Cost-Efficient Estimation of General Abilities Across Benchmarks

Michael Krumdick[1*] **Adam Wiemerslage**[1*] **Seth Ebner**[1] **Charles Lovering**[1]
Chris Tanner[1,2]
[1]Kensho Technologies [2]MIT
{michael.krumdick,chris.tanner}@kensho.com

Abstract

Thousands of diverse benchmarks have been developed to measure the quality of large language models (LLMs). Yet prior work has demonstrated that LLM performance is often sufficiently explained by a small set of latent factors, or *abilities*. This suggests the potential for more efficient and principled benchmarking, but it remains difficult to compare the quality of different methods. Motivated by predictive validity, we argue that the quality of a benchmarking framework should be grounded in how efficiently it enables the prediction of model performance on unseen tasks. To analyze this objective, we collect the "Wide-scale Item Level Dataset" (WILD), a dataset of item-model response pairs, comprising evaluations of 65 models on 109,564 unique items spanning 163 tasks drawn from 27 datasets. This dataset enables the first analysis of how different techniques can predict a model's performance on a large, diverse collection of unseen tasks under different budget constraints. We demonstrate that combining a modified multidimensional item response theory (IRT) model with adaptive item selection driven by optimal experimental design can predict performance on 112 held-out benchmark tasks with a mean absolute error (MAE) of less than 7% after observing only 16 items. We further demonstrate that incorporating cost-aware discount factors into our selection criteria can reduce the total tokens needed to reach 7% MAE from 141,000 tokens to only 22,000, an 85% reduction in evaluation cost.

1 Introduction

A surge of research in LLM evaluation has resulted in thousands of benchmarks varied in domain and format, from grade-school word problems (Cobbe et al., 2021) to completing code snippets (Zhuo et al., 2025) to answering multiple choice questions on foreign policy (Hendrycks et al., 2021a). Despite the diversity, previous work has empirically demonstrated that the underlying "capability space" of language models is low-rank (Ruan et al., 2024; Burnell et al., 2023; Ilić & Gignac, 2024), meaning that many of these evaluations are measuring redundant capabilities. This manifests in high correlations between tasks (Ye et al., 2023), creating the opportunity for developing "compressed" or efficient benchmarking techniques that leverage these relationships to reduce the total number of samples required to estimate performance (Perlitz et al., 2024; Kipnis et al., 2025; Maia Polo et al., 2024).

Task-oriented evaluations attempt to estimate how well a model performs on a specific task or class of tasks. For example, one may want to estimate the probability that a model generates a correct pull request on real-world GitHub issues (Jimenez et al., 2024). In contrast, an *ability-oriented* view of evaluation focuses on understanding how LLMs will perform across a broad range of (potentially nascent) tasks. For example, when benchmarking an LLM on MMLU (Hendrycks et al., 2021a), the goal is often not to assess performance on multiple-choice questions. Instead, two key assumptions are made: (i) Answering these questions requires some set of task-independent abilities, and (ii) Performance on these questions

*Equal Contribution.

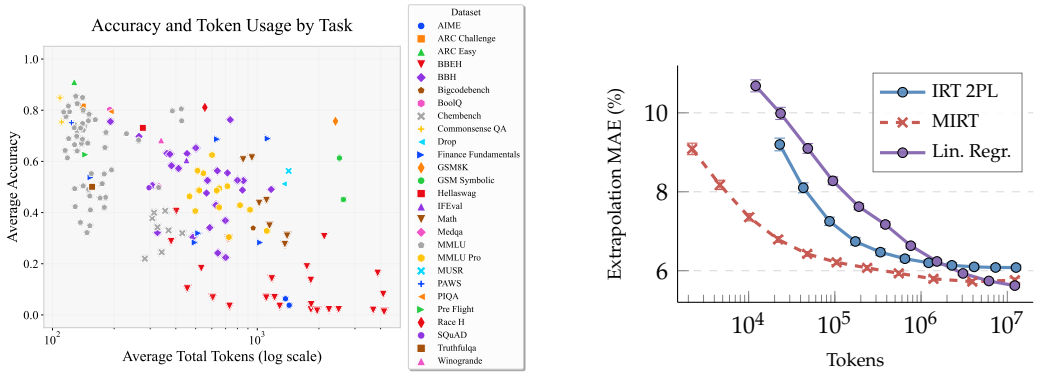

Figure 1: **Left**: Token usage and accuracy averaged across all LLMs for the tasks in WILD. Tasks vary in token usage, leading to radically different item costs. **Right**: MAE for predicting the sample mean over 112 held-out tasks. The MIRT model with Optimal Experimental Design selection achieves low MAE for small sample size and token cost.

measures those abilities *and gives us information about how the LLM might perform on other tasks that require them* (Hernández-Orallo, 2017).

While the quality of task-oriented evaluations can often be assessed by measuring how well they correlate with the desired behavior (e.g. how well does SWEBench predict a model's ability to close pull requests in practice), ability-oriented evaluations do not have a clear success metric. How, then, can one determine whether one set of evaluations is better than another? Psychometrics provides the framework of *predictive validity*: the quality of an ability-oriented evaluation is determined by how well it predicts performance on other related tasks (Schellaert & Martínez-Plumed, 2025).

In this work, we develop a framework centered on predictive validity that evaluates the quality of model ability estimates. Prior work that uses predictive validity typically considers hand-curated task pairs. In contrast, we focus on efficiently predicting model performance across more than 100 held-out tasks. To enable this scale, we build the Wide-scale Item Level Dataset (WILD): a large-scale dataset comprising 109,564 items from 163 distinct tasks derived from 27 benchmarks, with responses from 65 LLMs for every item.

Dataset compression techniques often measure cost in terms of the number of items in the compressed dataset. While this may be reasonable for items from a single dataset, items across datasets can have radically different token-costs (Figure 1). The average item from the "buggy tables" scenario in Big Bench Extra Hard (Kazemi et al., 2025) compared to a "world religions" question from MMLU is 60 times more expensive to evaluate with `gpt-4o`. This observation motivates our primary focus: which methods can best predict performance across a large number of held-out tasks *for the lowest total token cost*?

Our main contributions are:

1. We release WILD, a wide-scale item-level dataset of model responses consisting of 65 models evaluated on 109,564 unique items from 163 tasks.

2. We formulate a benchmark method comparison framework based on predictive validity, evaluating assessors on predicting model performance on over 100 benchmarks at a time.

3. We demonstrate that multidimensional IRT achieves <7% MAE on held-out tasks after observing only 16 items, and < 6% after observing 128 items, outperforming unidimensional IRT and simple baselines.

4. We introduce a cost-aware item-selection criterion that reduces evaluation token cost by over 80% while maintaining prediction quality.

2 Related Work

Psychometrics in NLP There is a burgeoning field of applying techniques from psychometrics to improve the evaluation of LLMs. Rodriguez et al. (2021) showed that IRT models rank LLMs more reliably than aggregate accuracy does, and that they can be used to select informative evaluation items. IRT models have been used more broadly to quantify the difficulty, discrimination, or other aspects of items and datasets (Yao et al., 2025; Vania et al., 2021; Rodriguez et al., 2022). Many works have tried to analyze the underlying structure of model performance using PCA (Ruan et al., 2024), matrix factorization (Burnell et al., 2023; Ilić & Gignac, 2024), performance predictability (Ye et al., 2023), and human-informed rubrics of capability (Hendrycks et al., 2025; Zhou et al., 2025). Each of these works suggests that model performance is largely explained by a small number of factors. Our method combines a multidimensional IRT model with optimal experimental design guided item selection. Although these ideas have been explored for human testing (Mulder & Linden, 2009; van der Linden, 1999), they have yet to be applied to LLM evaluation.

Efficient Evaluation Understanding an LLM's underlying abilities can yield practical benefits, such as reducing the cost of evaluations using metrics to evaluate subset quality (Perlitz et al., 2024), or using clustering to select informative items (Vivek et al., 2024). Towards this goal, many works explicitly use techniques from psychometrics. Maia Polo et al. (2024) propose tinyBenchmarks, which (1) adapts the clustering-based selection method anchor points to binary response patterns, and (2) develops IRT models to compress benchmarks, which are then used to estimate LLM abilities. Kipnis et al. (2025) fit IRT models on HELM (Liang et al., 2023) and use Fisher information-guided selection to create compressed versions of the benchmark. Hofmann et al. (2025) and Li et al. (2025) use IRT models to create evaluation frameworks that use Fisher information to adaptively select the most informative items for a particular model, based on the model's previous responses. In contrast, other work has highlighted some of the limitations of psychometric techniques for efficient evaluation of LLMs (Madaan et al., 2025).

Ability-Oriented Benchmarking Prior works have unified measures from different evaluations into estimates of general abilities. In computer vision, Prabhu et al. (2024) introduce a method for creating an ever-growing battery of both items and models. ONEBench (Ghosh et al., 2025) builds an embedding-based retrieval system to aggregate items across benchmarks that target particular capability keywords. Zhuang et al. (2025) introduce EmbedLLM, a method for learning descriptive embeddings for models inspired by matrix factorization. Most similar to our work is that of Liao et al. (2025), who utilize Bayesian MAP estimation and a structural prior to estimate latent abilities across benchmarks. However, these approaches either focus on task-specific compression or do not evaluate cross-task predictive validity with more than a handful of tasks.

3 Preliminaries

In this section, we provide a brief overview of the main IRT models used in this work, along with the optimal experimental design criteria used for selection.

3.1 Item Response Theory

A *record* is a tuple consisting of a model, item, and response: (m, x, r). In our work, every response r is represented as a binary label, 0 or 1, that indicates the correctness of the model's prediction. Given a dataset of records, $\mathcal{R} = \{(m_0, x_0, r_0), \dots\}$, we would like to learn an *assessor* that can predict if an LLM m will produce a correct response for a given item x. The assessor's prediction is defined as $p(r = 1|m, x)$. Psychometrics provides a family of models, originally developed for human testing, suited for this purpose. Specifically, IRT uses both the test-taker's ability $\theta_m \in \mathbb{R}$ and the item's difficulty $\delta_x \in \mathbb{R}$ to model the ability of a test-taker to correctly answer an item:

$$p(r = 1|m, x) = \sigma(\alpha_x(\theta_m - \delta_x)) = \frac{1}{1 + e^{-\alpha_x(\theta_m - \delta_x)}}.$$

The above model, referred to as a two-parameter logistic (2PL) model, has scalar parameters that capture both an item's difficulty δ_x and its discrimination α_x. A one-parameter logistic (1PL) model fixes $\alpha_x = 1$.

We can extend the above framework to a Multidimensional IRT Model (MIRT): rather than defining ability as a single scalar value θ_m, one can represent model m's abilities as a vector $\Theta_m \in \mathbb{R}^d$, where d is the number of abilities. Each item x will have a *loading* vector $K_x \in \mathbb{R}^d$ that weights the individual Θ_m dimensions. The model's ability for item x is defined as:

$$p(r = 1|m, x) = \sigma(\alpha_x(\langle \Theta_m, K_x \rangle - \delta_x)). \tag{1}$$

3.2 Item Selection

IRT models can *adaptively* select items for assessment based on Fisher information, $\mathcal{I}$. The Cramer-Rao bound states that for an unbiased estimator, the variance is at least $1/\mathcal{I}(\theta_m)$:

$$\text{Var}(\hat{\theta}_m) \geq \frac{1}{\mathcal{I}(\theta_m)}. \tag{2}$$

Fisher information is additive over independent samples. For the 2PL model:

$$\mathcal{I}_{m,x} = \alpha_x^2 \sigma(\alpha_x(\theta_m - \delta_x))(1 - \sigma(\alpha_x(\theta_m - \delta_x))). \tag{3}$$

This quantity is maximized when $p(r = 1|m, x) = 0.5$. Adaptive item selection proceeds by choosing items with the highest Fisher information for the current ability estimate, which equivalently corresponds to the item with the highest expected reduction in variance.

Instead of a single Fisher information value, the multidimensional case requires analysis of a Fisher information *matrix*. Given a loading vector K_x, we modify Equation (3) to the multidimensional case to get:

$$p = \sigma(\alpha_x(\Theta_m^T K_x - \delta_x)), \qquad \mathcal{I}_{m,x} = \alpha_x^2 p(1-p) K_x K_x^T. \tag{4}$$

When $|K| > 1$, we have a Fisher information matrix. The diagonals of this matrix represent the marginal information an item carries for $\Theta_{m,i}$ if its response is observed. The off-diagonals represent the joint information between $\Theta_{m,i}$ and $\Theta_{m,j}$. Selecting items with the maximal trace for this matrix is analogous to maximizing the Fisher information. However, while maximizing the Fisher information is equivalent to minimizing the variance of $\hat{\theta}$ in the single-dimensional case, this is not true with a MIRT model, where the variance of $\hat{\Theta}$ depends on the full matrix structure. To address this limitation, we borrow from the Optimal Experimental Design (OED) literature (Kiefer & Wolfowitz, 1959) to develop selection objectives that model the variance of a multidimensional $\hat{\Theta}$.

Optimal Experimental Design OED criteria typically model the inverse of the accumulated Fisher information after observing an item:

$$(\mathcal{I}_{\text{cumulative}} + \mathcal{I}_{m,x})^{-1}. \tag{5}$$

where $\mathcal{I}_{\text{cumulative}}$ is the current cumulative sum of the information contributed from each item. Given the Cramer-Rao bound in Equation (2), this acts as a lower bound on the estimator variance. Many popular OED criteria focus on minimizing a function of the information matrix parameters. We found that V-optimality, which minimizes the prediction variance over observed points, works best in preliminary experiments. Concretely, we define the *prediction matrix* $M = \sum_{x \in \mathcal{X}_t} \alpha_x^2 K_x K_x^\top$, where $\mathcal{X}_t$ is the set of items belonging to target benchmarks, whose parameters are known from training but whose responses for m are to be predicted. Given the current posterior covariance $C = \mathcal{I}_{\text{cumulative}}^{-1}$, the V-optimal selection score for candidate item x is:

$$s_x = \frac{w_x \alpha_x^2 \, c_x^\top M c_x}{1 + w_x \alpha_x^2 \, K_x^\top c_x}, \quad c_x = C K_x, \quad w_x = p_x(1 - p_x). \tag{6}$$

After each observation, the posterior covariance is updated via the Woodbury correction (Appendix A.2). V-Optimality thus prioritizes items whose loadings align with M, which we expect to prioritize "general" factors that are informative for predictive ability across tasks.

4 WILD Dataset

To support the ability to estimate LLM performance across diverse tasks, we needed a large dataset of model results on many tasks. Existing item-level response datasets contain limited individual tasks or have other quality issues. For example, the OpenLLM Leaderboard (Fourrier et al., 2024) contains thousands of different models, but most are adapted from a small set of base models. HELM (Liang et al., 2023) contains many tasks but lacks item coverage for many models.

Table 1: Comparison of WILD to other item-level response datasets.

	Models	Tasks	Items
Perlitz et al. (2024)	44	16	65,000
Kipnis et al. (2025)	5055*	63	28,632
Zhou et al. (2025)	15	63	16,108
Liao et al. (2025)	70	5	8,918
WILD	65	163	109,564

* Includes model variants.

WILD represents the largest complete cross-task evaluation dataset where all model-task pairs were evaluated using a single consistent framework. Table 1 compares this dataset to other item-level response datasets.[1]

We collect 109,564 items across 163 tasks from 27 benchmarks. We focused on high quality tasks with compatible implementations (Abbas et al., 2025), including: multiple choice (Wang et al., 2024, MMLU Pro), math (Hendrycks et al., 2021b, MATH), coding (Zhuo et al., 2025, bigcodebench), and domain-specific (Krumdick et al., 2024, BizBench) evaluations.

WILD includes models from 0.5B parameters to large, closed API models. For our main analysis, we exclude explicit reasoning models, motivated by the increased costs and additional hyperparameter complexities introduced by this family of models (reasoning effort, number of reasoning tokens, etc.). In Appendix D, we provide a preliminary analysis that suggests our results apply to reasoning models; however, we leave a larger investigation for future work. We evaluate all models using the Inspect AI framework (AI Security Institute, 2024). In total, we evaluated 65 unique models, sourced from 26 different organizations. A full list of tasks and models included in WILD can be found in Appendix J.

5 Methods

Our primary method for cross-task ability estimation modifies a bifactor model (Holzinger & Swineford, 1937). We describe the model in Section 5.1 and the fitting procedure in Section 5.2. Finally, in Section 5.3, we incorporate token costs into the selection methods.

5.1 Nested MIRT Model

We modify the standard bifactor model—where each item loads on a general factor and exactly one task specific factor—by replacing the task specific factors with latent factors, akin to the MIRT model,

$$p(r = 1|m, x) = \sigma(\alpha_x(\Theta_{m,g} + \langle \Theta_m, K_x \rangle - \delta_x)). \tag{7}$$

This ensures that a single factor remains active for all items, thereby creating a shared general factor. The utility of such a general factor in ability estimation is well-established in the literature (Spearman, 1904), and we confirm this in a preliminary factor analysis in Figure 5 in the appendix. To emphasize the importance of the general factor, we optimize a regularized log-likelihood that encourages shrinkage toward a nested unidimensional model, improving stability in low-sample regimes. Let $\ell(r, z)$ denote the Bernoulli log-likelihood with logit z,

$$\ell(r, z) = r \log \sigma(z) + (1 - r) \log(1 - \sigma(z)). \tag{8}$$

We extract two logits from the model and combine both for our loss:

$$z_{\text{full}} = \alpha_x(\Theta_{m,g} + \langle \Theta_m, K_x \rangle - \delta_x), \quad z_g = \alpha_x(\Theta_{m,g} - \delta_x), \quad \mathcal{L}_{\text{total}} = \ell(r, z_{\text{full}}) + \lambda\, \ell(r, z_g).$$

We include an ablation and analysis of these modifications in Appendix E, and refer to this implementation as MIRT in the remainder of the paper.

[1] WILD is available at https://huggingface.co/datasets/kensho/WILD

5.2 Fitting the MIRT Model

Our fitting procedure uses coordinate gradient descent with L-BFGS-B (Byrd et al., 1995) on the model and item parameters. We include a standard normal prior on Θ. We use this fitted MIRT model to efficiently form estimates of held-out LLMs' Θ via maximum a posteriori (MAP) estimation on the frozen item parameters from the available records. In order to learn stable updates from very small numbers of items, we include an empirical prior in the MAP update, $\Theta_m \sim \mathcal{N}(\mu_\Theta, \Sigma)$, where μ_Θ is the mean of the observed Θ parameters and Σ is the covariance matrix between the dimensions of Θ. We then maximize

$$\log p(\Theta_m | x, r) \propto \mathcal{L}_{\text{prior}}(\Theta_m) + \log p(r | m, x). \tag{9}$$

More details on the training and MAP process can be found in Appendix A.

5.3 Selecting Items with Token-Aware Discounting

Prior work explores efficient evaluation with the goal of reducing the number of samples an LLM must respond to, effectively considering the cost of all items to be equal. However, items often vary greatly in cost. Figure 1 demonstrates the variation in total tokens across tasks in WILD. This translates directly into compute and API cost. We thus incorporate a cost term to encourage selection of cheap items:

$$\overline{\tau}_x = \frac{1}{|\mathcal{M}_{train}|} \sum_{m \in \mathcal{M}_{train}} \tau(m, x), \tag{10}$$

where $\mathcal{M}_{train}$ is the set of all models used for training, and $\tau(\cdot)$ counts the sum of input tokens encoded by and output tokens generated by m for x. We can then modify any information based selector to get a ratio between (a) the information gain from observing an item, and (b) the item's expected cost for an unknown model: $\mathcal{I}_x / \overline{\tau}_x$. Intuitively, this identifies items with the best tradeoff between informativeness and cost. We denote cost-aware selectors with the -τ suffix.

6 Experimental Setup

We formulate our framework as item-level prediction over a dataset with a large number of records across a series of models and tasks. The item-level prediction models, or *assessors*, aim to infer how a model will perform on a set of held-out items given its observed performance on a separate set of items. We consider two types of prediction: **Interpolation** predicts performance on held-out items from the *observable tasks*. **Extrapolation** predicts performance on held-out items from a *different set of tasks*.

6.1 Splits

We split the data at two levels. First, models are partitioned into train and test sets. For the train models, the assessor is allowed to observe performance on all items across all tasks. In our experiments, we perform cross-validation over 20 random model partitions, each consisting of 55 train models and 10 test models. We additionally randomly partition tasks into interpolation and extrapolation sets. The assessor observes the performance of each test model on items from the interpolation tasks. The assessor then predicts performance on held-out items from those same tasks (interpolation), as well as performance on items from unseen tasks (extrapolation). For each task, we hold out 128 items and filter out tasks with fewer than 128 items for computing prediction metrics.

More formally, the **calibration** records consist of a set of test models $\mathcal{M}_{test}$ for evaluation (all of which are previously unobserved) with a subset of historic tasks, $\mathcal{T}^c \subset \mathcal{T}$, containing items $\mathcal{X}^c$. The **validation** records $\mathcal{X}^v$ are also defined over $\mathcal{M}_{test}$, but they comprise a set of extrapolation tasks $\mathcal{T}^v \subset \mathcal{T}$ that are disjoint from the interpolation tasks: $\mathcal{T}^v \cap \mathcal{T}^c = \emptyset$. See Table 4 in the Appendix for an overview of these splits.

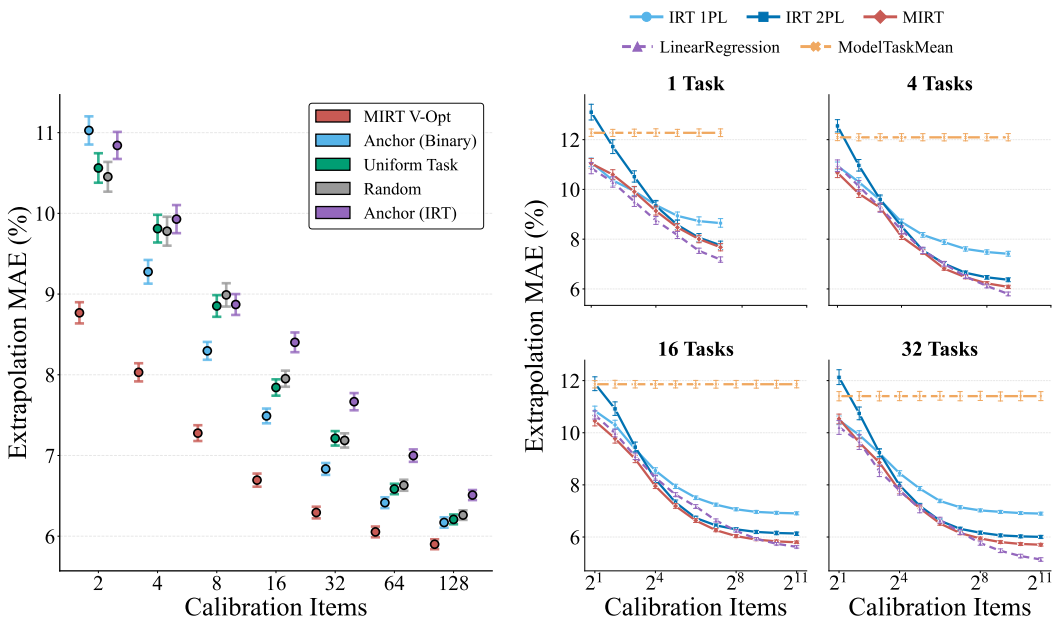

Figure 2: **Left**: Baseline selectors vs. Adaptive V-Optimal. **Right**: Prediction comparison. MIRT v.s. baseline predictors as we increase the number of extrapolation tasks. Model Task Mean (MTM) baselines are constant because extrapolation tasks are fully held out for the test models. See Figure 4 for interpolation results.

6.2 Evaluation Paradigms

We evaluate how well the assessor predicts an unseen model's performance when provided only N samples from the calibration records. We perform three types of evaluation based on how the sampled calibration records are provided to the assessor.

Prediction The assessor receives a random sample of N items from the calibration records. This evaluates how well the assessor predicts model performance from a static benchmark.

Selection The assessor selects the N items from the calibration records all at once. This evaluates how well the assessor can compose static benchmarks out of a pool of items.

Adaptation The assessor iteratively selects N items for each calibration model, conditioned on how the model has performed on previously selected items. This evaluates how well the assessor can create adaptive benchmarks for a given model.

For each task, we hold out a set of items for computing the evaluation metrics. Model performance on these held-out items can never be observed by the assessor. This is a subtle distinction with prior work which evaluates dataset compression performance by how well the model predicts the dataset performance mean from items that were used to compute that mean. Importantly, assessors within such a framework will always reach perfect performance once they observe all items. In contrast, our evaluation involves some amount of irreducible error. We provide further discussion in Appendix I. This makes exactly one interpolation setting in our experiments comparable to prior work on benchmark compression. We present that setting in isolation in Table 2 in the appendix.

6.3 Metrics

Although we require assessors to predict item-level performance, our main performance results evaluate aggregate accuracy over tasks. Item-level metrics may better capture how well the assessor can predict how a model will perform, but task-level metrics are generally more interpretable and align with how most practitioners interact with benchmarks.

Task-level MAE measures how well the predictor captures task-level performance:

$$\mathrm{MAE}_{\mathrm{task}}(a, \mathcal{R}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} |\mu_r(\mathcal{R}_t) - \mu_a(\mathcal{R}_t)|, \tag{11}$$

averaged across all tasks $\mathcal{T}$, where $\mu_r(\mathcal{R}_t)$ is the empirical mean over the records and $\mu_a(\mathcal{R}_t)$ is the mean of the assessor prediction of $p(r = 1)$ over the records.

Token Cost measures the cost of the N samples, computed as the sum of the input and output tokens across all items. In practice, because encoding costs are cheaper than generation costs in LLMs, we halve the cost of input tokens.

6.4 Prediction Baselines

Sample Mean This is the simplest baseline predictor. From a set of records, we estimate:

$$\hat{\theta}_{m,t} = \mu_r(\mathcal{R}_t) = \frac{1}{|\mathcal{R}_t|} \sum_{\langle m,x,r \rangle \in \mathcal{R}_t} \mathbb{I}(r = 1). \tag{12}$$

We implement a simple predictor, $p_{\mathrm{mean}}(r = 1|m, x_t) = \hat{\theta}_{m,t}$, which predicts performance for an unseen item based on the model sample mean for that task. However, when $|\mathcal{R}_t|$ is small, sample mean estimation may be poor. We develop a Bayesian extension using the training data as a prior and use Bayesian shrinkage on our estimate. For more details on the Bayesian estimate and results for sample mean baselines, see Appendix B.

Linear Regression We fit a ridge regression model for each task from the vector of responses for $\mathcal{M}_{train}$ on observed items to the overall mean performance on that task. Then we apply each task-specific model to $\mathcal{M}_{test}$. We train a new model for every set of calibration item observations. This is a strong and somewhat privileged baseline: it explicitly trains a new model to predict each target task, whereas IRT models learn a task-agnostic latent ability space with no knowledge of which tasks will be evaluated at test time, making the large set of evaluation tasks harder for the IRT models.

2PL IRT We include a single dimensional, standard IRT model trained across all items. We fit this model using the same method as the MIRT model. With Fisher selection, this matches the setup used by Hofmann et al. (2025) in the Adaptation setting. In the Selection setting, it matches the method from Kipnis et al. (2025). However, since our model universe is smaller than Kipnis et al. (2025), we use a single median θ rather than binning across all possible θ.

6.5 Selection Baselines

Selectors are responsible for choosing items for the **Selection** and **Adaptation** evaluations. We include some simple selection baselines. **Random** selects randomly from the pool of items. **Uniform** evenly allocates items over each task, randomly selecting tasks to receive the remainders in the likely case that the number of items is not divisible by the number of tasks. To test efficiency, **Min Cost** (See Figure 6 for results) selects the items with the lowest expected cost based on the estimator described in Section 5.3.

Anchor Points We compare with the anchor points (Vivek et al., 2024) implementations described in Maia Polo et al. (2024), where items are clustered with k-means and points close to the centroids are selected. We compare to two of their implementations that differ in the item representation: **Anchor-binary** represents each item as the vector of all of its historic model responses. **Anchor-IRT** instead represents items as the concatenation of their loading and difficulty parameters from the IRT predictor.

7 Results & Analysis

For all results we compare to the Nested MIRT model with K=3. This choice of K is based on a factor analysis presented in Appendix B.1. See Appendix E for full post-hoc ablations of K.

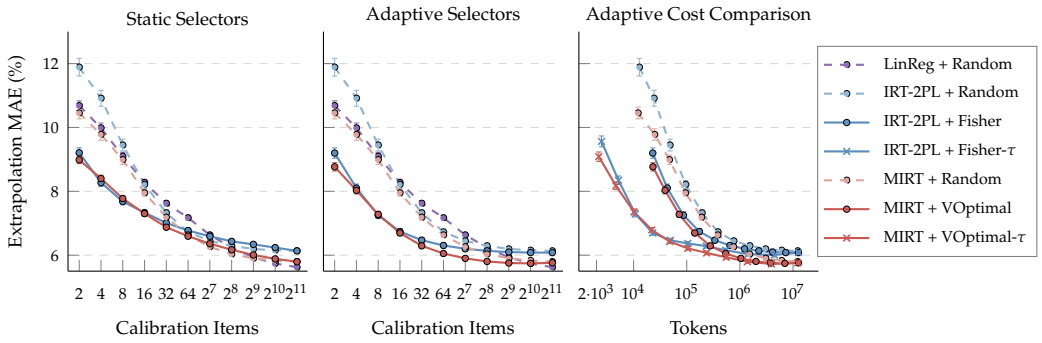

Figure 3: Summary of main results comparing the performance of the Nested MIRT and 2PL IRT model. (Left) Comparison in the static selection setting. (Center) Comparison in the adaptive selection setting. (Right) Comparison of token efficiency in the adaptive setting. In all three plots, Random selection corresponds to the Prediction setting (Section 6.2).

7.1 Prediction

Figure 2 compares the extrapolation performance of MIRT to all baseline predictors on the right. We vary the number of calibration tasks as a proxy for item diversity. Model performance tends to increase as the number of tasks increases even as the samples per task decreases. This is especially consistent for the MIRT model: as the number of tasks and samples increases, the improvements of MIRT over the IRT-2PL predictor also increases. This provides an interesting scaling direction. For a fixed number of samples, selecting from a larger set of tasks increases the efficiency of estimating LLM abilities with MIRT. The regression model performs best in the single task case, and when there are large numbers of calibration items. This highlights the IRT model's ability to generalize latent capability representations from other tasks. It also demonstrates that at a high enough number of calibration samples, the regression assessor, which fits a new model to predict each individual task and calibration item set, is unsurprisingly extremely well-calibrated. The IRT models may suffer from noise in the difficult cross-task generalization that they must solve. However, we show in Section 7.2 that sampling high-information items under the IRT model solves this issue, leading to better performance than the regression model.

In Appendix B we present interpolation results, where the differences between IRT models are much smaller than in the extrapolation setting. Interpolation is similar to prior work that focused on compressing a given benchmark, particularly at the 1-task setting, which highlights the difficulty of extrapolation, where we see more variance in the results.

7.2 Selection

In Figure 3 (left) we compare MIRT, IRT, and regression assessors with both random and information-based selectors. At low numbers of calibration items, the Fisher selector for the IRT model reduces MAE significantly compared to random selection, and performs on-par with MIRT V-Optimal until 32 calibration items. The V-Optimal selector for MIRT achieves consistently strong performance compared to other methods until the regression baseline surpasses it at 1024 (2^{10}) samples.

7.3 Adaptation

Figure 3 (center) presents results for the adaptive setting, where items are selected one at a time per LLM, adapting to the current ability estimate. At all but the maximum calibration item budgets, adaptive selection improves results for V-Optimal and Fisher selectors, with the adaptive MIRT V-Optimal assessor outperforming the IRT-2PL Fisher assessor by a larger gap (see Figure 10 for a direct comparison). In Figure 3 (right), we investigate the token efficiency of each method. Here, the x-axis is the total number of tokens after sampling N calibration items at the same intervals as the other two plots. Token counts are averaged

over all seeds and splits. For both the MIRT and IRT models, token-aware discounting leads to significant savings, reducing MAE to <7% with fewer than 50,000 tokens versus almost 400,000 tokens with random selection. Figure 2 (left) compares the performance of various baseline selectors for the MIRT model. Adaptive V-Optimal consistently outperforms the baselines. Once $N \approx 64$, each of the baseline selection methods performs roughly on par with random selection.

A natural question given the token-efficient results is: do we actually need to select by Fisher information if we can simply select the cheapest possible items? In Appendix C, we demonstrate that the cheapest items in terms of tokens do not lead to efficient results, and the Fisher information based selection is crucial.

8 Conclusion

We demonstrate that for large numbers of held-out tasks, IRT methods often greatly improve task-level prediction accuracy, in comparison to several strong baselines. Our approach is motivated by and leverages both psychometric literature (Lord et al., 1968) and Optimal Experimental Design (Kiefer & Wolfowitz, 1959). We introduce assessor models that are effective and cost-efficient not only in the base prediction task—predicting LLM performance on held-out items—but also in selecting assessment items out of a pool of benchmarks. A key contribution of this work is a novel cost-aware term which leads to high quality model ability estimates with an order-of-magnitude reduction in token costs.

LLM Use Disclosure

Claude Code was used extensively to write several parts of the experimental framework, and to help with code for generating plots.

References

Alexandra Abbas, Celia Waggoner, and Justin Olive. Developing and maintaining an open-source repository of AI evaluations: Challenges and insights. In *Championing Open-source DEvelopment in ML Workshop @ ICML25*, 2025. URL https://openreview.net/forum?id=yw33GWAEOK.

UK AI Security Institute. Inspect AI: Framework for Large Language Model Evaluations, 2024. URL https://github.com/UKGovernmentBEIS/inspect_ai.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez Orallo. Revealing the structure of language model capabilities, 2023. URL https://arxiv.org/abs/2306.10062.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2

reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

Adhiraj Ghosh, Sebastian Dziadzio, Ameya Prabhu, Vishaal Udandarao, Samuel Albanie, and Matthias Bethge. ONEBench to test them all: Sample-level benchmarking over open-ended capabilities. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32445–32481, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1560. URL https://aclanthology.org/2025.acl-long.1560/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Dan Hendrycks, Dawn Song, Christian Szegedy, Honglak Lee, Yarin Gal, Erik Brynjolfsson, Sharon Li, Andy Zou, Lionel Levine, Bo Han, Jie Fu, Ziwei Liu, Jinwoo Shin, Kimin Lee, Mantas Mazeika, Long Phan, George Ingebretsen, Adam Khoja, Cihang Xie, Olawale Salaudeen, Matthias Hein, Kevin Zhao, Alexander Pan, David Duvenaud, Bo Li, Steve Omohundro, Gabriel Alfour, Max Tegmark, Kevin McGrew, Gary Marcus, Jaan Tallinn, Eric Schmidt, and Yoshua Bengio. A definition of agi, 2025. URL https://arxiv.org/abs/2510.18212.

José Hernández-Orallo. *The measure of all minds: evaluating natural and artificial intelligence*. Cambridge University Press, 2017.

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. URL https://openreview.net/forum?id=mxcCg9YRqj.

Karl J Holzinger and Frances Swineford. The bi-factor method. *Psychometrika*, 2(1):41–54, 1937.

David Ilić and Gilles E. Gignac. Evidence of interrelated cognitive-like capabilities in large language models: Indications of artificial general intelligence or achievement? *Intelligence*, 106:101858, September 2024. ISSN 0160-2896. doi: 10.1016/j.intell.2024.101858. URL http://dx.doi.org/10.1016/j.intell.2024.101858.

Inspect Evals. Pre-flight: Aviation operations knowledge evaluation, 2025. URL https://ukgovernmentbeis.github.io/inspect_evals/evals/knowledge/pre_flight/. Dataset created by Airside Labs. Documentation hosted as part of the Inspect Evals project. Accessed: 2026-01-29.

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.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14), 2021. ISSN 2076-3417. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Peter Chen, Nishanth Dikkala, Gladys Tyen, Xin Liu, Uri Shalit, Silvia Chiappa, Kate Olszewska, Yi Tay, Vinh Q. Tran, Quoc V Le, and Orhan Firat. BIG-bench extra hard. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26473–26501, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1285. URL https://aclanthology.org/2025.acl-long.1285/.

Jack Kiefer and Jacob Wolfowitz. Optimum designs in regression problems. *The annals of mathematical statistics*, 30(2):271–294, 1959.

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.

Michael Krumdick, Rik Koncel-Kedziorski, Viet Dac Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. BizBench: A quantitative reasoning benchmark for business and finance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8309–8332, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.452. URL https://aclanthology.org/2024.acl-long.452/.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Peiyu Li, Xiuxiu Tang, Si Chen, Ying Cheng, Ronald Metoyer, Ting Hua, and Nitesh V Chawla. Adaptive testing for llm evaluation: A psychometric alternative to static benchmarks. *arXiv preprint arXiv:2511.04689*, 2025.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=iO4LZibEqW. Featured Certification, Expert Certification, Outstanding Certification.

Lele Liao, Qile Zhang, Ruofan Wu, and Guanhua Fang. Toward a unified framework for data-efficient evaluation of large language models. *arXiv preprint arXiv:2510.04051*, 2025.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Iryna Gurevych, Michael Strube, and Ryan Cotterell (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3231, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org.

FM Lord, MR Novick, and Allan Birnbaum. *Statistical theories of mental test scores.* Addison-Wesley, 1968.

Lovish Madaan, Aaditya K Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. In *NeurIPS 2024 Workshop on Regulatable ML*, 2025. URL https://openreview.net/forum?id=M9dCa4vYgp.

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.

Mathematical Association of America. American invitational mathematics examination (aime), 1983. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime. Annual mathematics competition. Problems used for evaluation via the Inspect Evals framework. Accessed: 2026-01-29.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Amir Mohammad Elahi, Mehrdad Asgari, Juliane Eberhardt, Hani M. Elbeheiry, María Victoria Gil, Maximilian Greiner, Caroline T. Holick, Christina Glaubitz, Tim Hoffmann, Abdelrahman Ibrahim, Lea C. Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole Roesner, Johanna Schreiber, Ulrich S. Schubert, Leanne M. Stafast, Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. Are large language models superhuman chemists?, 2024. URL https://arxiv.org/abs/2404.01475.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AjXkRZIvjB.

Joris Mulder and Wim Linden. Multidimensional adaptive testing with optimal design criteria for item selection. *Psychometrika*, 74:273–296, 06 2009. doi: 10.1007/s11336-008-9097-5.

Yotam Perlitz, Elron Bandel, Ariel Gera, Ofir Arviv, Liat Ein-Dor, Eyal Shnarch, Noam Slonim, Michal Shmueli-Scheuer, and Leshem Choshen. Efficient benchmarking (of language models). In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2519–2536, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.139. URL https://aclanthology.org/2024.naacl-long.139/.

Ameya Prabhu, Vishaal Udandarao, Philip Torr, Matthias Bethge, Adel Bibi, and Samuel Albanie. Efficient lifelong model evaluation in an era of rapid progress. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=A7wC1CTkYl.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

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/.

Pedro Rodriguez, Phu Mon Htut, John Lalor, and João Sedoc. Clustering examples in multi-dataset benchmarks with item response theory. In Shabnam Tafreshi, João Sedoc, Anna Rogers, Aleksandr Drozd, Anna Rumshisky, and Arjun Akula (eds.), *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pp. 100–112, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.insights-1.14. URL https://aclanthology.org/2022.insights-1.14/.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. 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. 15841–15892. Curran Associates, Inc., 2024. doi: 10.52202/079017-0506. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1cded4f97cf5f01a284c574110b7e3b9-Paper-Conference.pdf.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Wout Schellaert and Fernando Martínez-Plumed. *The evaluation of artificial intelligence as a Prediction Problem*. PhD thesis, PhD thesis, Universitat Politecnica de Valencia, 2025.

C. Spearman. *General Intelligence Objectively Determined and Measured*. American journal of psychology. University of Illinois Press, 1904.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=jenyYQzue1.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Wim J. van der Linden. Multidimensional adaptive testing with a minimum error-variance criterion. *Journal of Educational and Behavioral Statistics*, 24(4):398–412, 1999. ISSN 10769986, 19351054. URL http://www.jstor.org/stable/1165370.

Clara Vania, Phu Mon Htut, William Huang, Dhara Mungra, Richard Yuanzhe Pang, Jason Phang, Haokun Liu, Kyunghyun Cho, and Samuel R. Bowman. Comparing test sets with item response theory. 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. 1141–1158, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.92. URL https://aclanthology.org/2021.acl-long.92/.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17 (3):261–272, 2020.

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/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. 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. 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

Jihan Yao, Peter Jin, Ke Bao, Qiaolin Yu, Khushi Bhardwaj, Chang Su, Jialei Wang, YIKAI ZHU, Sugam Devare, Damon Mosk-Aoyama, Zhen Dong, Venkat Krishna Srinivasan, Yineng Zhang, Oleksii Kuchaiev, Jiantao Jiao, and Banghua Zhu. The measure of all measures: Quantifying LLM benchmark quality. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling*, 2025. URL https://openreview.net/forum?id=HpnGmTIs7Z.

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/.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL https://aclanthology.org/N19-1131/.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Lexin Zhou, Lorenzo Pacchiardi, Fernando Martínez-Plumed, Katherine M. Collins, Yael Moros-Daval, Seraphina Zhang, Qinlin Zhao, Yitian Huang, Luning Sun, Jonathan E.

Prunty, Zongqian Li, Pablo Sánchez-García, Kexin Jiang Chen, Pablo A. M. Casares, Jiyun Zu, John Burden, Behzad Mehrbakhsh, David Stillwell, Manuel Cebrian, Jindong Wang, Peter Henderson, Sherry Tongshuang Wu, Patrick C. Kyllonen, Lucy Cheke, Xing Xie, and José Hernández-Orallo. General scales unlock ai evaluation with explanatory and predictive power, 2025. URL https://arxiv.org/abs/2503.06378.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fs9EabmQrJ.

Terry Yue Zhuo, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen GONG, James Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, David Lo, Binyuan Hui, Niklas Muennighoff, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YrycTjllL0.

A Training Details

MIRT Fitting We maximize the log-likelihood of Equation (1):

$$\ell = \sum_{(m,x,r)\in\mathcal{R}_{\text{train}}} r \log p_{mx} + (1-r)\log(1-p_{mx}) \tag{13}$$

where $p_{mx} = \sigma(\alpha_x(\Theta_m^\top K_x - \delta_x))$.

We add a standard normal prior on abilities:

$$\log p(\Theta) = -\frac{1}{2}\sum_{m,d}\Theta_{md}^2$$

and an ℓ_1 penalty on loadings for sparsity. The objective is:

$$\mathcal{L} = \ell + \log p(\Theta) - \lambda \sum_{x,d} |K_{xd}|$$

Defining residuals $e_{mx} = p_{mx} - r$ for each observed record, and indexing parameter dimensions with d, we compute the gradients:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \delta_x} &= -\sum_m \alpha_x \cdot e_{mx} \\
\frac{\partial \mathcal{L}}{\partial \Theta_{md}} &= \sum_x \alpha_x \cdot e_{mx} \cdot K_{xd} + \Theta_{md} \\
\frac{\partial \mathcal{L}}{\partial K_{xd}} &= \sum_m \alpha_x \cdot e_{mx} \cdot \Theta_{md} + \lambda\,\text{sign}(K_{xd}) \\
\frac{\partial \mathcal{L}}{\partial \alpha_x} &= \sum_m (\Theta_m^\top K_x - \delta_x) e_{mx}
\end{aligned}$$

We optimize the parameters jointly using a coordinate descent algorithm.

MAP update We update $\hat{\Theta}_m$ for new models by optimizing Equation (9) with coordinate descent. We initialize new $\hat{\Theta}_m$ as the mean Θ from $\mathcal{M}_{\text{train}}$ after fitting.

The gradient with respect to the ability parameters is given by:

$$\begin{aligned}
\frac{\partial}{\partial \Theta_m}\log p(\Theta_m|x,r) = &-\Sigma^{-1}(\Theta_m - \mu_\Theta) \\
&+ \frac{\partial}{\partial \Theta_m}\log p(r|m,x)
\end{aligned}$$

We compute this over all items selected by the assessor, while keeping their parameters frozen.

For both phases of optimization, we use the minimize function implemented in SciPy (Virtanen et al., 2020). We clip probabilities to the range $[\epsilon, 1-\epsilon]$ where $\epsilon = 10^{-10}$ to avoid numerical instability in the logarithm. For fitting on the full historic set of records, we use the L-BFGS-B implementation for up to 1000 steps. For MAP estimation, we use the more precise trust-ncg algorithm for up to 100 steps.

A.1 Degenerate Items

Certain items in a training set may be *degenerate*. That is, they may be constant in terms of model responses, containing all correct (or all incorrect) responses for every single model in the training split. This may happen in cases where an item is too hard or otherwise impossible due to errors. This can also occur for an item that is too easy, perhaps having

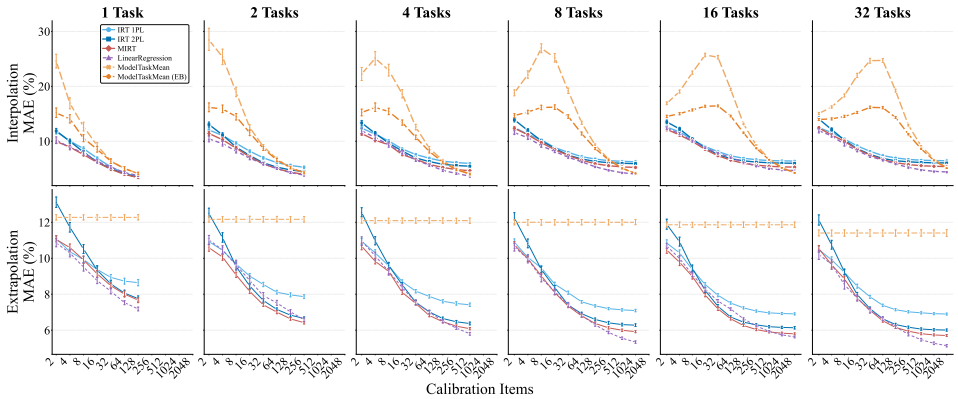

Figure 4: Full comparison of the (Nested) MIRT model against the IRT and Model Task Mean predictors. Model Task Mean (EB) represents the empirical bayes version of the estimator. Assessors receive random samples from T calibration tasks (interpolation) and then must predict how the model will perform on those T calibration tasks and on $128 - T$ validation tasks (extrapolation). The MIRT model validation MAE drops both when the number of calibration tasks or the number of samples increases.

been leaked into LLM training data. These items cause numerical stability issues for learning IRT models. We thus filter out degenerate items at runtime during training. At inference, when an IRT model is asked to make a prediction for a degenerate item, we then have no information for that item. To solve this, all models will heuristically predict the majority outcome for that item in the training data (correct or incorrect). This intuitively works quite well as these items will always have full agreement across train models.

A.2 Woodbury Correction

When doing V-Optimal selection, we need to update the posterior covariance after each observation. However, computing the inversion $\mathcal{I}^{-1}_{\text{cumulative}}$ is expensive. Instead, we apply the Woodbury matrix identity to perform a rank-1 update directly to C:

$$C \leftarrow C - \frac{w_x \, \alpha_x^2 C K_x K_x^\top C}{1 + w_x \, \alpha_x^2 K_x^\top C K_x} \tag{14}$$

This enables $O(d^2)$ sequential updates without recomputing $\mathcal{I}^{-1}_{\text{cumulative}}$ at each step.

B Additional Prediction Results

Model Task Mean Implementation The quality of the sample mean baseline described in Section 6.4 depends on the number of samples that we have per task. As $|\mathcal{R}_t| \to \infty$, this is the optimal predictor for the interpolation task. However, when $|\mathcal{R}_t|$ is small, it performs rather poorly. The Bayesian extension introduced in Section 6.4 resolves this with Bayesian shrinkage on the estimate with a Beta-Binomial prior. For each task, we estimate the mean performance over all train models $\mu_{\mathcal{M},t}$:

$$\hat{\theta}_{m,t} = \frac{\alpha + \sum_{\langle m,x,r \rangle \in \mathcal{R}_t} \mathbb{I}(r = 1)}{\alpha + \beta + |\mathcal{R}_t|}, \tag{15}$$

where $\alpha = \lambda \, \mu_{\mathcal{M},t}$ and $\beta = (1 - \mu_{\mathcal{M},t}) \, \lambda$ and λ is a parameter used to control the shrinkage strength. We set $\lambda = 5$ based on validation experiments.

Interpolation Results Figure 4 presents the full grid of interpolation and extrapolation results for the prediction experiments. Note that the second row here focused on extrapolation results is partially redundant with Figure 2 in the main body of this work, but is given

Table 2: Interpolation MAE (%) for $n_{\text{tasks}} = 1$. Each method uses its best available selector. Lower is better; **bold** marks the best method per column. This setting simulates prior work that focused on compressing a single benchmark.

	Items						
Method	**2**	**4**	**8**	**16**	**32**	**64**	**128**
IRT 1PL	11.76	10.03	8.65	6.91	5.33	4.19	3.52
IRT 2PL + Fisher	8.93	7.68	6.48	**5.39**	4.67	**3.90**	3.41
MIRT + VOptimal	**8.53**	**7.43**	**6.42**	5.46	**4.58**	3.94	**3.38**
Linear Regression	10.20	8.86	7.90	6.33	5.29	4.29	3.74
ModelTaskMean (EB)	15.17	14.05	10.55	8.52	6.35	5.01	4.11

again here for reference. In Table 2 we present interpolation results for the single task case. This reflects the setup used in prior work where a single benchmark is sampled, and the task is to compress it to a small set of items. In this setting, our MIRT implementation with VOptimal selection tends to still outperform all other methods with few exceptions.

B.1 Factor Analysis

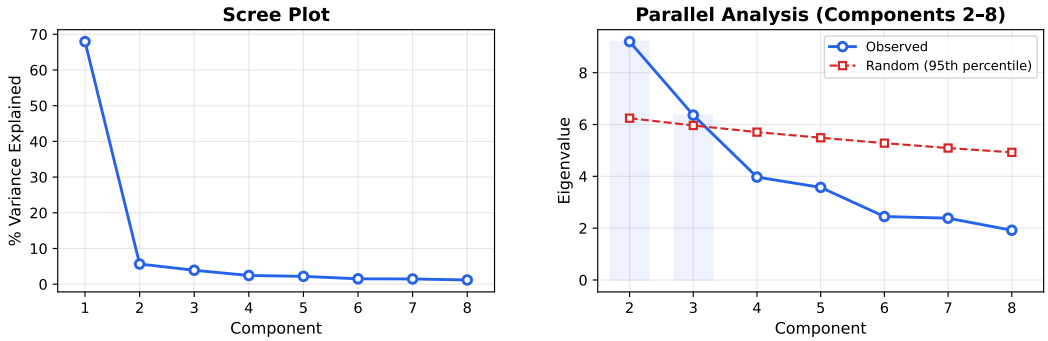

Figure 5: Scree plot and parallel analysis showing the most important factors in the training response matrix. A single factor explains almost 70% of variance, and two more latent factors can be selected via the elbow method.

We perform a factor analysis of the WILD dataset. Figure 5 presents a scree plot (left) of principal components, and a parallel analysis to visualize the first 8 eigenvalues (right) starting at the second, in order to zoom in on the factors >1. The scree plot shows that one factor explains almost 70% of the variance, consistent with one general ability, or "g-factor" across benchmarks. The parallel analysis shows that up to 3 components exceed the 95th percentile eigenvalue threshold of random data of equivalent dimensions. Given this, we set K=3 for all MIRT experiments in the main results. We also ablate K in further analysis (Figure 8), which shows that in some cases either 2, or more than 3 dimensions may attain stronger performance in the prediction setting. However, we maintain K=3 as to not pollute the main results from observations on the test set.

C Minimum Cost Selection

Figure 6 compares the token-aware discounted models against minimum cost baseline selectors, where the cheapest items in terms of token cost are greedily selected. The discounted models outperform the minimum cost models by a large margin, which shows that these cheap items are not informative enough for capability estimation, and information-aware selection is crucial.

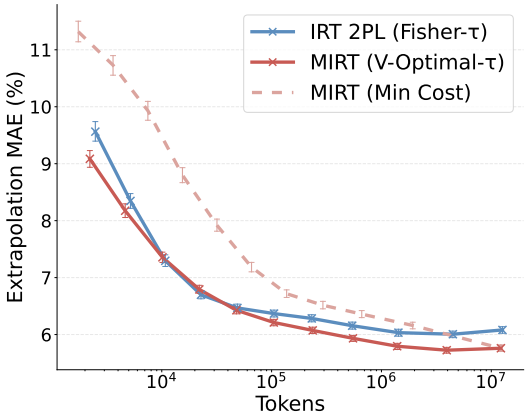

Figure 6: Comparison of cost discounted models to the minimum cost baseline. All results presented for 16 calibration tasks.

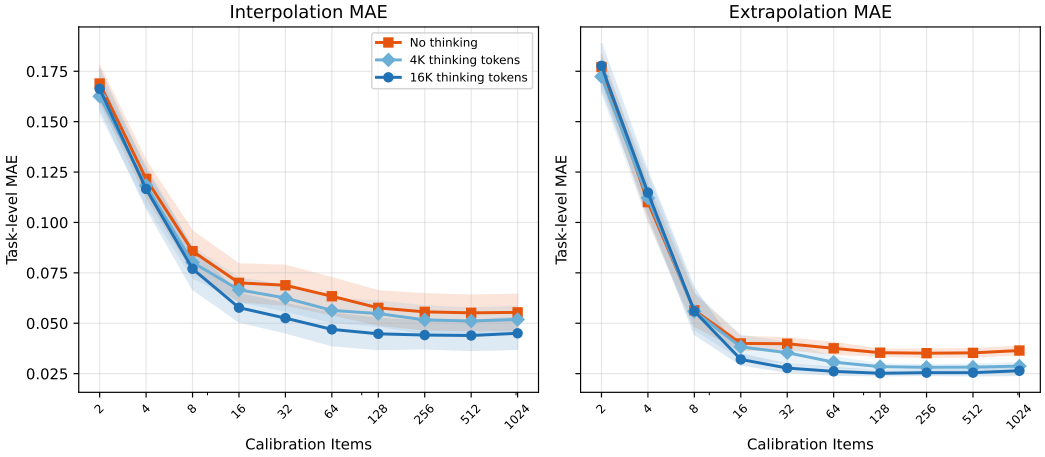

Figure 7: Analysis of interpolation and extrapolation prediction performance for Sonnet 4.6 at different reasoning budgets. Our method is robust to the model behavior changes related to newer reasoning abilities with large thinking budgets.

D Reasoning Model Preliminary Study

To evaluate the potential impact of extending the presented approach to reasoning models, we ran a preliminary experiment with Sonnet 4.6, a reasoning model not contained within WILD. We evaluated Sonnet on samples from 16 tasks. We used the same evaluation setting described in Section 6, with 8 of the tasks selected as calibration tasks and the remaining 8 held out as validation tasks. The full list of tasks is in Table 3. We ran the MIRT model described in the paper with the VOptimal selection in the adaptive setting. The results are shown in Figure 7. We did not observe any significant degradation in performance when the model is run with reasoning. However, we leave a more thorough investigation of the predictability of reasoning models for future work.

E MIRT Ablations

In Figure 8, we show the impact of varying K at different numbers of calibration tasks and items. Surprisingly, for 1 task, setting K=3 as we do in our main experiments is rather weak, though it is a reasonable hyperparameter as we increase the number of tasks and items. We also find that higher K is a strong choice at higher numbers of tasks and items, likely due to its higher representational capacity, and the fact that with sufficient items overfitting is not

Calibration Tasks	Held-out Tasks
BoolQ	BBH (Object Counting)
ChemBench (Organic Chemistry)	BigCodeBench
DROP	GSM8K
HellaSwag	MedQA
MATH (Precalculus)	MMLU-Pro (Health)
PAWS	MuSR
SQuAD	PiQA
TruthfulQA	RACE-H

Table 3: Tasks used for Sonnet 4.6 evaluation.

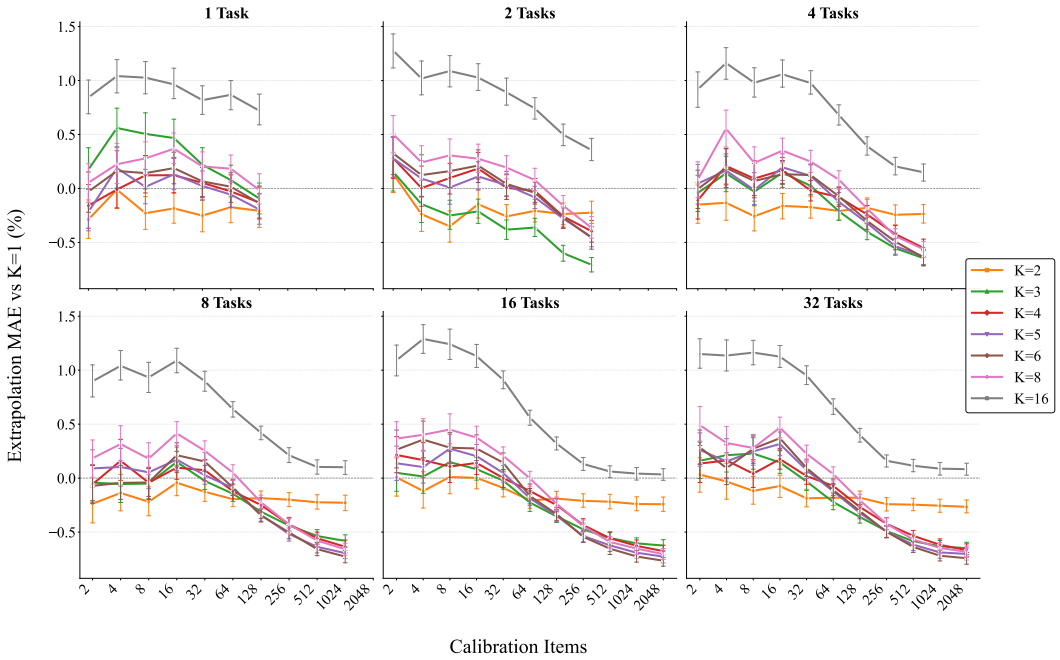

Figure 8: Ablation over K value for the MIRT model. Extrapolation MAE as a % difference from K=1. These results reflect the vanilla MIRT model, not Nested MIRT.

a concern. K=2 is a strong choice for low numbers of tasks and items, but is a weak model as more items are sampled. In Figure 9 we show that the adaptive IRT 2PL model often outperforms MIRT at low numbers of calibration tasks and items. The Nested MIRT model closes this gap while maintaining the MIRT model's strong performance as items and tasks increase.

F Selection Analysis

We analyze the rate at which the VOptimal and VOptimal-τ select tasks. We compute the average number of items selected per task, given the task is one of the calibration tasks. Figure 11 shows the top 10 tasks by this metric when there are 64 items to select. VOptimal seems to have a strong preference for Squad and various MMLU-Pro tasks. The cost-aware version focuses on similar MMLU-Pro tasks at lower rates, but heavily up-samples Squad and MMLU Miscellaneous. We hypothesize that the bias towards MMLU Miscellaneous is driven mainly by its low token cost, with an average of 120 total tokens per answer. Other low-cost tasks like ARC, Commonsense QA, and PAWS are noticeably absent, indicating that they were not as highly informative according to our model. MATH Intermediate Algebra and DROP, relatively costly tasks, are frequently sampled by the VOptimal model but are absent in the cost aware selection which samples from cheaper QA tasks and hellaswag.

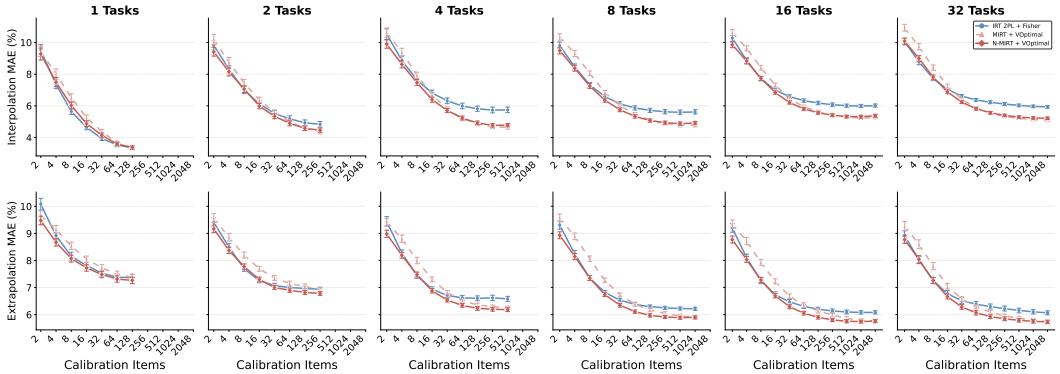

Figure 9: Comparison of MIRT with the Nested MIRT model for adaptive experiments. We plot the IRT 2PL model as a baseline comparison. Both MIRT models have K=3 parameters (including the general loading).

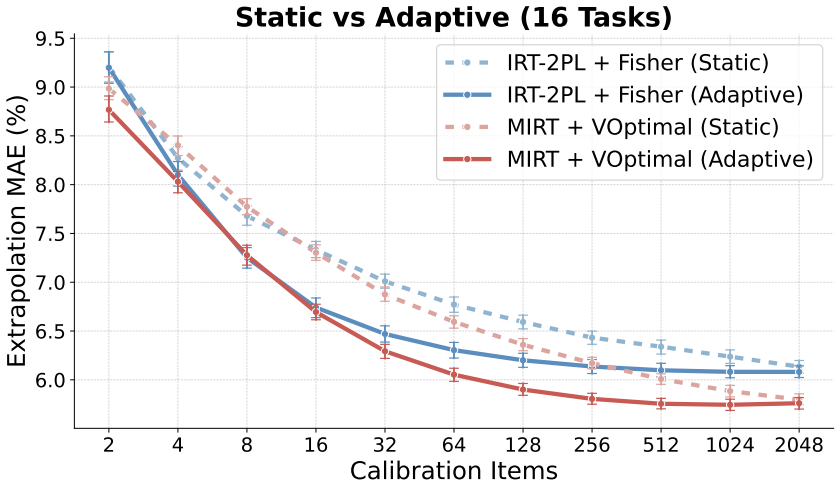

Figure 10: Comparison of adaptive and static selection on the same plot.

G Limitations

This work has a few potential limitations. First, we model only binary correctness labels, ignoring richer response information such as partial credit, answer quality, or confidence scores that could provide more nuanced ability estimates. IRT models can certainly be fit on non-binary data, but we focus our results on the binary case only. Second, our benchmark consists primarily of short-horizon tasks with clear correct answers. Long-horizon tasks (e.g., multi-step planning, extended dialog, or complex tool use) may require abilities not well-represented in our current item pool, potentially limiting the generalizability of our ability estimates to such settings.

H Further Results Analysis: Task and Model Variance

We compare the IRT-2PL with Fisher selection assessor to the MIRT with VOptimal selection assessor at the task level in Figure 12 and Figure 13, where results are averaged over LLMs. At 16 calibration items (Figure 12), where both models perform similarly, we see a split in which model is best for which task. MIRT is much stronger for MATH, whereas IRT-2PL is stronger for SQUAD. Both of these tasks are among the highest MAE for both models. Big Bench Hard (BBH) is very high MAE at approximately the same performance for both models. At 128 calibration items (Figure 13), MIRT attains a lower MAE on every task, and the gap in MATH increases. SQUAD remains notably difficult to predict, whereas

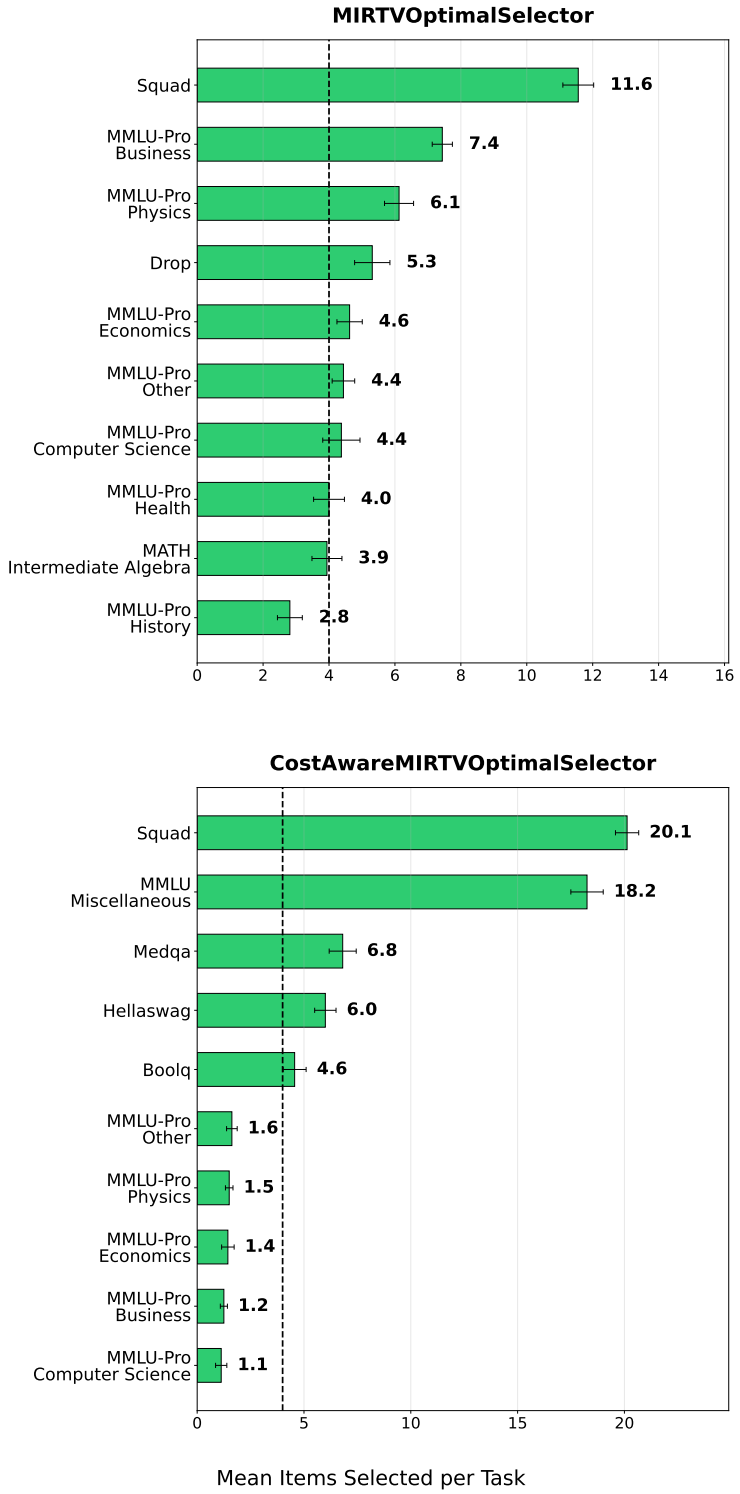

Figure 11: Comparison of the task selection rates between the V-Optimal and cost-aware V-Optimal (V-Optimal-τ) when selecting 64 calibration items. Dotted line represents expected amount of occurrences if the selection was random.

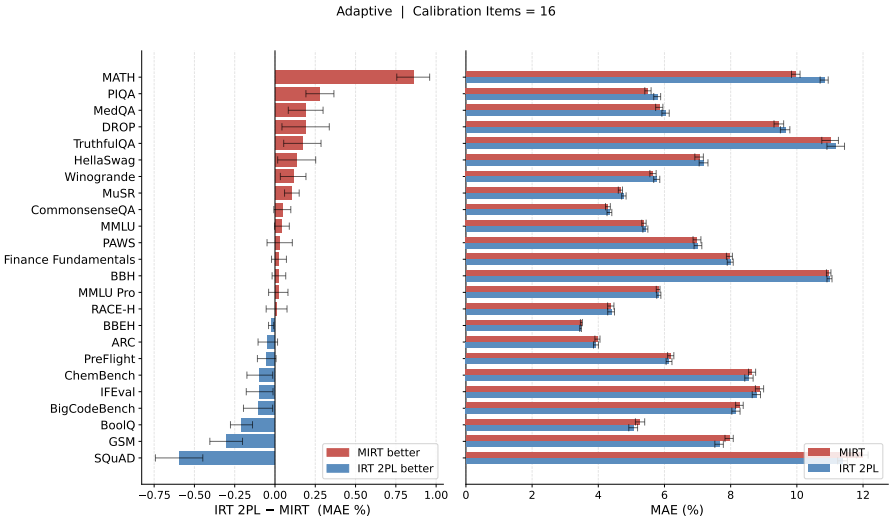

Figure 12: Results breakdown per-task for adaptive models with 16 calibration tasks and 16 calibration items. **Left**: Relative MAE comparison of the IRT-2PL model with Fisher selection vs. MIRT with VOptimal selection. Further left in blue means that the 2PL model attained a lower MAE, whereas further right in red means that the MIRT model had a lower MAE. **Right**: Absolute MAE comparisons per task. Lower is better.

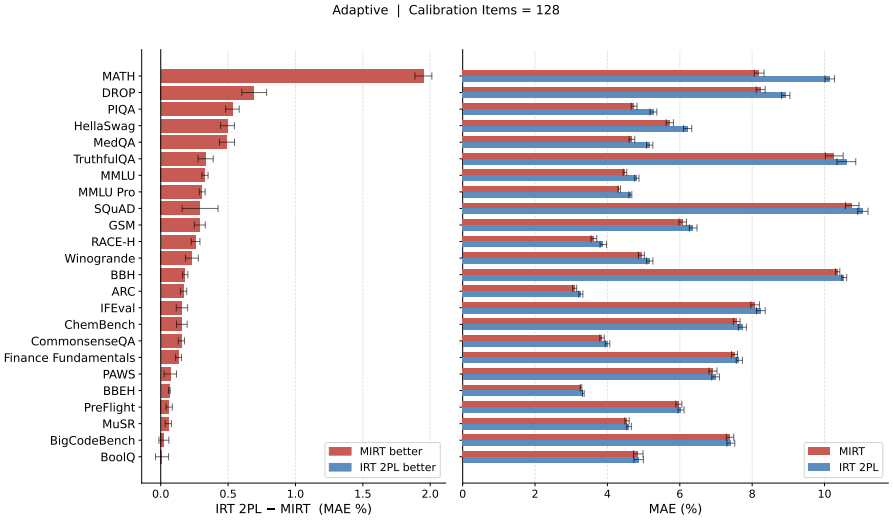

Figure 13: Results breakdown per-task for adaptive models with 16 calibration tasks and 128 calibration items. **Left**: Relative MAE comparison of the IRT-2PL model with Fisher selection vs. MIRT with VOptimal selection. Further left in blue means that the 2PL model attained a lower MAE, whereas further right in red means that the MIRT model had a lower MAE. **Right**: Absolute MAE comparisons per task. Lower is better.

HellaSwag, for instance, has a large drop in MAE. Figure 14 shows that as we sample more tasks, adaptive MIRT consistently outperforms the 2PL model on its best performing tasks, and, with a smaller gap in MAE, on its worse performing tasks too.

In Figure 15 and Figure 16 we compare these assessors at the model level, where results are averaged over tasks. At 16 calibration items, both assessors perform similarly for several models, with a slightly longer tail of models that IRT-2PL performs best on. The MIRT assessor has a large drop in MAE for Qwen2-MATH-7b-instruct—the most difficult model to predict—relative to the IRT-2PL assessor. At 128 calibration items, the MIRT assessor is

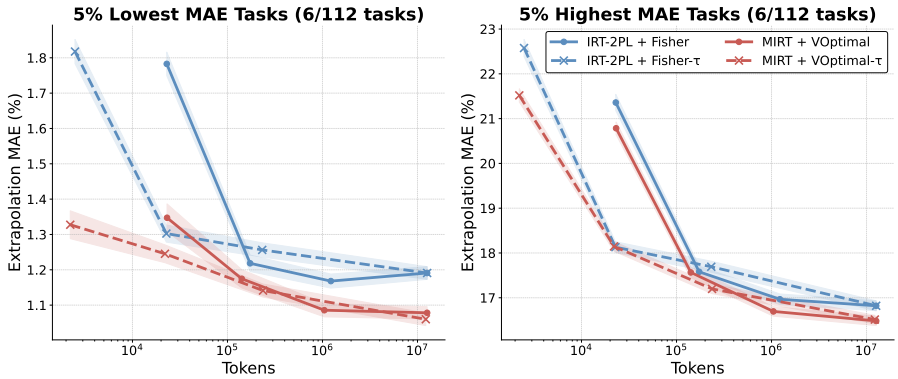

Figure 14: Comparison of assessor performance for the 5% best-performing, i.e., lowest MAE tasks (left) and 5% worst-performing, i.e., highest MAE tasks (right).

best for almost all models. One exception to this is phi-4, for which IRT-2PL maintains a much smaller MAE than MIRT.

I Held Out Estimation

The goal of our analysis is evaluating how well our assessors can predict $\theta_{m,t}$, the true probability of model m correctly answering an item from task t. This is distinct from evaluating how well the model can predict $\mu_r(\mathcal{R}_t)$.

For each task t, we can partition the total set of records $\mathcal{R}_t$ for that task into two sets, $\mathcal{O}_t$ representing the observed items for the assessor and $\mathcal{U}_t$ the complementary unobserved items s.t. $\mathcal{O}_t \cup \mathcal{U}_t = \mathcal{R}_t$ and $\mathcal{O}_t \cap \mathcal{U}_t = \emptyset$. We do not constrain the observed set to be of equal size across tasks (e.g. $|\mathcal{O}_1| \neq |\mathcal{O}_2|$), but we do enforce that any sampling algorithm may only be a function of $\mathcal{O}_t$.

$\theta_{m,t}$ is a latent variable we can never directly observe. In order to measure the quality of the assessor we need a proxy to use as ground truth. It is common for existing work on data set compression to use the mean over the entire set of available data, $\mu_r(\mathcal{R}_t)$. In this work, we take special care to always use a separate set of data unobserved by the assessor, $\mu_r(\mathcal{U}_t)$.

For an arbitrary prediction $q_{m,t}$ that is a function of $\mathcal{O}_t$, we can write for the held-out set,

$$\mathbb{E}\left[(q_{m,t} - \mu_r(U_t))^2\right] = \mathbb{E}[(q_{m,t} - \theta_{m,t})^2] + \mathrm{Var}(\mu_r(U_t)). \tag{16}$$

Whereas for the total set, the error decomposes into three terms,

$$\mathbb{E}\left[(q_{m,t} - \mu_r(\mathcal{R}_t))^2\right] = \mathbb{E}[(q_{m,t} - \theta_{m,t})^2] + \mathrm{Var}(\mu_r(\mathcal{R}_t)) - 2\frac{|\mathcal{O}_t|}{|\mathcal{R}_t|}\mathrm{Cov}(q_{m,t}, \mu_r(\mathcal{O}_t)). \tag{17}$$

With the held-out target, every assessor's error decomposes into its squared error with the true $\theta_{m,t}$, plus a per-task sampling noise constant. Since this constant only varies by task, assessor performance can be compared directly without controlling for the size of the observed set.

With the full-set target, the error contains an additional assessor-specific covariance term to account for the fact the observed items are used to both form a prediction and to compute the target itself. This particularly confounds comparisons between assessors who have received disparate, non-uniform item samples across the tasks.

When the proxy is $\mu_r(\mathcal{U}_t)$, $\theta_{m,t}$ is the optimal prediction,

$$\mathbb{E}\left[(\theta_{m,t} - \mu_r(\mathcal{U}_t))^2\right] = \mathrm{Var}(\mu_r(\mathcal{U}_t)) = \frac{\theta_{m,t}(1 - \theta_{m,t})}{|\mathcal{U}_t|}. \tag{18}$$

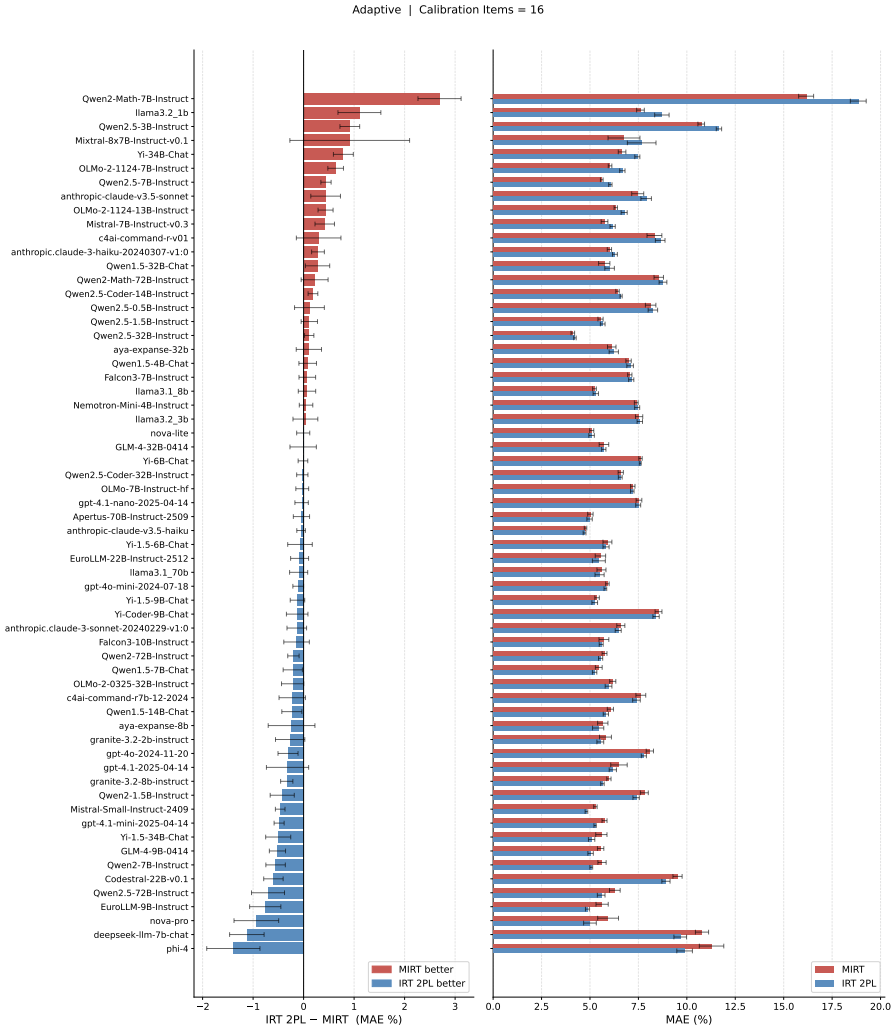

Figure 15: Results breakdown per-model for adaptive models with 16 calibration tasks and 16 calibration items. **Left**: Relative MAE comparison of the IRT-2PL model with Fisher selection vs. MIRT with VOptimal selection. Further left in blue means that the 2PL model attained a lower MAE, whereas further right in red means that the MIRT model had a lower MAE. **Right**: Absolute MAE comparisons per task. Lower is better.

However, when the proxy is instead $\mu_r(\mathcal{R}_t)$, we can improve the predictor by smoothing it with the observed items. The expected error is,

$$\mathbb{E}\left[(\theta_{m,t} - \mu_r(\mathcal{R}_t))^2\right] = \mathrm{Var}(\mu_r(\mathcal{R}_t)) = \frac{\theta_{m,t}(1-\theta_{m,t})}{|\mathcal{R}_t|}. \tag{19}$$

We can instead use

$$\tilde{\theta}_{m,t} = \frac{|\mathcal{O}_t|}{|\mathcal{R}_t|}\mu_r(\mathcal{O}_t) + \frac{|\mathcal{U}_t|}{|\mathcal{R}_t|}\theta_{m,t}. \tag{20}$$

This gives us a lower expected error of

$$\mathbb{E}\left[(\tilde{\theta}_{m,t} - \mu_r(\mathcal{R}_t))^2\right] = \mathrm{Var}(\mu_r(\mathcal{R}_t))\left(1 - \frac{|\mathcal{O}_t|}{|\mathcal{R}_t|}\right). \tag{21}$$

Since our goal is recovering $\theta_{m,t}$ rather than directly compressing the exact benchmark scores, we grade only against the held-out targets.

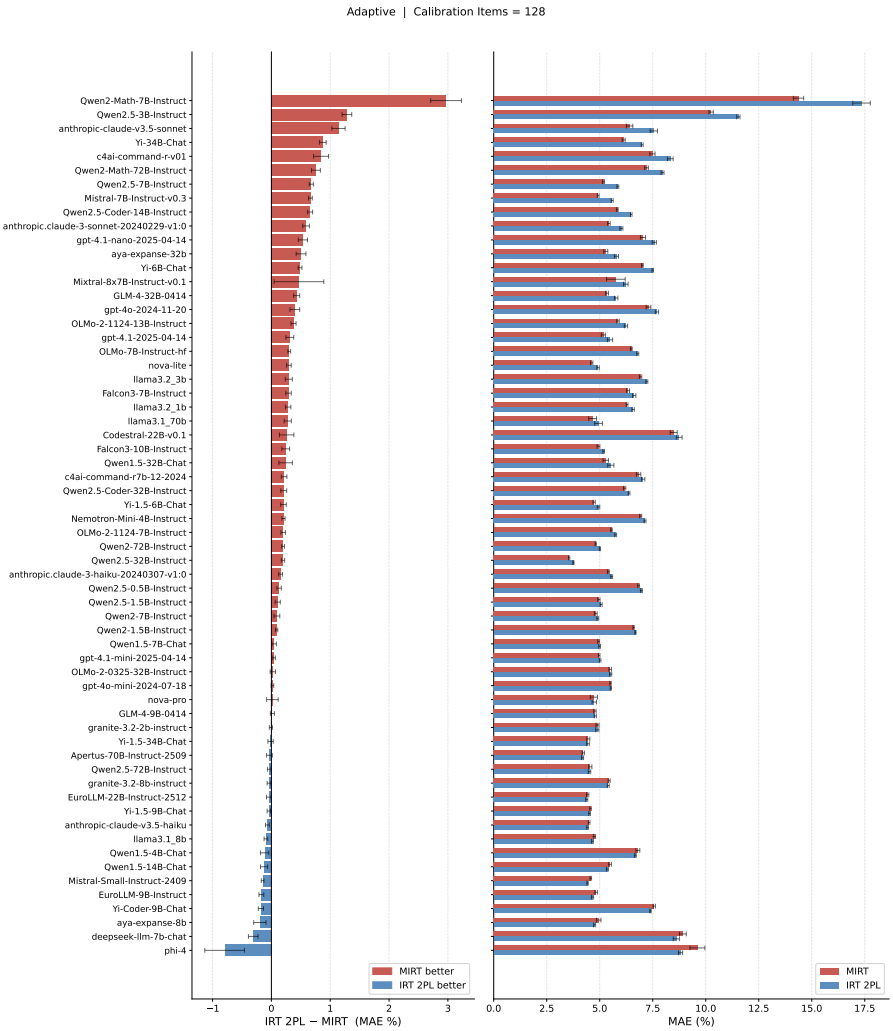

Figure 16: Results breakdown per-model for adaptive models with 16 calibration tasks and 128 calibration items. **Left**: Relative MAE comparison of the IRT-2PL model with Fisher selection vs. MIRT with VOptimal selection. Further left in blue means that the 2PL model attained a lower MAE, whereas further right in red means that the MIRT model had a lower MAE. **Right**: Absolute MAE comparisons per task. Lower is better.

Table 4: Data partitioning schema for calibration and validation.

Category	Models	Tasks	Access
Training	$\mathcal{M}_{train}$	$\mathcal{T}^c \cup \mathcal{T}^v$	Full obs.
Calibration	$\mathcal{M}_{test}$	$\mathcal{T}^c$ (item subset)	Test-time
Interpolation	$\mathcal{M}_{test}$	$\mathcal{T}^c$ (held-out items)	**Predicted**
Extrapolation	$\mathcal{M}_{test}$	$\mathcal{T}^v$	**Predicted**

J Model & Task List

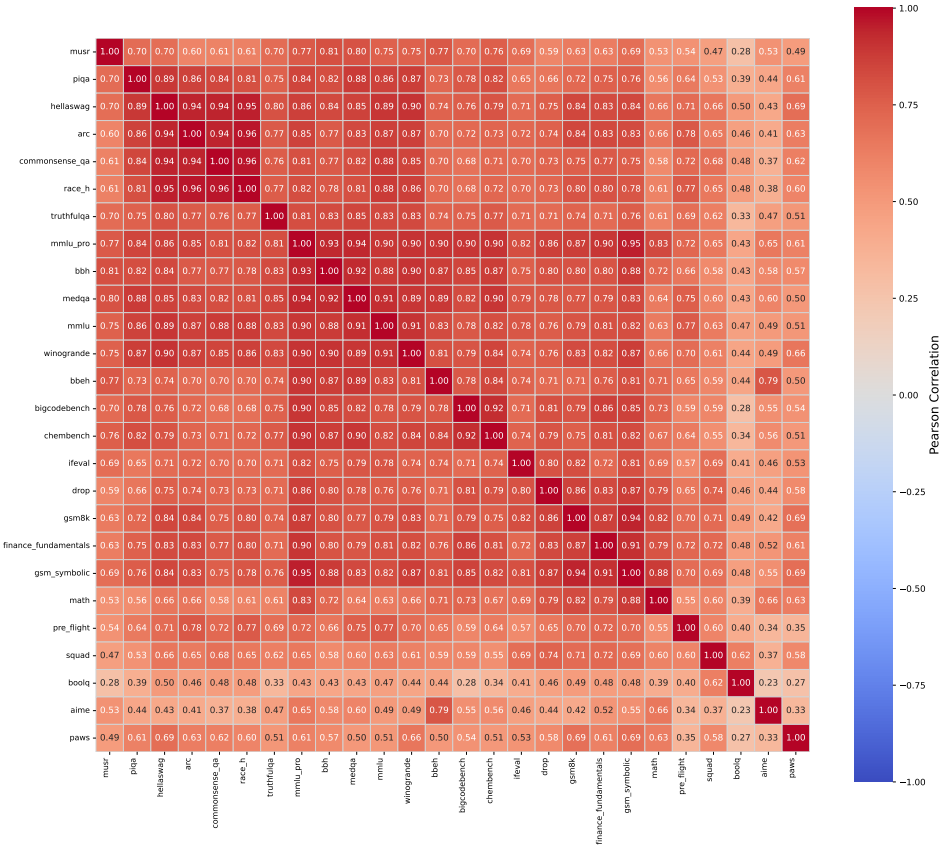

Figure 17: Pearson correlation between LLM performance across all base datasets. Datasets are ordered based on hierarchical clustering of their correlation strength.

Table 5: Benchmarks, tasks, and number of unique evaluation items contained in WILD

Dataset	Task	Items
WinoGrande Sakaguchi et al. (2021)		1,267
TruthfulQA Lin et al. (2022)		817
SQuAD Rajpurkar et al. (2016)		11,866
RACE-H Lai et al. (2017)		3,498
Pre-Flight Inspect Evals (2025)		300
PIQA Bisk et al. (2020)		1,803
PAWS Zhang et al. (2019)		7,982
MUSR Sprague et al. (2024)		250
DROP Dua et al. (2019)		8,687
BoolQ Clark et al. (2019)		3,270
BigCodeBench Zhuo et al. (2025)		1,139
BBH Suzgun et al. (2023)	Boolean Expressions	250
	Date Understanding	250
	Disambiguation QA	250
	Dyck Languages	250
	Formal Fallacies	250
	Geometric Shapes	250
	Multistep Arithmetic (Two)	250
	Navigate	250
	Object Counting	250
	Reasoning About Colored Objects	250
	Salient Translation Error Detection	250
	Temporal Sequences	250
	Tracking Shuffled Objects (3)	250
	Tracking Shuffled Objects (5)	250
	Tracking Shuffled Objects (7)	250
	Web of Lies	250
	Word Sorting	250
	Movie Recommendation	249
	Sports Understanding	248
	Ruin Names	210
	Causal Judgement	185
	Penguins in a Table	146
	Logical Deduction (3)	99
	Logical Deduction (5)	98
	Logical Deduction (7)	95
BBEH Kazemi et al. (2025)	Boardgame QA	200
	Boolean Expressions	200
	Buggy Tables	200
	Causal Understanding	200
	Dyck Languages	200
	Geometric Shapes	200
	Hyperbaton	200
	Movie Recommendation	200
	Multistep Arithmetic	200
	NYCC	200
	Object Counting	200
	Object Properties	200
	Sarcasm Triples	200
	Shuffled Objects	200
	SportQA	200
	Temporal Sequence	200
	Time Arithmetic	200
	Web of Lies	200
	Word Sorting	200
	Zebra Puzzles	200
	Spatial Reasoning	199
	Linguini	194
	Disambiguation QA	120
ARC Clark et al. (2018)	Easy	2,371
	Challenge	1,170
AIME Mathematical Association of America (1983)	2024	30
	2025	30
MMLU Pro Wang et al. (2024)	Math	1,349
	Physics	1,294
	Chemistry	1,132
	Law	1,023
	Engineering	968
	Other	924
	Economics	840
	Psychology	796
	Business	779
	Health	778
	Biology	634
	Philosophy	498
	Computer Science	409
	History	381

Dataset	Task	Items
	Professional Law	1,534
	Moral Scenarios	895
	Miscellaneous	783
	Professional Psychology	609
	High School Psychology	534
	High School Macroeconomics	386
	Elementary Mathematics	376
	Moral Disputes	346
	Prehistory	323
	Philosophy	311
	High School Biology	310
	Nutrition	299
	Professional Accounting	281
	Professional Medicine	272
	High School Mathematics	265
	Clinical Knowledge	264
	Security Studies	244
	High School World History	237
	Marketing	234
	Conceptual Physics	233
	High School Microeconomics	233
	Human Aging	223
	High School Statistics	215
	High School US History	204
	Sociology	201
	High School Chemistry	198
	High School Geography	198
	High School Government and Politics	193
MMLU Hendrycks et al. (2021a)	College Medicine	172
	World Religions	171
	Virology	166
	High School European History	164
	Logical Fallacies	162
	High School Physics	149
	Electrical Engineering	145
	Astronomy	145
	College Biology	144
	Anatomy	135
	Human Sexuality	131
	Formal Logic	126
	International Law	121
	Econometrics	114
	Machine Learning	110
	Public Relations	108
	Jurisprudence	108
	Management	103
	Medical Genetics	100
	College Mathematics	100
	Computer Security	100
	Global Facts	100
	Business Ethics	100
	Abstract Algebra	100
	College Computer Science	100
	High School Computer Science	100
	College Chemistry	99
	US Foreign Policy	99
	College Physics	84
MedQA Jin et al. (2021)		1,273
	Algebra	1,187
	Intermediate Algebra	903
	Prealgebra	871
MATH Hendrycks et al. (2021b)	Precalculus	546
	Number Theory	540
	Geometry	479
	Counting & Probability	474
IFEval Zhou et al. (2023)		541
HellaSwag Zellers et al. (2019)		10,018
GSM8K Cobbe et al. (2021)		1,319

Dataset	Task	Items
GSM-Symbolic Mirzadeh et al. (2025)	P1	5,000
	P2	2,500
Finance Fundamentals Krumdick et al. (2024)	Finknow	131
	TAT-QA (Extract)	129
	CodeTAT-QA	105
	ConvFinQA (Extract)	94
	FinCode	92
	CodeFinQA	49
CommonsenseQA Talmor et al. (2019)		1,186
	Person	35
ChemBench Mirza et al. (2024)	Toxicity and Safety	674
	Organic Chemistry	403
	Physical Chemistry	165
	Analytical Chemistry	152
	General Chemistry	148
	Inorganic Chemistry	92
	Materials Science	83
	Technical Chemistry	40

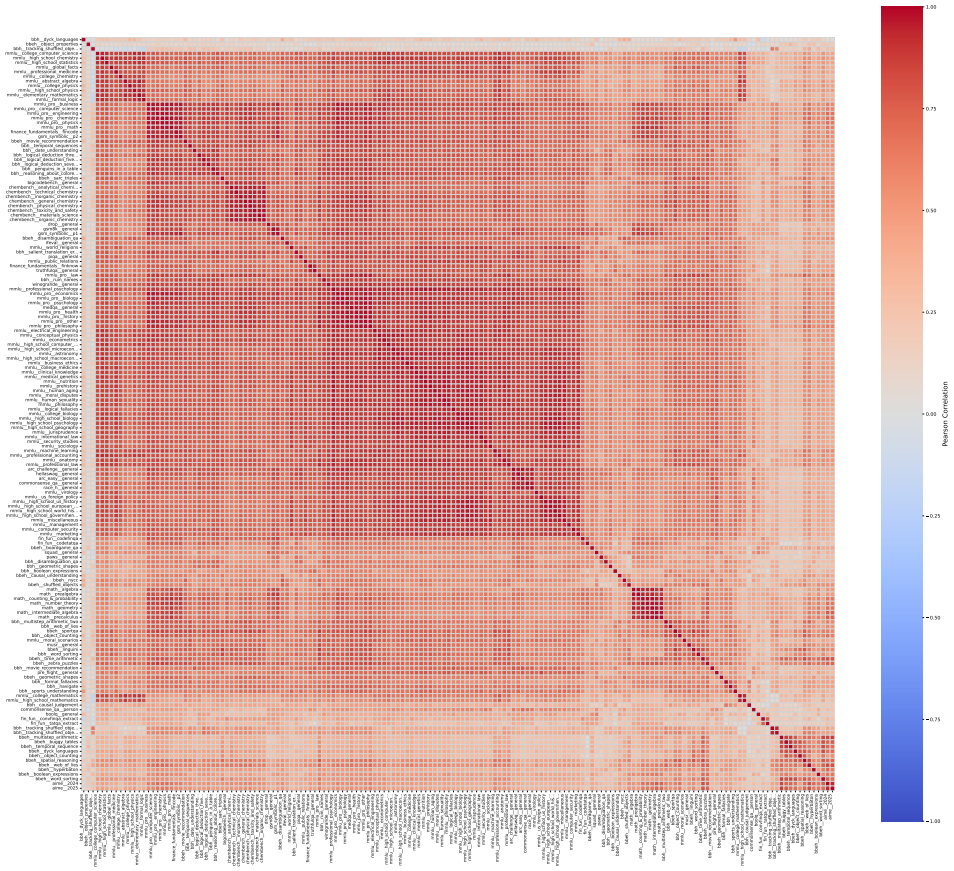

Figure 18: Pearson correlation between LLM performance across all sub-tasks. Tasks are ordered based on hierarchical clustering of their correlation strength.

Table 6: List of models contained in WILD.

Model	Access?
Claude 3 Haiku	API
Claude 3 Sonnet	API
Claude 3.5 Haiku	API
Claude 3.5 Sonnet	API
GPT 4.1	API
GPT 4.1 mini	API
GPT 4.1 nano	API
GPT-4o	API
GPT-4o mini	API
Nova Lite	API
Nova Pro	API
meta-llama/Llama-3.1-70B	HuggingFace
meta-llama/Llama-3.1-8B	HuggingFace
meta-llama/Llama-3.2-1B	HuggingFace
meta-llama/Llama-3.2-3B	HuggingFace
01-ai/Yi-1.5-34B-Chat	HuggingFace
01-ai/Yi-1.5-6B-Chat	HuggingFace
01-ai/Yi-1.5-9B-Chat	HuggingFace
01-ai/Yi-34B-Chat	HuggingFace
01-ai/Yi-6B-Chat	HuggingFace
01-ai/Yi-Coder-9B-Chat	HuggingFace
CohereForAI/aya-expanse-32b	HuggingFace
CohereForAI/aya-expanse-8b	HuggingFace
CohereForAI/c4ai-command-r-plus-08-2024	HuggingFace
CohereForAI/c4ai-command-r-v01	HuggingFace
CohereForAI/c4ai-command-r7b-12-2024	HuggingFace
Qwen/Qwen1.5-14B-Chat	HuggingFace
Qwen/Qwen1.5-32B-Chat	HuggingFace
Qwen/Qwen1.5-4B-Chat	HuggingFace
Qwen/Qwen1.5-7B-Chat	HuggingFace
Qwen/Qwen2-1.5B-Instruct	HuggingFace
Qwen/Qwen2-72B-Instruct	HuggingFace
Qwen/Qwen2-7B-Instruct	HuggingFace
Qwen/Qwen2-Math-72B-Instruct	HuggingFace
Qwen/Qwen2-Math-7B-Instruct	HuggingFace
Qwen/Qwen2.5-0.5B-Instruct	HuggingFace
Qwen/Qwen2.5-1.5B-Instruct	HuggingFace
Qwen/Qwen2.5-32B-Instruct	HuggingFace
Qwen/Qwen2.5-3B-Instruct	HuggingFace
Qwen/Qwen2.5-72B-Instruct	HuggingFace
Qwen/Qwen2.5-7B-Instruct	HuggingFace
Qwen/Qwen2.5-Coder-14B-Instruct	HuggingFace
Qwen/Qwen2.5-Coder-32B-Instruct	HuggingFace
allenai/OLMo-2-0325-32B-Instruct	HuggingFace
allenai/OLMo-2-1124-13B-Instruct	HuggingFace
allenai/OLMo-2-1124-7B-Instruct	HuggingFace
allenai/OLMo-7B-Instruct-hf	HuggingFace
deepseek-ai/deepseek-llm-7b-chat	HuggingFace
ibm-granite/granite-3.2-2b-instruct	HuggingFace
ibm-granite/granite-3.2-8b-instruct	HuggingFace
microsoft/phi-4	HuggingFace
mistralai/Codestral-22B-v0.1	HuggingFace
mistralai/Ministral-8B-Instruct-2410	HuggingFace
mistralai/Mistral-7B-Instruct-v0.3	HuggingFace
mistralai/Mistral-Small-Instruct-2409	HuggingFace
mistralai/Mixtral-8x7B-Instruct-v0.1	HuggingFace
nvidia/Nemotron-Mini-4B-Instruct	HuggingFace
swiss-ai/Apertus-70B-Instruct-2509	HuggingFace
swiss-ai/Apertus-8B-Instruct-2509	HuggingFace
tiiuae/Falcon3-10B-Instruct	HuggingFace
tiiuae/Falcon3-7B-Instruct	HuggingFace
utter-project/EuroLLM-22B-Instruct-2512	HuggingFace
utter-project/EuroLLM-9B-Instruct	HuggingFace
zai-org/GLM-4-32B-0414	HuggingFace
zai-org/GLM-4-9B-0414	HuggingFace