and knowledge in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4T33izzFpK.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023.

Zhengzhong Liu, Bowen Tan, Hongyi Wang, Willie Neiswanger, Tianhua Tao, Haonan Li, Fajri Koto, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. Llm360 k2: Building a 65b 360-open-source large language model from scratch. *arXiv preprint arXiv:2501.07124*, 2025.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Felipe Maia Polo, Aida Nematzadeh, Virginia Aglietti, Adam Fisch, and Isabela Albuquerque. Rich insights from cheap signals: Efficient evaluations via tensor factorization. *arXiv preprint arXiv:2603.02029*, 2026. URL https://arxiv.org/abs/2603.02029.

Fernando Martínez-Plumed, Ricardo BC Prudêncio, Adolfo Martínez-Usó, and José Hernández-Orallo. Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42, 2019.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(80):2287–2322, 2010.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Dimitris Papailiopoulos. You don't need to run every eval, 2026. URL https://github.com/anadim/llm-benchmark-matrix.

B. D. Ripley. *Stochastic simulation*. John Wiley & Sons, Inc., New York, NY, USA, 1987. ISBN 0-471-81884-4.

Pedro Rodriguez, Joe Barrow, Alexander Hoyle, John P. Lalor, Robin Jia, and Jordan Boyd-Graber. Evaluation examples are not equally informative: How should that change NLP leaderboards? In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4486–4503, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.346. URL https://aclanthology.org/2021.acl-long.346/.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In Yvette Graham and Matthew Purver (eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.95. URL https://aclanthology.org/2024.eacl-long.95/.

Qinyuan Ye, Harvey Fu, Xiang Ren, and Robin Jia. How predictable are large language model capabilities? a case study on BIG-bench. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 7493–7517, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.503. URL https://aclanthology.org/2023.findings-emnlp.503/.

Guanhua Zhang, Florian E. Dorner, and Moritz Hardt. How benchmark prediction from fewer data misses the mark. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 135638–135678. Curran Associates, Inc., 2025. doi: 10.52202/085713-4528. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c57bacb5ea621582235c54ce2a83136d-Paper-Conference.pdf.

Qiyuan Zhang, Fuyuan Lyu, Xue Liu, and Chen Ma. Collaborative performance prediction for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2576–2596, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.150. URL https://aclanthology.org/2024.emnlp-main.150/.

Jin Peng Zhou, Christian K Belardi, Ruihan Wu, Travis Zhang, Carla P Gomes, Wen Sun, and Kilian Q Weinberger. On speeding up language model evaluation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3cvwO5DBZn.

Tijana Zrnic and Emmanuel J. Candès. Cross-prediction-powered inference. *Proceedings of the National Academy of Sciences*, 121(15):e2322083121, 2024. doi: 10.1073/pnas.2322083121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2322083121.

CONTENTS

A Theoretical results **14**

A.1 Asymptotic normality 14

A.2 Power tuning 15

B Dataset details **16**

C Additional experimental results **17**

C.1 Effective rank analysis 17

C.2 Additional baselines 17

C.3 Per-task results 17

C.4 Anchor set ablation 21

D Matrix completion algorithm details and ablations **21**

D.1 IterativeSVD ablations 24

D.2 Comparison to other matrix completion options 24

E Proofs **24**

E.1 Proof of Theorem A.2 24

E.2 Proof of Corollary A.3 28

E.3 Proof of Corollary A.4 28

E.4 Proof of Theorem A.5 28

E.5 Proof of Proposition A.7 28

E.6 Proof of Proposition A.9 29

A THEORETICAL RESULTS

A.1 ASYMPTOTIC NORMALITY

The purpose of this section is to prove that the estimate $\hat{\theta}$ is asymptotically multivariate normal, which will then justify the form of confidence interval proposed in Section 4.4. To start, we require a stability condition (Assumption A.1; adapted from Zrnic & Candès (2024) to our setting) on the matrix completion algorithm $\mathcal{M}$. Informally, this requires that as we collect data, the cross-fold predictions become insensitive to their specific fold and converge to a fixed population-level imputation $\underline{\hat{S}}_{ij}$ that depends only on the observed data for item j.

Assumption A.1 (Imputation stability)**.** *Let K be the number of folds, let $n_i = |\mathcal{J}_i|$ be the number of observed scores for a target model i, and let $\underline{\hat{S}}_{ij} = h_i(X_j)$ where $X_j = \{S_{i',j} : \Omega_{i'j} = 1, i' \in \mathcal{A}\}$ be some deterministic population-limit prediction that depends exclusively on the observed anchor model scores on the j-th prompt. Furthermore, let $\hat{S}^{(1)} = \mathcal{M}(S, \Omega^{(1)})$ be the matrix completion evaluated on a random fold mask $\Omega^{(1)}$. We say that $\mathcal{M}$ is stable if as $N \to \infty$ and $n_i/N \to p_i \in (0, 1]$:*

$$\sqrt{K\mathrm{Var}\left(\hat{S}_{ij}^{(1)} - \underline{\hat{S}}_{ij} \mid \Omega^{(1)}\right)} \xrightarrow{L_1} 0,$$

where $\xrightarrow{L_1}$ denotes convergence in mean.

Intuitively, this ensures that the row entries of the cross-fold-imputed matrix $\hat{Y}$ from Equation (2) behave like asymptotically i.i.d. random variables as the matrix dimensions grow, as the cross-fold predictions converge to a single, stable predictor that is only a function of the covariates (in this case, the anchor model scores) for prompt j. In practice, this form of stability is a typical property of appropriately regularized matrix completion methods.[4]

Under this condition, we establish the following central limit theorem.

Theorem A.2 (Multivariate CLT for $\hat{\theta}$)**.** *Suppose Assumption A.1 holds, and that the random variables $(S_{ij}, \underline{\hat{S}}_{ij})$ have finite covariance. Let $n_i = |\mathcal{J}_i|$, $n_j = |\mathcal{J}_j|$, and $n_{ij} = |\mathcal{J}_i \cap \mathcal{J}_j|$, where as $N \to \infty$, $n_i/N \to p_i \in (0, 1]$, $n_j/N \to p_j \in (0, 1]$, and $n_{ij}/N \xrightarrow{p} p_{ij} \in [0, 1]$. Then as $N \to \infty$,*

$$\sqrt{N}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \Sigma), \quad \textit{where}$$

$$\Sigma_{ij} = \frac{p_{ij}}{p_i p_j}\mathrm{Cov}\left(S_i, S_j\right) + \left(\frac{p_{ij}}{p_i p_j} - 1\right)\left[\mathrm{Cov}\left(\underline{\hat{S}}_i, \underline{\hat{S}}_j\right) - \mathrm{Cov}\left(\underline{\hat{S}}_i, S_j\right) - \mathrm{Cov}\left(S_i, \underline{\hat{S}}_j\right)\right]. \quad (4)$$

As $\hat{\theta}$ is asymptotically normal, any linear transformation of $\hat{\theta}$ is also asymptotically normal. Specifically, for any $a \in \mathbb{R}^M$, we have $\sqrt{N}(a^\top \hat{\theta} - a^\top \theta) \xrightarrow{d} \mathcal{N}(0, a^\top \Sigma a)$. We can derive the limiting distributions for various metrics by choosing a appropriately: for the individual mean θ_i, we set $a = \mathbf{e}_i$ (the standard basis vector), and for the pairwise difference Δ_{ij}, we set $a = \mathbf{e}_i - \mathbf{e}_j$.

Corollary A.3 (Univariate CLTs for $\hat{\theta}_i$ and $\hat{\Delta}_{ij}$)**.** *Under the assumptions of Theorem A.2,*

$$\sqrt{N}(\hat{\theta}_i - \theta_i) \xrightarrow{d} \mathcal{N}(0, \Sigma_{ii}) \quad \textit{and} \quad \sqrt{N}(\hat{\Delta}_{ij} - \Delta_{ij}) \xrightarrow{d} \mathcal{N}(0, \Sigma_{ii} + \Sigma_{jj} - 2\Sigma_{ij}).$$

These results allow us to construct confidence intervals for the true θ_i and Δ_{ij}, provided any consistent estimate $\hat{\Sigma}$ of the asymptotic covariance matrix Σ in Equation (4):

[4] It is, for example, possible to argue using standard concentration and perturbation results from high-dimensional statistics (e.g., the Matrix Bernstein inequality and Wedin's $\sin\Theta$ theorem) that our cross-fold iterative SVD algorithm concentrates toward a deterministic linear projection onto the principal singular subspace of the population covariance matrix, with mild assumptions. This analysis, however, is outside the primary scope of this paper; we provide this theoretical sketch for intuition and rely on our empirical results to demonstrate the stability and coverage of the algorithm in practice.

Corollary A.4 (Confidence intervals). *Let $\hat{\Sigma} \xrightarrow{p} \Sigma$, for Σ defined in Equation (4). Let $z_{1-\alpha/2}$ denote the $(1-\alpha/2)$-quantile of the standard normal distribution for $\alpha \in (0,1)$, and let*

$$\mathcal{C}_\alpha(\hat{\theta}_i) = \left[\hat{\theta}_i \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\Sigma}_{ii}}{N}}\right] \quad \text{and} \quad \mathcal{C}_\alpha(\hat{\Delta}_{ij}) = \left[\hat{\Delta}_{ij} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{\Sigma}_{ii}+\hat{\Sigma}_{jj}-2\hat{\Sigma}_{ij}}{N}}\right].$$

Then, under the assumptions of Theorem A.2, the intervals $\mathcal{C}_\alpha(\hat{\theta}_i)$ and $\mathcal{C}_\alpha(\hat{\Delta}_{ij})$ satisfy:

$$\liminf_{N\to\infty} \mathbb{P}\big(\theta_i \in \mathcal{C}_\alpha(\hat{\theta}_i)\big) \geq 1-\alpha \quad \text{and} \quad \liminf_{N\to\infty} \mathbb{P}\big(\Delta_{ij} \in \mathcal{C}_\alpha(\hat{\Delta}_{ij})\big) \geq 1-\alpha.$$

We are now ready to provide the formal statement of Theorem 1.

Theorem A.5 (CI Coverage of CollabEval: Formal). *Suppose Assumption A.1 holds, the evaluation scores S_{ij} are strictly bounded, and the sampling conditions of Theorem A.2 are satisfied. Let $\hat{\Sigma}$ be the empirical covariance estimator constructed using the cross-fold predictions $\hat{Y}$ and true observed evaluation scores S. Then, $\hat{\Sigma} \xrightarrow{p} \Sigma$, and consequently,*

$$\liminf_{N\to\infty} \mathbb{P}\big(\theta_i \in \mathcal{C}_\alpha(\hat{\theta}_i)\big) \geq 1-\alpha \quad \text{and} \quad \liminf_{N\to\infty} \mathbb{P}\big(\Delta_{ij} \in \mathcal{C}_\alpha(\hat{\Delta}_{ij})\big) \geq 1-\alpha.$$

Remark A.6. *Note that these intervals guarantee marginal coverage for individual θ_i and Δ_{ij}. To obtain simultaneous coverage for the full vector of parameters $\theta \in \mathbb{R}^M$, one can construct a confidence ellipsoid based on the χ^2 distribution, or apply a Bonferroni correction by using marginal intervals with level $1-\alpha/M$. See a similar discussion for multivariate PPI++ in Angelopoulos et al. (2023b), and for deriving intervals for full model rankings from pairwise comparisons in Chatzi et al. (2024).*

A.2 Power tuning

Angelopoulos et al. (2023b) proposed "power tuning" as a way to improve upon the standard PPI estimator by adapting to the quality of the predictions $\hat{Y}$ through the use of a learned scalar $\lambda \in \mathbb{R}$ that can reduce variance when setting $\hat{Y}_i' = \lambda_i \hat{Y}_i$. We extend this to our setting to learn a finite set of weights $\hat{\lambda} \in \mathbb{R}^M$ for the predictions $\hat{Y}_i$ of each row i. By Slusky's theorem, this weighting preserves asymptotic normality as long as $\hat{\lambda} \xrightarrow{p} \lambda$, and is picked to minimize variance. The power-tuned estimate for θ_i is:

$$\hat{\theta}_i^{\lambda_i} = \frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i} S_{ij} - \lambda_i\left(\frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i}\hat{Y}_{ij} - \frac{1}{N}\sum_{j=1}^{N}\hat{Y}_{ij}\right),$$

while the power-tuned estimate for the difference is $\hat{\Delta}_{ij}^{\lambda_{ij}} = \hat{\theta}_i^{\lambda_i} - \hat{\theta}_j^{\lambda_j}$. Optimal values for λ differ depending on if the goal is to estimate θ_i and θ_j individually, or their difference Δ_{ij}.

Proposition A.7 (Optimal weights for $\hat{\theta}_i^\lambda$ and $\hat{\Delta}_{ij}^\lambda$). *Assume $\mathrm{Var}(\hat{S}_i) > 0$. Then $\lambda_i^* = \frac{\mathrm{Cov}(S_i,\hat{S}_i)}{\mathrm{Var}(\hat{S}_i)}$ minimizes the asymptotic variance of $\hat{\theta}_i^{\lambda_i}$. Further assume $\mathrm{Var}(\hat{S}_j) > 0$. Let $\gamma_{ij} = \frac{p_{ij}}{p_i p_j} - 1$. Then*

$$\begin{bmatrix}\lambda_i^* \\ \lambda_j^*\end{bmatrix} = \underbrace{\begin{bmatrix}\gamma_{ii}\mathrm{Var}(\hat{S}_i) & -\gamma_{ij}\mathrm{Cov}(\hat{S}_i,\hat{S}_j) \\ -\gamma_{ij}\mathrm{Cov}(\hat{S}_i,\hat{S}_j) & \gamma_{jj}\mathrm{Var}(\hat{S}_j)\end{bmatrix}^{-1}}_{Q} \underbrace{\begin{bmatrix}\gamma_{ii}\mathrm{Cov}(S_i,\hat{S}_i) - \gamma_{ij}\mathrm{Cov}(\hat{S}_i,S_j) \\ \gamma_{jj}\mathrm{Cov}(S_j,\hat{S}_j) - \gamma_{ij}\mathrm{Cov}(S_i,\hat{S}_j)\end{bmatrix}}_{u} \tag{5}$$

minimizes the asymptotic variance of $\hat{\Delta}_{ij}^{\lambda_{ij}}$.

Remark A.8. *Note that the joint optimal weights $(\lambda_i^*, \lambda_j^*)^\top$ defined in Equation (5) are equivalent to the individual optimal weights $\lambda_i^* = \mathrm{Cov}(S_i,\hat{S}_i)/\mathrm{Var}(\hat{S}_i)$ and $\lambda_j^* = \mathrm{Cov}(S_j,\hat{S}_j)/\mathrm{Var}(\hat{S}_j)$ when the cross-model terms are zero. This occurs under either of the following conditions:*

- *The observation sets are independent such that $\gamma_{ij} = 0$. As $N \to \infty$, this implies the joint observation fraction p_{ij} equals the product of individual fractions $p_i p_j$.*

- *The predicted scores of one model provide no statistical information regarding the true scores or predictions of the other model, specifically* $\mathrm{Cov}(\hat{S}_i, \hat{S}_j) = 0$ *and* $\mathrm{Cov}(\hat{S}_i, S_j) = 0$.

In these cases, the matrix Q becomes diagonal and the vector u simplifies, allowing the joint system to decouple into separate univariate optimizations. However, in practical settings where models are highly correlated or paired sampling is used ($\mathcal{J}_i = \mathcal{J}_j$), the joint weights typically diverge from individual ones to better minimize the variance of the difference estimate.

Power-tuning ensures that the asymptotic variance of either of our estimates $\hat{\theta}_i^{\lambda_i^*}$ and $\hat{\Delta}_{ij}^{\lambda_{ij}^*}$ is at most that of the classical empirical estimator. To see this, let $\hat{\theta}_i^{\text{classic}} = \frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i} S_{ij}$ be the empirical mean estimate, and let $\hat{\Delta}_{ij}^{\text{classic}} = \hat{\theta}_i^{\text{classic}} - \hat{\theta}_j^{\text{classic}}$ be the empirical mean difference estimate. The following result then gives the asymptotic variance of the power-tuned estimate in terms of the variance of the corresponding classical estimate, less a non-negative term:

Proposition A.9. *Let $\hat{\lambda} \xrightarrow{p} \lambda^*$, where λ^* is defined accordingly for $\hat{\theta}_i^{\lambda_i^*}$ and $\hat{\Delta}_{ij}^{\lambda_{ij}^*}$. Then*

$$\sqrt{N}(\hat{\theta}_i^{\hat{\lambda}_i} - \theta_i) \xrightarrow{d} \mathcal{N}\left(0, \mathrm{Var}(\hat{\theta}_i^{\text{classic}}) - \left(\frac{1}{p_i} - 1\right)\frac{\mathrm{Cov}(S_i, \hat{S}_i)^2}{\mathrm{Var}(\hat{S}_i)}\right) \quad \textit{and}$$

$$\sqrt{N}(\hat{\Delta}_{ij}^{\hat{\lambda}_{ij}} - \Delta_{ij}) \xrightarrow{d} \mathcal{N}(0, \mathrm{Var}(\hat{\Delta}_{ij}^{\text{classic}}) - u^\top Q^{-1} u),$$

for u and Q defined in Equation (5), assuming $\mathrm{Var}(\hat{S}_i) > 0$ and $\mathrm{Var}(\hat{S}_j) > 0$.

Remark A.10. *Note that Q^{-1} is PSD, which implies $u^\top Q^{-1} u \geq 0$.*

We use the empirical covariance matrix of $(S, \hat{S})$ to estimate λ^*.

B Dataset details

AlpacaEval 2.0. AlpacaEval (Li et al., 2023) evaluates the instruction-following capabilities of AI models. Models generate responses to user instructions, which an AI autorater (GPT-4 Turbo) then scores against a baseline (also based on GPT-4 Turbo) to determine a win-rate. We select the latest 17 submissions to the AlpacaEval Github repository as our target models, and use the remaining 204 submissions that were submitted before them as anchor models.

MMLU. The Massive Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2020) scores over 14k multiple-choice questions for binary accuracy. We evaluate six target LLMs from the Open LLM Leaderboard — Amber-6.7B (Liu et al., 2023), OLMo1/2-7B (Groeneveld et al., 2024; OLMo et al., 2024), Pythia-2.8B/6.9B (Biderman et al., 2023), K2-65B (Liu et al., 2025) — alongside 102 anchor LLMs, following Hofmann et al. (2025).

Attributed Question Answering (AQA). AQA (Bohnet et al., 2023) assesses whether a generated answer is both correct and supported by provided evidence, yielding an NLI-based attribution score. We evaluate 21 systems from Bohnet et al. (2023) using a leave-two-out protocol: two models serve as targets while the remaining 19 act as anchors. This enables both target-vs-target and target-vs-anchor model comparisons.

WMT24++ Machine Translation Benchmark. WMT24++ evaluates translations of 998 English paragraphs into 55 languages (Deutsch et al., 2025). Full evaluation of a new model requires both consuming and generating nearly two million tokens (55 languages $\times$ 998 paragraphs $\times \sim$ 35 tokens per paragraph). We use the initial release of 15 translation systems provided by Deutsch et al. (2025), and flatten all language-model pairs into an 825×998 evaluation matrix that uses MetricX (Juraska et al., 2024) as the evaluation score. Applying a leave-two-out protocol similar to the AQA setting within each language yields 2×55 target rows. However, we restrict all comparisons and paired sampling to be strictly within-language (e.g., we compare model A vs. model B on en-de translations separately from en-fr).

SWE-bench. SWE-bench (Jimenez et al., 2024) evaluates the ability of language models to resolve real-world software engineering issues from GitHub. Given a codebase and an issue description,

models are tasked with generating a patch that fixes the bug or implements the requested feature. We evaluate the success rate of 12 systems for which item-level scores are publicly available from the SWE-bench leaderboard using the mini-SWE-agent V2 harness, and use the same leave-one-out and leave-two-out protocol as our AQA experiment.

C Additional Experimental Results

C.1 Effective Rank Analysis

Figure 3 plots the top 32 normalized singular values (σ_i/σ_1) across all five datasets. The rapid decay of the singular values and the rapid increase of the cumulative explained variance support the hypothesis that the score matrices have low effective rank (which CollabEval can exploit).

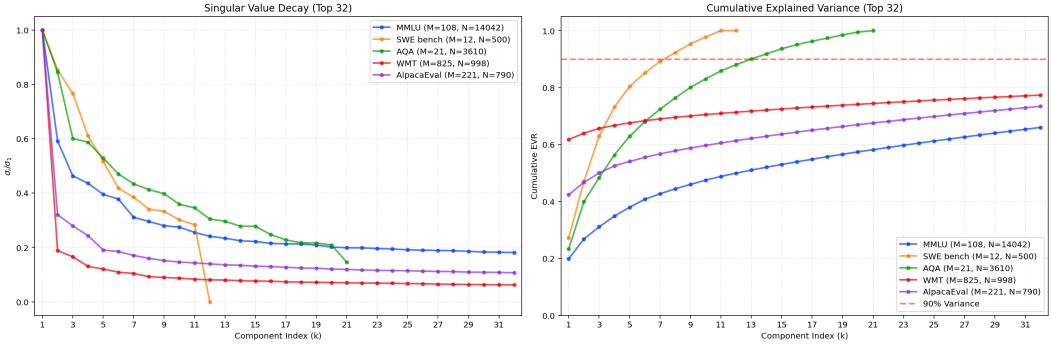

Figure 3: Singular value decay of the evaluation data matrices for the tasks we consider.

C.2 Additional Baselines

Figure 4 compares the mean squared error of CollabEval against naive imputation, Anchor Points, and standard prediction-powered inference. The Naive Estimate uses the same matrix completion algorithm (IterativeSVD) as CollabEval, but does not correct for bias using the cross-fold procedure. For these results, CollabEval, the naive estimate, and PPI all use paired sampling (Anchor Points also uses paired samples by design). CollabEval consistently yields the lowest error across varying labeled data fractions. Furthermore, while Anchor Points can perform quite well at higher labeled fractions, it also exhibits high variance across datasets—whereas CollabEval is empirically quite robust.

C.3 Per-Task Results

While the average performance in Figure 2 demonstrates the overall efficacy of CollabEval, examining the results on a per-dataset basis reveals specific strengths and other interesting dynamics depending on the evaluation context. For instance, on AQA (Figure 7) we see particularly strong performance for individual mean estimation, with over a 30% reduction in confidence interval (CI) width when p is small, and up to a 20% reduction even when half the data was labeled ($p = 0.5$). Similarly, the WMT24++ Machine Translation evaluation (Figure 8) shows large efficiency gains for pairwise difference estimations, and yields CI width reduction of up to over 30% in target-vs-target settings at low sampling rates, and also maintains up to a 20% reduction at moderate sampling rates.

Across these benchmarks, the empirical results further highlight how the choice of sampling strategy interacts with the estimand; while independent (IID) and paired sampling performed virtually identically for estimating individual means, paired sampling is advantageous for estimating differences between target models. In these comparative target-vs-target settings, CollabEval combined with paired sampling consistently yielded the tightest confidence intervals and lowest overall error, noticeably improving upon the already competitive classical paired estimator.

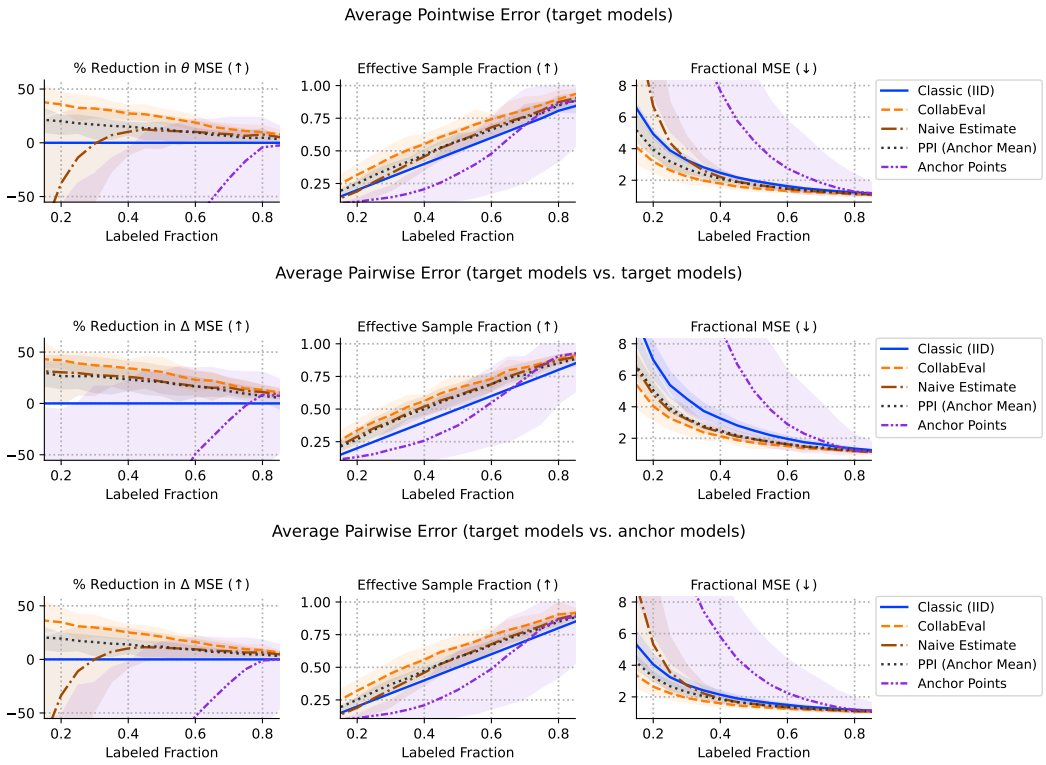

Figure 4: Results when comparing CollabEval to naive imputation, Anchor Points, and standard PPI with per-prompt anchor means. Shaded bands show the worst and best performance across datasets.

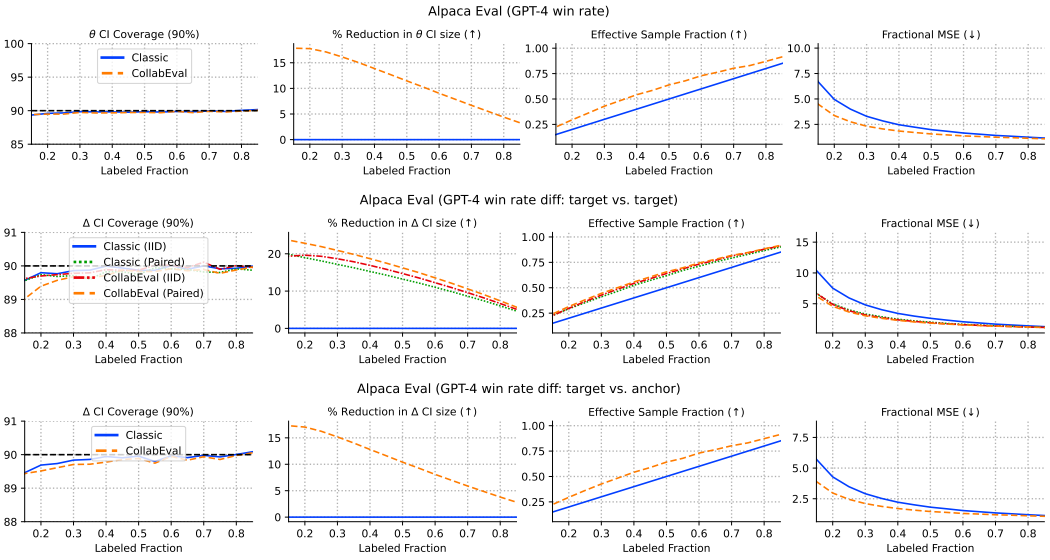

Figure 5: Results for AlpacaEval.

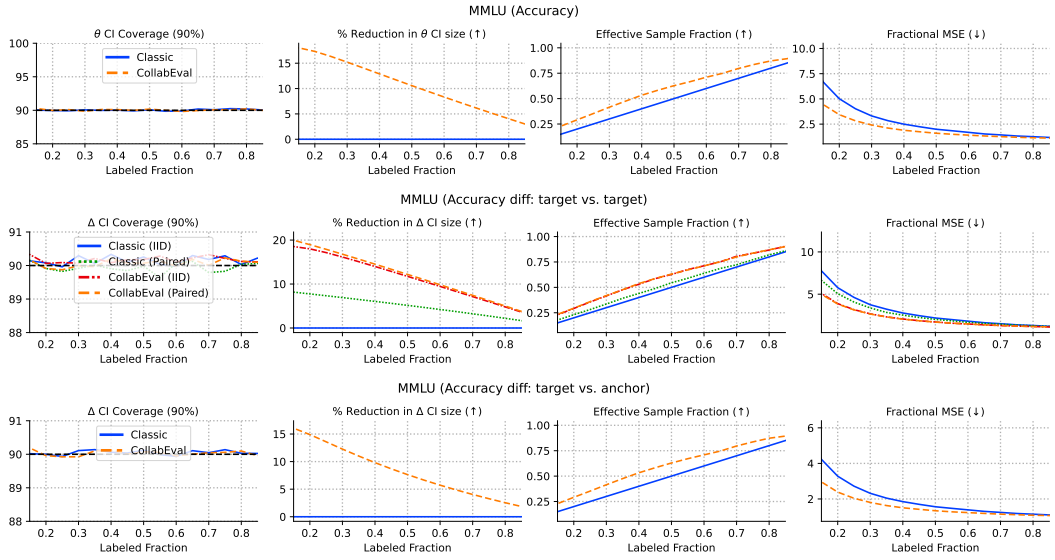

Figure 6: Results for MMLU.

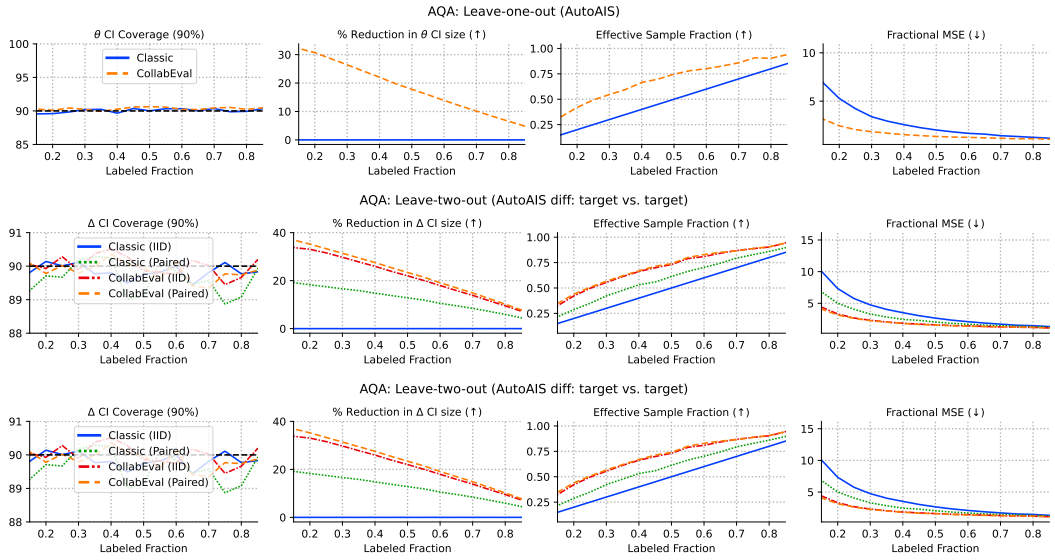

Figure 7: Results for AQA.

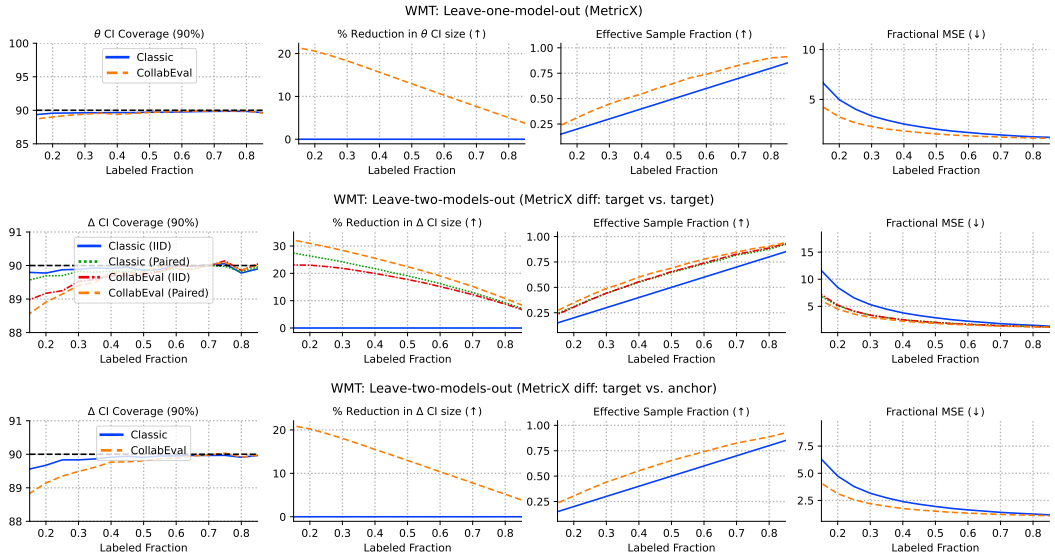

Figure 8: Results for WMT24++.

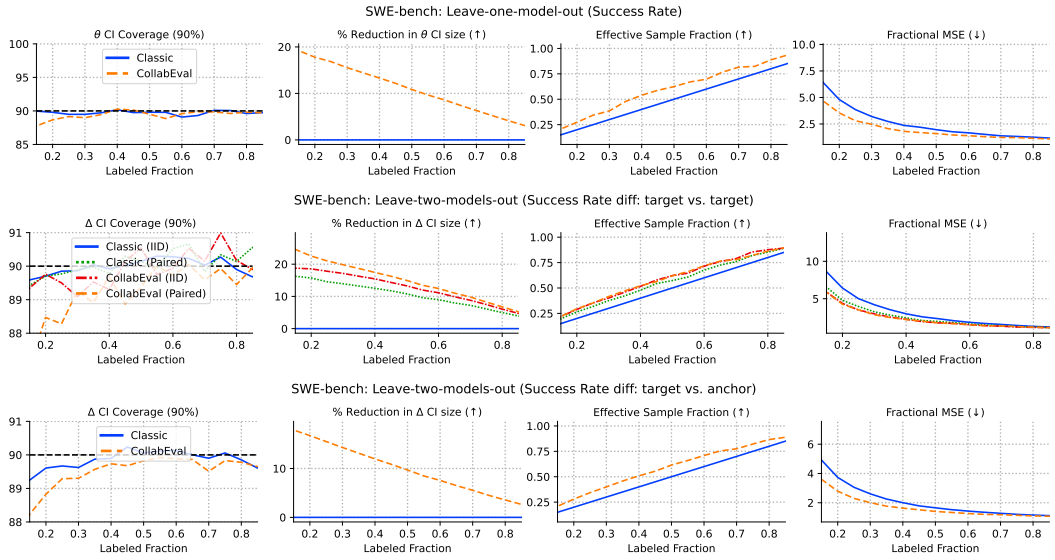

Figure 9: Results for SWE-bench.

C.4 ANCHOR SET ABLATION

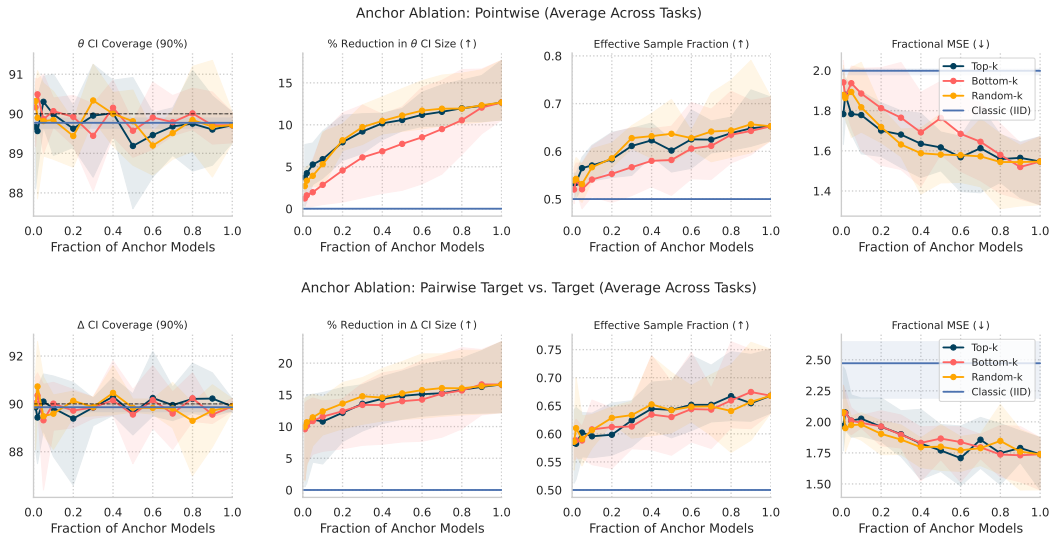

Figure 10: Performance metrics averaged across all five datasets when varying the number of anchor models in the anchor set $\mathcal{A}$, for a fixed labeled data fraction of 50%. Shaded bands show the worst and best performance across datasets. We test three selection strategies: (1) Top-k, in which we keep only the highest scoring anchor models in $\mathcal{A}$; Bottom-k, in which we keep only the lowest scoring anchor models in $\mathcal{A}$; and Random-k in which we sample k random models from $\mathcal{A}$. We choose k as a percentage of the total number of available anchors starting at 2% and always include at least 1 anchor, i.e., `k = max(1, int(pct * total_anchors))`. Note that there is some variability in the total number of anchors across tasks. As supported by our theory, CI coverage is largely unaffected by the anchor set, while the estimation efficiency improves as the number of anchor models increases (and becomes more predictive of the target models). As also expected, unbiased selection of anchors (i.e., Random-k) generally gives the best results, by a small margin. We do not report pairwise metrics for target vs. anchor models, as the anchor set changes at each iteration.

D MATRIX COMPLETION ALGORITHM DETAILS AND ABLATIONS

This appendix presents algorithms for iterative matrix completion (Algorithm 1), cross-fold matrix completion (Algorithm 2), CollabEval estimation for individual estimands (Algorithm 3), and estimation for pairwise comparisons (Algorithm 4). The default sequence of ranks that we use for IterativeSVD is $\mathcal{R} = (1, 2, 4, 8, 16, 16, 16, 16)$, and the default number of folds is $K = 10$. These were not extensively tuned; we picked a setting that had a good tradeoff between speed and predictive ability. IterativeSVD is based on HardImpute (Mazumder et al., 2010) which is intended to iterate until convergence with progressively higher ranks. Section D.1 provides an ablation across datasets at a fixed labeled fraction of 50% that suggests that performance improves as the max rank in $\mathcal{R}$ increases. Section D.2 compares IterativeSVD to several alternative matrix completion algorithms: a neural network baseline, nuclear norm minimization, and an item response theory (IRT) model.

Algorithm 1 Iterative Matrix Completion via Truncated SVD (`IterativeSVD`)

Definitions: $S \in \mathbb{R}^{M \times N}$ is the incomplete label matrix (M models, N items). $\Omega \in \{0,1\}^{M \times N}$ is the binary mask of observed entries. $\mathcal{R}$ is a sequence of ranks. $\texttt{SVD}_r(A) = \sum_{k=1}^{r} \sigma_k u_k v_k^\top$, where σ_k, u_k, v_k are the k-th singular value and singular vectors of A respectively.

Initialize missing entries of S with row means.
$\hat{Y} = \texttt{RowMean}(S, \Omega)$
for $r \in \mathcal{R}$ **do**
    # Compute rank-r truncated SVD reconstruction and update unobserved entries.
    $\hat{Y}_{new} = \texttt{SVD}_r(\hat{Y})$
    $\hat{Y}_{ij} = \hat{Y}_{new,ij}$ for $(i,j) \notin \Omega$
return $\hat{Y}$

Algorithm 2 Cross-Fold Matrix Completion (`XFoldComplete`)

Definitions: $S \in \mathbb{R}^{M \times N}$ is the incomplete label matrix (M models, N items). $\Omega \in \{0,1\}^{M \times N}$ is the binary mask of observed entries, where $\mathcal{J}_i = \{j : \Omega_{ij} = 1\}$. $\mathcal{T}$ is the set of target models. K is the number of folds. $\mathcal{R}$ is a sequence of ranks. The default is (1, 2, 4, 8, 16, 16, 16, 16).

Identify active prompts $\mathcal{U} = \bigcup_{i \in \mathcal{T}} \mathcal{J}_i$ and partition them into K disjoint folds $\mathcal{U}_1, \ldots, \mathcal{U}_K$.
for $k = 1, \ldots, K$ **do**
    # Define training mask and complete fold using iterative SVD.
    Define $\Omega^{(k)}_{train}$ such that $\Omega^{(k)}_{train,ij} = 0$ if $i \in \mathcal{T}$ and $j \in \mathcal{U}_k$, and Ω_{ij} otherwise.
    $\hat{Y}^{(k)} = \texttt{IterativeSVD}(S, \Omega^{(k)}_{train}, \mathcal{R})$
Aggregate predictions for unobserved and anchor entries.
$\hat{Y}_{ij} = \hat{Y}^{(k)}_{ij}$ if $(i,j) \in \Omega \setminus \Omega^{(k)}_{train}$, and $\frac{1}{K}\sum_{k=1}^{K} \hat{Y}^{(k)}_{ij}$ otherwise.
return $\hat{Y}$

Algorithm 3 CollabEval for Single Model Performance (`EstimateTheta`)

Definitions: $\hat{Y}$ is the completed matrix from Algorithm 2. S is the incomplete matrix of true evaluation scores. Ω is the observation mask. α is the desired coverage of the CI for θ_i.
Input: i, the index of the model.

Get raw mean estimate + control variate.
$\hat{\mu}_i = \frac{1}{n_i}\sum_{j \in \mathcal{J}_i} S_{ij}$ where $\mathcal{J}_i = \{j : \Omega_{ij} = 1\}$ and $n_i = |\mathcal{J}_i|$
$\widehat{CV}_i = \frac{1}{n_i}\sum_{j \in \mathcal{J}_i} \hat{Y}_{ij} - \frac{1}{N}\sum_{j=1}^{N} \hat{Y}_{ij}$

Power-tune single model scaling, using empirical covariance estimates.
$\hat{\lambda}_i = \frac{\widehat{\text{Cov}}_{j \in \mathcal{J}_i}(S_{ij}, \hat{Y}_{ij})}{\widehat{\text{Var}}_{j \in [N]}(\hat{Y}_{ij})}$

Combine raw mean estimate with weighted control variate.
$\hat{\theta}_i^{\hat{\lambda}_i} = \hat{\mu}_i - \hat{\lambda}_i \widehat{CV}_i$

Estimate variance for confidence interval.
$\widehat{\sigma}_i^2 = \left[\frac{1}{n_i}\widehat{\text{Var}}_{j \in \mathcal{J}_i}(S_{ij}) - \left(\frac{1}{n_i} - \frac{1}{N}\right)\hat{\lambda}_i^2 \widehat{\text{Var}}_{j \in [N]}(\hat{Y}_{ij})\right]^+$
return $\left(\hat{\theta}_i^{\hat{\lambda}_i}, \left\{\hat{\theta}_i^{\hat{\lambda}_i} \pm z_{1-\frac{\alpha}{2}}\widehat{\sigma}_i\right\}\right)$

Algorithm 4 CollabEval for Model Comparisons (`EstimateDelta`)

Definitions: $\hat{Y}$ is the completed matrix from Algorithm 2. S is the incomplete matrix of true evaluation scores. Ω is the observation mask. α is the desired coverage of the CI for Δ_{ij}.
Input: (i, j), the indices of the model pair.

Compute overlap metrics.
$\mathcal{J}_i, \mathcal{J}_j = \{k \colon \Omega_{ik} = 1\}, \{k \colon \Omega_{jk} = 1\}$
$n_i, n_j, n_{ij}, \gamma_{ij} = |\mathcal{J}_i|, |\mathcal{J}_j|, |\mathcal{J}_i \cap \mathcal{J}_j|, \frac{|\mathcal{J}_i \cap \mathcal{J}_j|}{|\mathcal{J}_i||\mathcal{J}_j|} - \frac{1}{N}$
Construct $\hat{Q}$ and $\hat{u}$ using empirical covariance estimates.
$$\hat{Q} = \begin{pmatrix} \gamma_{ii}\widehat{\mathrm{Var}}_{k\in[N]}(\hat{Y}_{ik}) & -\gamma_{ij}\mathrm{Cov}_{k\in[N]}(\hat{Y}_{ik}, \hat{Y}_{jk}) \\ -\gamma_{ji}\mathrm{Cov}_{k\in[N]}(\hat{Y}_{jk}, \hat{Y}_{ik}) & \gamma_{jj}\widehat{\mathrm{Var}}_{k\in[N]}(\hat{Y}_{jk}) \end{pmatrix}$$
$$\hat{u} = \begin{pmatrix} \gamma_{ii}\widehat{\mathrm{Cov}}_{k\in\mathcal{J}_i}(S_{ik}, \hat{Y}_{ik}) - \gamma_{ij}\widehat{\mathrm{Cov}}_{k\in\mathcal{J}_j}(\hat{Y}_{ik}, S_{jk}) \\ \gamma_{jj}\widehat{\mathrm{Cov}}_{k\in\mathcal{J}_j}(S_{jk}, \hat{Y}_{jk}) - \gamma_{ij}\widehat{\mathrm{Cov}}_{k\in\mathcal{J}_i}(\hat{Y}_{jk}, S_{ik}) \end{pmatrix}$$
Solve linear system. Note $\hat{\lambda} = (\hat{\lambda}_i, \hat{\lambda}_j)^T$ and $\dagger$ is the pseudo-inverse.
$\hat{\lambda} = \hat{Q}^{\dagger}\hat{u}$

Compute difference estimate from mean estimates using $\hat{\lambda}$ (same as Algorithm 3).
$\hat{\Delta}_{ij}^{\hat{\lambda}} = \hat{\theta}_i^{\hat{\lambda}_i} - \hat{\theta}_j^{\hat{\lambda}_j}$

Estimate variance for confidence interval (sample covariance evaluates to 0 if $|\mathcal{J}_i \cap \mathcal{J}_j| \leq 1$).
$\widehat{\sigma}_{ij}^2 = \left[\frac{1}{n_i}\widehat{\mathrm{Var}}_{k\in\mathcal{J}_i}(S_{ik}) + \frac{1}{n_j}\widehat{\mathrm{Var}}_{k\in\mathcal{J}_j}(S_{jk}) - 2\frac{n_{ij}}{n_i n_j}\widehat{\mathrm{Cov}}_{k\in\mathcal{J}_i\cap\mathcal{J}_j}(S_{ik}, S_{jk}) - \hat{\lambda}^{\top}\hat{u}\right]^{+}$
return $\left(\hat{\Delta}_{ij}^{\hat{\lambda}}, \left\{\hat{\Delta}^{\hat{\lambda}_{ij}} \pm z_{1-\frac{\alpha}{2}}\widehat{\sigma}_{ij}\right\}\right)$

D.1 IterativeSVD ablations

Table 1 provides an ablation for IterativeSVD (Algorithm 1) across datasets at a fixed labeled fraction of 50%, showing that increasing the max rank in $\mathcal{R}$ leads to better performance (it starts to plateau around rank 16). We progress the rank sequence in powers of 2 from $(1), (1,2), \ldots, (1,2,\ldots,32)$).

Task	rank=1	rank=2	rank=4	rank=8	rank=16	rank=32
WMT	+8.4%	+10.7%	+11.4%	+11.6%	+12.4%	+13.4%
AQA NLI	+9.3%	+11.2%	+13.4%	+16.8%	+17.5%	+17.9%
MMLU	+5.0%	+5.7%	+8.2%	+9.5%	+10.6%	+10.8%
Alpaca Eval	+5.9%	+8.1%	+9.5%	+10.8%	+11.4%	+12.0%
SWE-bench	+9.3%	+10.3%	+10.3%	+10.2%	+10.7%	+10.8%
Average	+7.6%	+9.2%	+10.5%	+11.8%	+12.5%	+13.0%

Table 1: IterativeSVD rank ablation at a fixed 50% labeled fraction. Values represent the % CI size reduction vs Classic (IID) (higher is better).

Table 2 provides a similar ablation over the number of folds K, again at a fixed 50% labeled fraction. We find that IterativeSVD is empirically not particularly sensitive to this choice—though, in theory, the quality of the matrix completion can improve with higher K (more observed data per completion) which can narrow CI width. CI coverage remains unaffected, consistent with our theoretical results.

Task	$K=2$	$K=3$	$K=5$	$K=10$
WMT	90.1%, +12.7%	89.9%, +13.2%	89.6%, +12.4%	90.0%, +12.0%
AQA NLI	88.1%, +17.8%	89.7%, +17.4%	91.6%, +17.4%	89.1%, +17.6%
MMLU	90.1%, +10.6%	89.4%, +10.6%	89.2%, +10.6%	89.9%, +10.6%
Alpaca Eval	89.6%, +11.2%	90.5%, +11.4%	89.5%, +11.5%	90.2%, +11.5%
SWE-bench	89.5%, +10.6%	91.2%, +10.7%	90.5%, +10.6%	89.8%, +10.7%
Average	89.5%, +12.6%	90.1%, +12.7%	90.1%, +12.5%	89.8%, +12.5%

Table 2: IterativeSVD fold ablation at a fixed 50% labeled fraction. Values represent average coverage and % CI size reduction vs Classic (IID), using default ranks $(1, 2, 4, 8, 16, 16, 16, 16)$.

D.2 Comparison to other matrix completion options

Table 3 compares CollabEval across four matrix completion methods at a fixed 50% labeled data fraction: IterativeSVD, a neural network (NN), nuclear norm minimization, and a 2-parameter logistic (2PL) item response theory (IRT) model. The NN baseline uses a 1-layer, 64-dimensional multilayer perceptron applied to the vector of anchor model scores to predict target model scores, trained with either a regression or binary cross-entropy loss depending on the task. The 2PL IRT model applies only to the two binary classification datasets (MMLU and SWE-bench).

All evaluated methods reduce confidence interval size relative to classical estimation. The NN yields the largest average reduction (+13.5%) but exhibits higher variance across individual tasks at lower labeled data fractions. IterativeSVD provides a better tradeoff of computational speed, implementation simplicity, and consistent empirical performance. The statistical validity of CollabEval remains independent of the specific matrix completion algorithm used.

E Proofs

E.1 Proof of Theorem A.2

We proceed in two steps, following a similar analysis as Zrnic & Candès (2024). First, we define an idealized estimator $\tilde{\theta}$ using the deterministic population-limit predictions $\hat{S}_{ij}$, and prove its asymptotic normality via standard central limit theorem arguments. Then, we show that the empirical estimator $\hat{\theta}$ that uses cross-prediction is asymptotically equivalent.

Task	2PL IRT	IterativeSVD	NN	Nuclear Norm
WMT	N/A	+12.9%	+12.6%	+9.5%
AQA	N/A	+17.3%	+19.1%	+15.4%
MMLU	+1.7%	+10.6%	+15.0%	+9.5%
Alpaca Eval	N/A	+11.5%	+9.5%	+5.9%
SWE-bench	+5.9%	+10.5%	+11.2%	+10.6%
Average	+3.8%	+12.5%	+13.5%	+10.2%

Table 3: Matrix completion method ablation at a fixed 50% labeled fraction. Values represent the % reduction in pointwise θ CI size relative to the classic (IID) estimate.

Lemma E.1. *Define the idealized predictor:*

$$\tilde{\theta}_i = \frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i} S_{ij} - \left(\frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i}\hat{\underline{S}}_{ij} - \frac{1}{N}\sum_{j=1}^{N}\hat{\underline{S}}_{ij}\right).$$

Then, under the conditions of Theorem A.2, $\sqrt{N}(\tilde{\theta}-\theta)\xrightarrow{d}\mathcal{N}(0,\Sigma)$*, where*

$$\Sigma_{ij} = \frac{p_{ij}}{p_i p_j}\mathrm{Cov}\left(S_i, S_j\right) + \left(\frac{p_{ij}}{p_i p_j} - 1\right)\left[\mathrm{Cov}\left(\hat{\underline{S}}_i, \hat{\underline{S}}_j\right) - \mathrm{Cov}\left(\hat{\underline{S}}_i, S_j\right) - \mathrm{Cov}\left(S_i, \hat{\underline{S}}_j\right)\right].$$

Proof. We begin by rewriting $\tilde{\theta}$ as

$$\tilde{\theta} = \frac{1}{N}\sum_{j=1}^{N} Z_j,$$

where Z_j is an M-dimensional random vector with

$$Z_j(i) = \hat{\underline{S}}_{ij} + \frac{N}{n_i}\Omega_{ij}(S_{ij} - \hat{\underline{S}}_{ij}).$$

Because the N evaluation prompts are drawn i.i.d. and the population-limit prediction $\hat{\underline{S}}_{ij} = h_i(X_j)$ where $X_j = \{S_{kj} \colon \Omega_{kj} = 1, k \in \mathcal{A}\}$ is a deterministic function of only the observed anchor model scores for the j-th prompt (Assumption A.1), the vectors $Z_1, Z_2, \ldots, Z_N$ are conditionally independent given the observation mask Ω, and the observed anchor model scores.

To establish asymptotic normality, we view $\{Z_j\}_{j=1}^N$ as a triangular array, since the scaling coefficients $\frac{N}{n_i}$ depend on N. By assumption, the sampling budgets scale deterministically as $n_i/N \to p_i \in (0,1]$, meaning $\frac{N}{n_i}$ are bounded by a constant for sufficiently large N. Because $(S_{ij}, \hat{\underline{S}}_{ij})$ have finite covariance by assumption, and Z_j is a linear combination of these variables with bounded coefficients and $\Omega_{ij} \in \{0,1\}$, the array is bounded by a square-integrable distribution; this implies the Lindeberg condition.

Applying the Multivariate Lindeberg-Feller CLT to this array, we have:

$$\sqrt{N}(\tilde{\theta} - \mathbb{E}[\tilde{\theta} \mid \Omega]) \mid \Omega \xrightarrow{d} \mathcal{N}(0, \Sigma_N(\Omega))$$

for some conditional covariance matrix $\Sigma_N(\Omega)$. Furthermore, taking the conditional expectation yields $\mathbb{E}[Z_{ij} \mid \Omega] = \mathbb{E}[\hat{\underline{S}}_{ij}] + \frac{N}{n_i}\Omega_{ij}\mathbb{E}[S_{ij} - \hat{\underline{S}}_{ij}]$. Summing over j and substituting $\sum_{j=1}^N \Omega_{ij} = n_i$ deterministically gives $\mathbb{E}[\tilde{\theta} \mid \Omega] = \theta$. Therefore, we can substitute:

$$\sqrt{N}(\tilde{\theta} - \theta) \mid \Omega \xrightarrow{d} \mathcal{N}(0, \Sigma_N(\Omega)).$$

We now calculate $\Sigma_N(\Omega)$. By the conditional independence of Z_j:

$$\Sigma_N(\Omega) = \mathrm{Var}(\sqrt{N}\tilde{\theta} \mid \Omega) = \frac{1}{N}\sum_{j=1}^{N}\mathrm{Var}(Z_j \mid \Omega).$$

Looking at the covariance between the i-th and k-th components of Z_j, we have

$$\mathrm{Cov}(Z_{ij}, Z_{kj} \mid \Omega) = \mathrm{Cov}\left(\hat{\underline{S}}_{ij} + \frac{N}{n_i}\Omega_{ij}(S_{ij} - \hat{\underline{S}}_{ij}), \hat{\underline{S}}_{kj} + \frac{N}{n_k}\Omega_{kj}(S_{kj} - \hat{\underline{S}}_{kj})\right).$$

Expanding this via bilinearity of covariance and summing over j yields:

$$[\Sigma_N(\Omega)]_{ik} = \frac{1}{N}\sum_{j=1}^{N}\Bigg[\mathrm{Cov}(\hat{\underline{S}}_{ij}, \hat{\underline{S}}_{kj}) + \frac{N}{n_k}\Omega_{kj}\mathrm{Cov}(\hat{\underline{S}}_{ij}, S_{kj} - \hat{\underline{S}}_{kj}) + \frac{N}{n_i}\Omega_{ij}\mathrm{Cov}(S_{ij} - \hat{\underline{S}}_{ij}, \hat{\underline{S}}_{kj}) + \frac{N^2}{n_i n_k}\Omega_{ij}\Omega_{kj}\mathrm{Cov}(S_{ij} - \hat{\underline{S}}_{ij}, S_{kj} - \hat{\underline{S}}_{kj})\Bigg].$$

Because the underlying prompts and prompt features (X_j, that $\hat{\underline{S}}_{ij}$ are fixed functions of) are i.i.d., the population covariances do not depend on j. Using the identities $\sum_{j=1}^{N}\Omega_{ij} = n_i$, $\sum_{j=1}^{N}\Omega_{kj} = n_k$, and $\sum_{j=1}^{N}\Omega_{ij}\Omega_{kj} = n_{ik}$ from our sampling design, the equation simplifies to:

$$\begin{aligned}
[\Sigma_N(\Omega)]_{ik} &= \mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k) + \left(\frac{1}{N}\frac{N}{n_k}n_k\right)\mathrm{Cov}(\hat{\underline{S}}_i, S_k - \hat{\underline{S}}_k) + \left(\frac{1}{N}\frac{N}{n_i}n_i\right)\mathrm{Cov}(S_i - \hat{\underline{S}}_i, \hat{\underline{S}}_k) \\
&\quad + \left(\frac{1}{N}\frac{N^2}{n_i n_k}n_{ik}\right)\mathrm{Cov}(S_i - \hat{\underline{S}}_i, S_k - \hat{\underline{S}}_k) \\
&= \mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k) + \mathrm{Cov}(\hat{\underline{S}}_i, S_k) - \mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k) + \mathrm{Cov}(S_i, \hat{\underline{S}}_k) - \mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k) \\
&\quad + \frac{N n_{ik}}{n_i n_k}\left[\mathrm{Cov}(S_i, S_k) - \mathrm{Cov}(S_i, \hat{\underline{S}}_k) - \mathrm{Cov}(\hat{\underline{S}}_i, S_k) + \mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k)\right].
\end{aligned}$$

Letting $\gamma_{ik}^N = \frac{N n_{ik}}{n_i n_k}$ and grouping terms, we finally arrive at the following conditional covariance matrix for our triangular array:

$$\Sigma_N(\Omega)_{ik} = \gamma_{ik}^N\mathrm{Cov}(S_i, S_k) + (\gamma_{ik}^N - 1)\big[\mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k) - \mathrm{Cov}(\hat{\underline{S}}_i, S_k) - \mathrm{Cov}(S_i, \hat{\underline{S}}_k)\big].$$

By our sampling assumptions, as $N \to \infty$, the deterministic budgets scale as $n_i/N \to p_i$ and $n_k/N \to p_k$, while the random overlap fraction converges in probability: $n_{ik}/N \xrightarrow{p} p_{ik}$. By the Continuous Mapping Theorem, the coefficient γ_{ik}^N converges in probability:

$$\gamma_{ik}^N = \frac{n_{ik}/N}{(n_i/N)(n_k/N)} \xrightarrow{p} \frac{p_{ik}}{p_i p_k}.$$

Consequently, $\Sigma_N(\Omega)_{ik}$ converges in probability to a constant limit matrix Σ:

$$\Sigma_N(\Omega) \xrightarrow{p} \Sigma,$$

where the entries of Σ are exactly:

$$\Sigma_{ik} = \frac{p_{ik}}{p_i p_k}\mathrm{Cov}(S_i, S_k) + \left(\frac{p_{ik}}{p_i p_k} - 1\right)\big[\mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_k) - \mathrm{Cov}(\hat{\underline{S}}_i, S_k) - \mathrm{Cov}(S_i, \hat{\underline{S}}_k)\big].$$

Since $\sqrt{N}(\tilde{\theta} - \theta) \mid \Omega \xrightarrow{d} \mathcal{N}(0, \Sigma_N(\Omega))$ and $\Sigma_N(\Omega) \xrightarrow{p} \Sigma$, we can apply Slutsky's Theorem to conclude unconditionally that $\sqrt{N}(\tilde{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \Sigma)$. □

We now are ready to state our second lemma.

Lemma E.2. *Under the conditions of Theorem A.2, as $N \to \infty$, we have $\sqrt{N}(\hat{\theta} - \tilde{\theta}) \xrightarrow{p} 0$.*

Proof. We start by writing

$$\sqrt{N}(\hat{\theta}_i - \tilde{\theta}_i) = \sqrt{N}\left(\frac{1}{N}\sum_{j=1}^{N}(\hat{Y}_{ij} - \hat{S}_{ij}) + \frac{N}{n_i}\Omega_{ij}(\hat{S}_{ij} - \hat{Y}_{ij})\right).$$

Let $w_{ij} = \sqrt{N}\left(\frac{1}{N} - \frac{\Omega_{ij}}{n_i}\right)$ and define $V_{N,k}$ as

$$\begin{aligned} V_{N,k} &= \sum_{j \in \mathcal{J}_i^{(k)}} w_{ij}(\hat{S}_{ij}^{(k)} - \hat{\underline{S}}_{ij}) + \frac{1}{K}\sum_{j \in \mathcal{G}_i} \frac{1}{\sqrt{N}}(\hat{S}_{ij}^{(k)} - \hat{\underline{S}}_{ij}) \\ &= \sum_{j \in \mathcal{J}_i^{(k)}} w_{ij}\Gamma_{ij} + \frac{1}{K}\sum_{j \in \mathcal{G}_i} \frac{1}{\sqrt{N}}\Gamma_{ij} \end{aligned}$$

where $\Gamma_{ij} = \hat{S}_{ij}^{(k)} - \hat{\underline{S}}_{ij}$, $\mathcal{J}_i^{(k)}$ are the observed indices in fold k, and $\mathcal{G}_i = \{N\} \setminus \mathcal{J}_i$ are the unobserved indices. Let $V_N = \sum_{k=1}^K V_{N,k}$. Note that $V_N = \sqrt{N}(\hat{\theta}_i - \tilde{\theta}_i)$.

Using Fact 1 from Zrnic & Candès (2024), $V_N \xrightarrow{p} 0$ iff $\mathbb{E}[\min(1, |V_N|)] \to 0$. We now apply the same analysis as in Lemma A.1 and A.2 of Zrnic & Candès (2024). Let $\psi(x) = \min(1, x)$. Since ψ is subadditive and $V_N = \sum_{k=1}^K V_{N,k}$,

$$\mathbb{E}[\psi(|V_N|)] \le \sum_{k=1}^K \mathbb{E}[\psi(|V_{N,k}|)]$$

Since ψ is concave, we condition on the mask $\Omega^{(k)}$ used for the k-th fold, and apply Jensen's:

$$\begin{aligned} \mathbb{E}[\psi(|V_{N,k}|)] &= \mathbb{E}[\mathbb{E}[\psi(|V_{N,k}|) \mid \Omega^{(k)}]] \\ &\le \mathbb{E}\left[\psi\left(\mathbb{E}[|V_{N,k}| \mid \Omega^{(k)}]\right)\right] \\ &\le \mathbb{E}\left[\psi\left(\sqrt{\mathrm{Var}(V_{N,k} \mid \Omega^{(k)})}\right)\right]. \end{aligned}$$

Since the indices $j \in \mathcal{J}_i^{(k)}$ are held-out from the training of $\hat{S}^{(k)}$, the terms in the $V_{N,k}$ are conditionally independent given $\Omega^{(k)}$, so we can write:

$$\mathrm{Var}(V_{N,k} \mid \Omega^{(k)}) = \sum_{j \in \mathcal{J}_i^{(k)}} w_{ij}^2 \mathrm{Var}(\Gamma_{ij} \mid \Omega^{(k)}) + \frac{1}{K^2 N}\sum_{j \in \mathcal{G}_i} \mathrm{Var}(\Gamma_{ij} \mid \Omega^{(k)}).$$

Note that for $j \in \mathcal{J}_i^{(k)}$, the weights $w_{ij}^2 = N(\frac{1}{N} - \frac{1}{n_i})^2 = \frac{1}{N}(1 - \frac{N}{n_i})^2$ are $O(1/N)$, since $\frac{N}{n_i} \to 1/p_i$. Since the number of terms in each sum is $O(N)$, the total variance for fold k is bounded by a constant multiple of the average stability error:

$$\mathrm{Var}(V_{N,k} \mid \Omega^{(k)}) \le C \cdot \mathrm{Var}(\hat{S}_i^{(k)} - \hat{\underline{S}}_i \mid \Omega^{(k)})$$

for some constant C, depending on the sampling fraction p_i.

Substituting this back into the expectation of the concave function ψ:

$$\sum_{k=1}^K \mathbb{E}[\psi(|V_{N,k}|)] \le \sum_{k=1}^K \mathbb{E}\left[\psi\left(\sqrt{C \cdot \mathrm{Var}(\hat{S}_i^{(k)} - \hat{\underline{S}}_i \mid \Omega^{(k)})}\right)\right].$$

By Assumption A.1, $\sqrt{K\mathrm{Var}(\hat{S}_i^{(1)} - \hat{\underline{S}}_i \mid \Omega^{(1)})} \xrightarrow{L_1} 0$ as $N \to \infty$. Since $\psi(x) = \min(1, x) \le x$ for $x \ge 0$, and the expectation of the square root vanishes, it follows that:

$$\mathbb{E}[\psi(|V_N|)] \le \sum_{k=1}^K \mathbb{E}[\psi(|V_{N,k}|)] \to 0.$$

By Fact 1, $\mathbb{E}[\min(1, |V_N|)] \to 0$ implies $V_N \xrightarrow{p} 0$. Thus, $\sqrt{N}(\hat{\theta}_i - \tilde{\theta}_i) \xrightarrow{p} 0$ for each $i \in [M]$. Since M is finite, the vector convergence $\sqrt{N}(\hat{\theta} - \tilde{\theta}) \xrightarrow{p} 0$ holds simultaneously. □

We can now prove Theorem A.2.

Proof. Applying Slutsky's Theorem with the results of Lemmas E.1 and E.2 yields:

$$\sqrt{N}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \Sigma).$$

□

E.2 Proof of Corollary A.3

Proof. Follows directly from Theorem A.2 and the fact that

$$X \xrightarrow{d} \mathcal{N}(\mu, \Sigma) \Longrightarrow a^\top X \xrightarrow{d} \mathcal{N}(a^\top \mu, a^\top \Sigma a).$$

for constant vector $a \in \mathbb{R}^M$. □

E.3 Proof of Corollary A.4

Proof. This follows from Theorem A.2 and Corollary A.3, together with Slutsky's Theorem. □

E.4 Proof of Theorem A.5

Proof. We prove this by establishing the convergence in probability of the individual components of $\hat{\Sigma}_{ij}$ to constants, and then applying Slutsky's Theorem.

By Assumption A.1, $\sqrt{K\mathrm{Var}(\hat{S}_{ij}^{(k)} - \underline{\hat{S}}_{ij} \mid \Omega^{(k)})} \xrightarrow{L_1} 0$. Because $\underline{\hat{S}}_{ij}$ is the deterministic population limit, the conditional bias $\mathbb{E}[\hat{S}_{ij}^{(k)} - \underline{\hat{S}}_{ij} \mid \Omega^{(k)}]$ also vanishes. This combined with strictly bounded evaluation scores guarantees unconditional mean-square (L_2) convergence, which implies the convergence in probability of the individual fold predictions: $\hat{S}_{ij}^{(k)} \xrightarrow{p} \underline{\hat{S}}_{ij}$. Since the final prediction $\hat{Y}_{ij}$ is constructed via finite sums of these fold predictions, the Continuous Mapping Theorem yields $\hat{Y}_{ij} \xrightarrow{p} \underline{\hat{S}}_{ij}$.

Finally, all terms are bounded. The empirical covariance of bounded i.i.d. variables is a consistent estimator of the true covariance; substituting the converged predictions $\hat{Y}_{ij}$ preserves this. Since the deterministic sampling weights also converge to constants, Slutsky's Theorem gives $\hat{\Sigma}_{ij} \xrightarrow{p} \Sigma_{ij}$. □

E.5 Proof of Proposition A.7

Proof. We begin with $\hat{\theta}_i^{\lambda_i}$. From Corollary A.3, the asymptotic variance as a function of λ is

$$V_{\hat{\theta}_i}(\lambda_i) = \frac{1}{p_i}\mathrm{Var}(S_i) + \left(\frac{1}{p_i} - 1\right)\left[\lambda_i^2 \mathrm{Var}(\underline{\hat{S}}_i) - 2\lambda_i \mathrm{Cov}(S_i, \underline{\hat{S}}_i)\right].$$

Differentiating with respect to λ_i and setting it to zero gives the optimal weight:

$$\lambda_i^* = \frac{\mathrm{Cov}(S_i, \underline{\hat{S}}_i)}{\mathrm{Var}(\underline{\hat{S}}_i)}.$$

We now derive the optimal weight for $\hat{\Delta}_{ij}^{\lambda_{ij}}$, again using Corollary A.3. Let $\gamma_{ij} = \frac{p_{ij}}{p_i p_j} - 1$.

Proceeding as before:

$$\begin{aligned} V_{\hat{\Delta}_{ij}}(\lambda_i, \lambda_j) = &\frac{1}{p_i}\mathrm{Var}(S_i) + \gamma_{ii}\left[\lambda_i^2 \mathrm{Var}(\underline{\hat{S}}_i) - 2\lambda_i \mathrm{Cov}(S_i, \underline{\hat{S}}_i)\right] \\ &+ \frac{1}{p_j}\mathrm{Var}(S_j) + \gamma_{ij}\left[\lambda_j^2 \mathrm{Var}(\underline{\hat{S}}_j) - 2\lambda_j \mathrm{Cov}(S_j, \underline{\hat{S}}_j)\right] \\ &- 2\left[\frac{p_{ij}}{p_i p_j}\mathrm{Cov}(S_i, S_j) + \gamma_{ij}\left[\lambda_i\lambda_j \mathrm{Cov}(\underline{\hat{S}}_i, \underline{\hat{S}}_j) - \lambda_i \mathrm{Cov}(\underline{\hat{S}}_i, S_j) - \lambda_j \mathrm{Cov}(S_i, \underline{\hat{S}}_j)\right]\right] \end{aligned}$$

Consolidating terms, and writing $\mathrm{Var}(\hat{\Delta}_{ij}^{\text{classic}}) = \frac{1}{p_i}\mathrm{Var}(S_i) + \frac{1}{p_j}\mathrm{Var}(S_j) - 2\frac{p_{ij}}{p_i p_j}\mathrm{Cov}(S_i, S_j)$,

$$
\begin{aligned}
V_{\hat{\Delta}_{ij}}(\lambda_i, \lambda_j) &= \mathrm{Var}(\hat{\Delta}_{ij}^{\text{classic}}) \\
&\quad + \underbrace{\lambda_i^2 \gamma_{ii}\mathrm{Var}(\hat{\underline{S}}_i) + \lambda_j^2 \gamma_{jj}\mathrm{Var}(\hat{\underline{S}}_j) - 2\lambda_i\lambda_j\gamma_{ij}\mathrm{Cov}(\hat{\underline{S}}_i, \hat{\underline{S}}_j)}_{\lambda^\top Q \lambda} \\
&\quad - \underbrace{2[\lambda_i(\gamma_{ii}\mathrm{Cov}(S_i, \hat{\underline{S}}_i) - \gamma_{ij}\mathrm{Cov}(\hat{\underline{S}}_i, S_j)) + \lambda_j(\gamma_{jj}\mathrm{Cov}(S_j, \hat{\underline{S}}_j) - \gamma_{ij}\mathrm{Cov}(S_i, \hat{\underline{S}}_j))]}_{2\lambda^\top u}
\end{aligned}
$$

where Q and u are defined as in Proposition A.7.

To minimize $V(\lambda_i, \lambda_j)$, we again take the gradient with respect to λ and set it to zero:

$$\nabla_\lambda V(\lambda) = 2Q\lambda - 2u = 0,$$

which gives $\lambda^* = Q^{-1}u$. □

E.6 Proof of Proposition A.9

Proof. Slutsky's Theorem preserves convergence in distribution. The rewritten form of the asymptotic variance can then be seen from substituting $\lambda_i^* \hat{\underline{S}}_i$ directly into Equation (4) and simplifying. Following the derivations in the proof of Proposition A.7 in Section E.5,

$$
\begin{aligned}
V_{\hat{\theta}_i}(\lambda_i^*) &= \frac{1}{p_i}\mathrm{Var}(S_i) + \left(\frac{1}{p_i} - 1\right)\left[\frac{\mathrm{Cov}(S_i, \hat{\underline{S}}_i)^2}{\mathrm{Var}(\hat{\underline{S}}_i)^2}\mathrm{Var}(\hat{\underline{S}}_i) - 2\frac{\mathrm{Cov}(S_i, \hat{\underline{S}}_i)}{\mathrm{Var}(\hat{\underline{S}}_i)}\mathrm{Cov}(S_i, \hat{\underline{S}}_i)\right] \\
&= \mathrm{Var}(\hat{\theta}_i^{\text{classic}}) - \left(\frac{1}{p_i} - 1\right)\frac{\mathrm{Cov}(S_i, \hat{\underline{S}}_i)^2}{\mathrm{Var}(\hat{\underline{S}}_i)},
\end{aligned}
$$

and

$$
\begin{aligned}
V_{\hat{\Delta}_{ij}}(\lambda_i^*, \lambda_j^*) &= \mathrm{Var}(\hat{\Delta}_{ij}^{\text{classic}}) + (\lambda^*)^\top Q\lambda^* - 2(\lambda^*)^\top u \\
&= \mathrm{Var}(\hat{\Delta}_{ij}^{\text{classic}}) + (Q^{-1}u)^\top QQ^{-1}u - 2(Q^{-1}u)^\top u \\
&= \mathrm{Var}(\hat{\Delta}_{ij}^{\text{classic}}) - u^\top Q^{-1}u,
\end{aligned}
$$

since Q^{-1} is symmetric. □