How Predictable is AI Progress?

Anonymous Authors[1]

Abstract

Benchmarks only serve to measure what models are capable of now, not what they *will be* capable of in the future. We find that the ordering of acquired capabilities is remarkably consistent across large populations of AI models, which begs the question of whether one can forecast which specific examples and capabilities future models will solve next. We propose formalizing this problem into a new evaluation task called *progress prediction*: Can we forecast which unsolved problems will be solved next as future models improve? We find that progress follows a structured pattern, a sign of potential predictability. Through an empirical study of hundreds of millions of predictions made by 1,000+ vision models and 1,500+ language models, we find that this predictability may be possible due to the consistent order in which models acquire capabilities across architectures, datasets, and modalities.

1. Introduction

Benchmarks offer an evaluation of where model capabilities currently stand, but offer little predictive power of where model capabilities will be. When a model improves from 70% to 80% accuracy, which examples will it learn to solve? The default approach is to wait for the capabilities of models to reach this level of performance and evaluate the examples in question at that time. But we argue this may not be the only way to determine which examples are solvable in the future.

We study the fine-grained structure of AI progress across over 1,000 vision models and 1,500 language models, spanning a decade of architectural and algorithmic advances. We do this by analyzing the patterns of correct and incorrect model inference on several vision and language tasks. Our central finding is that models solve benchmark examples in a remarkably consistent order. This ordering is weaker for language than for vision, but it remains far above a random baseline. Easy examples are easy for everyone; hard examples are hard for everyone; and the transition from failure to success follows a shared sequence. This is not a trivial observation about task difficulty, but rather an empirical claim about convergence across models with different architectures, training data, and optimization procedures. Phrased another way, given only a model's overall accuracy on ImageNet, for example, and knowing nothing else about the model we can tell what examples the model will get correct and incorrect on the test set. This stability reveals a capability frontier, a boundary separating the set of examples solvable by a model of a given accuracy from those that are not.

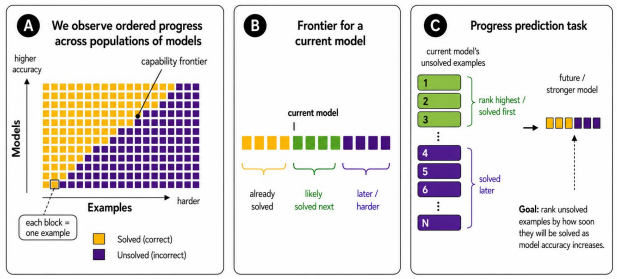


Figure 1. We pose the problem of progress prediction: forecasting which currently unsolved examples will be solved next as AI models improve.

We formalize measuring the universality of example difficulty by introducing a new metric, Prediction Order Coherence (POC), which measures the extent to which higher-accuracy models solve a strict superset of what lower-accuracy models solve. A POC of 1 indicates perfect ordering; a POC of 0 indicates models solve disjoint subsets. We find POC values of 0.6–0.77 on vision benchmarks and 0.17–0.45 on language benchmarks—far above the random baseline of near zero.

The structure present in the pattern of increasingly powerful models' predictions begs the question: Given today's models, can we predict which unsolved problems will be solved next as future models improve through increasing scale and algorithmic efficiency?

We propose, as a new challenge to the deep learning com-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

munity, the task of AI progress prediction, predicting where the capability frontier will advance. For a given model M and dataset D, we consider the set of examples $N^{M,D}$ that the model currently fails to solve. The goal is to identify the subset of examples from $N^{M,D}$ that appear earliest in the population consensus ordering, as these represent the next capabilities the model is most likely to acquire.

Our contributions are:

1. We show that AI progress shows substantial ordering: models solve benchmark examples in a consistent sequence, with POC scores of 0.6–0.77 (vision) and 0.17–0.45 (language) across 1,000+ vision and 1,500+ language models.
2. We formalize *progress prediction*—predicting which examples a model will solve given only its accuracy—as a new task for the community.
3. We establish baselines for progress prediction using confidence, entropy, and other proxy metrics, achieving AUC 0.64 (vision) and 0.60 (language).

2. Related Work

We study example difficulty through cross-model agreement and training dynamics. Prior work shows that neural networks broadly agree on which images are easy or hard (Hacohen et al., 2020), and that learning exhibits stable instance-level structure through forgetting events, dataset cartography, and difficulty proxies such as the *c-score*, *prediction depth*, and *VoG* (Toneva et al., 2019; Swayamdipta et al., 2020; Jiang et al., 2021; Baldock et al., 2021; Agarwal et al., 2022). Human-grounded measures such as Minimum Viewing Time provide an orthogonal perceptual notion of difficulty (Mayo et al., 2023). Related work uses example-level signals for curriculum learning, data pruning, and targeted data selection to improve training efficiency (Bengio et al., 2009; Hacohen & Weinshall, 2019; Paul et al., 2021; Sinha et al., 2020; Xia et al., 2024; Bi et al., 2025; Sorscher et al., 2022), while forecasting studies model aggregate performance trends over scale or time (Hestness et al., 2017; Kwa et al., 2025; Sevilla et al., 2022). In contrast, we aggregate predictions across many models to forecast the instance-level order in which future models acquire capabilities, proposing a cross-domain approach to progress prediction rather than a method for optimizing current training.

3. Problem Formulation

As AI models improve—through scale, data, or algorithmic advances—they solve problems they previously could not. But which problems? We formalize this as *progress prediction*: given the set of examples a model currently fails to solve, can we predict which will be solved next as model capabilities improve?

This task hinges on an empirical question: do models follow a consistent ordering as they improve, or does each model solve a different random subset? If it is the former prediction is possible.

Towards this end, we begin by introducing *Prediction Order Coherence* (POC) to measure the degree of ordering in a population's predictions.

3.1. Methods

Consider a population of m models evaluated on a shared benchmark of n examples. We binarize each model's prediction on each example as correct (1) or incorrect (0).

We represent the population's collective individual example performance as a *prediction matrix* $P \in \{0,1\}^{m\times n}$, where $P_{ij} = 1$ if model i correctly solves example j. Rows correspond to models; columns correspond to examples. We sort rows by model accuracy (highest at top) and columns by *population solve rate*—the fraction of models that solve each example (most-solved on left, hardest on right).

3.2. Measuring Orderedness Using POC

For any pair of models H,L, where H has higher accuracy than L, each example belongs to one of four cases: Q1 if both H and L are correct; Q2 if H is correct and L is incorrect; Q3 if H is incorrect and L is correct; and Q4 if both H and L are incorrect.

Q3 counts ordering violations—examples where a worse model succeeds but a better model fails. In a perfectly ordered world, Q3 $= 0$: better models solve everything worse models solve, plus additional examples.

Given a population of models with fixed accuracies, we define three reference points. *Matched order*: models solve maximally overlapping subsets, with Q3 $= 0$ for all pairs. *Opposite order*: models solve maximally disjoint subsets, with Q3 maximized given the accuracy constraints. *Random order*: each model's predictions are independent draws given its accuracy.

To quantify where an observed population falls on this spectrum, we construct three prediction matrices that share the same row sums (matching empirical model accuracies) but differ in structure: P_{observed} is the empirical prediction matrix; P_{matched} has 1s left-aligned in each row, achieving perfect ordering; and P_{opposite} has 1s distributed to equalize column sums, achieving maximal disorder. Since $\text{Q2} = \text{Q3} + (\text{acc}_H - \text{acc}_L)$ and the accuracy difference is fixed, minimizing Q3 is equivalent to minimizing Q2; we use Q2 to formulate our metric. For each matrix, we sum Q2 across all $\binom{m}{2}$ model pairs. See Appendix A.6 and A.4 for construction algorithms. We define Prediction Order

Coherence (POC) as:

$$\text{POC} = \frac{\text{Q2}_{\text{observed}} - \text{Q2}_{\text{opposite}}}{\text{Q2}_{\text{matched}} - \text{Q2}_{\text{opposite}}} \quad (1)$$

where each term is the sum of Q2 across all model pairs for the corresponding matrix.

POC $= 1$ indicates perfect ordering; POC $= 0$ indicates maximal disorder. For independent random predictions, $\text{Q2}_{\text{random}} = \sum_{i<j}\left(L_i - \frac{L_i L_j}{n}\right)$ where L_i is the number of examples model i solves; this yields a population-dependent baseline (see Appendix A.7).

3.3. Progress Prediction Task

Given that progress is at least partially ordered (Section 4), a natural question arises: can we estimate the difficulty ordering cheaply, without training and evaluating a large population of models?

We formalize this as the *progress prediction task*. The population ordering provides a ground truth ranking of example difficulty, examples solved by more models are easier and examples solved by fewer models are harder. The goal is to find *proxy metrics* that approximate this ordering using only a single model, with the hope that a proxy which predicts well in the difficulty range of examples that at least some models get right the proxy will generalize to predicting the ordering of examples that remain unsolved by all current models.

3.4. Backtesting Framework

How can we evaluate progress prediction when we don't know the future? We backtest. Given a model M whose accuracy falls within our population's range, we consider its unsolved set U_M. Each example in U_M has a ground truth difficulty rank derived from the population: some examples are just beyond M's capability frontier (solved by slightly better models), others are far beyond it (solved only by the best models, or by none).

A proxy metric assigns a score to each example in U_M—an estimate of how close that example is to being solved. We evaluate the proxy by asking: does ranking examples by proxy score recover the true difficulty ordering?

3.5. Evaluating Proxy Metrics

For each model M, we backtest on

$$V_M = \{x \in U_M : \exists\, M' \text{ with } \text{acc}(M') > \text{acc}(M) \text{ and } M' \text{ solves } x\},$$

the unsolved examples that at least one higher-accuracy model solves; examples solved by no model are excluded. This treats M as the present and higher-accuracy models as the future. A proxy score $s(x)$ ranks examples in V_M from predicted easiest to hardest and is compared with the population difficulty ranking.

For $K = 1, \ldots, |V_M|$, let $\text{Top}_K(s)$ be the K highest-scoring examples and $\text{Top}_K(\text{pop})$ the K easiest examples under the population ordering. We define

$$\text{Prec}(K) = \frac{|\text{Top}_K(s) \cap \text{Top}_K(\text{pop})|}{K},$$

$$\text{AUC} \approx \frac{1}{|V_M|}\sum_{K=1}^{|V_M|}\text{Prec}(K).$$

equivalently the area under the precision curve with x-axis $K/|V_M|$. A random proxy yields the diagonal $\text{Prec}(K) = K/|V_M|$ with AUC 0.5, while a perfect proxy has $\text{Prec}(K) = 1$ and AUC 1. We assume proxies that recover the known ordering in this backtest generalize to examples unsolved by all current models.

3.6. Baseline Proxy Metrics

We establish initial baselines for the progress prediction task. Our goal is not to find the optimal proxy, but to demonstrate that the task is tractable and to provide a framework for future work to develop more sophisticated metrics. We evaluate two families of proxy metrics: with ground truth labels (the softmax probability assigned to the correct class) and without ground truth labels (Predicted logit: confidence on the model's top predic- tion, Entropy: uncertainty in the output distribution, and Total variation distance from uniform: how peaked the distribution is).

These baselines are intentionally simple. We report results using the evaluation framework defined above, and encourage future work to develop proxies that incorporate richer signals—example features, model internals, or historical training dynamics—and to report comparable AUC scores.

3.7. Data Collection

We collected model predictions across language and vision tasks. For language models, we leveraged the HuggingFace Open LLM Leaderboard evaluations, which provides evaluation results for over 1,500 language models across 5 challenging datasets: Big Bench Hard (BBH) (Suzgun et al., 2022; bench authors, 2023), MATH (Hendrycks et al., 2021b), Graduate-Level Google-Proof Q&A Benchmark (GPQA) (Rein et al., 2024), Multistep Soft Reasoning (MUSR) (Sprague et al., 2024), Massive Multitask Language Understanding Pro (MMLU-Pro) (Hendrycks et al., 2021a; Wang et al., 2024).

For vision models, we evaluated 1,000+ models from the timm repository (Wightman, 2019) across three diverse

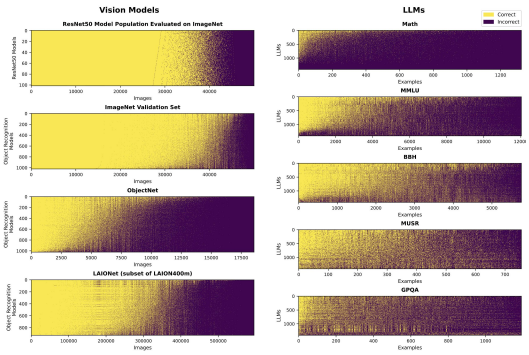


Figure 2. Model population predictions on several vision and language tasks. Rows in each of the bars are models, columns are dataset examples, and each cell is a model prediction scored as correct (yellow) or incorrect (purple).

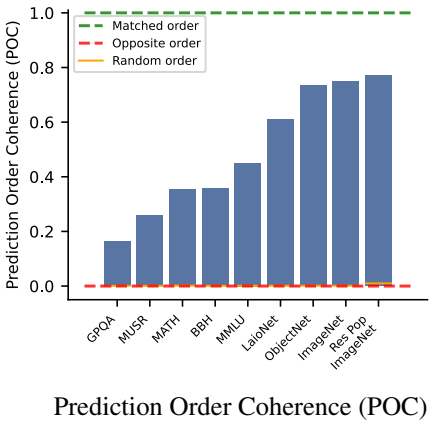


Figure 3. Agreement between model predictions and population patterns across different tasks. Green dashed line indicates perfect agreement (matched order), red dashed line represents maximal disagreement (opposite order), and orange line shows agreement with a simulated random ordering of predictions.

computer vision benchmarks: ImageNet (Krizhevsky et al., 2012), ObjectNet (Barbu et al., 2019), and LAIONet (Shirali & Hardt, 2023). ImageNet represents a standard object recognition task, ObjectNet tests out of distribution robustness controlling for viewpoint, rotation, and background, and LAIONet tests generalization to an ImageNet like subset of LAION400m, offering a larger scale of over 500k images that could in the future be used along with population difficulty information for training. Additionally, we trained 100 ResNet50 models on ImageNet, each with a different random seed.

4. Results

4.1. Observed Population Prediction Patterns

The aggregated prediction results from the model populations are shown in Figure 2. The matrices show a far more matched than random ordering in which higher accuracy models tend to subsume the successes of lower accuracy models across all modalities and tasks, though it is most visible for vision models.

We measure the degree of this ordering using the Prediction Order Coherence metric as seen in Figure 3. Vision models on object recognition tasks exhibit a higher POC while LLMs exhibit lower POC values. We hypothesize that tasks that require less trivial knowledge and more compositional skill will exhibit a higher POC.

4.2. Next Solvable Example AUC

For every model M and dataset D we generate a plot of $\frac{K}{N^{M,D}} \in [0,1]$ vs the set intersection percent (or precision) at K (also $\in [0,1]$) which represents the next K examples that the proxy metric predicts will be solved. We repeat this for all the models and average the lines to get a single mean line representing the population average. We then calculate the AUC of this plot. A perfect predictor would obtain an AUC of 1 while a completely random predictor would get 0.5. In Appendix Figure 4 we report vision models and LLMs separately as they operate on different datasets. A comprehensive list of metrics are listed in Appendix A.3

Our findings show that consistently across almost all models and multiple modalities, models are able to predict progress on the subset of examples that they cannot solve yet (0.642 AUC and 0.597 AUC compared to random 0.5). To highlight the strength of our findings, model-based metrics can predict progress even when knowledge of the ground truth label is not used.

Appendix Figure 4b and Figure 4d show that we can consistently, above random, predict progress without having access to ground truth labels (0.545 for vision and 0.550 for language, random is 0.5).

5. Discussion

As AI systems approach solving high-value problems in science, medicine, and other domains, predicting not just if, but when specific problems will be solved becomes an important problem for prioritizing research and reasoning about the future.

We pose the problem of progress prediction: forecasting which currently unsolved examples will be solved next as AI models improve. We offer baseline methods for predicting future solvable examples.

We invite the research community to develop more sophisticated progress prediction methods and apply our evaluation framework to measure their success.

References

Agarwal, C., D'souza, D., and Hooker, S. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10368–10378, 2022.

Baldock, R., Maennel, H., and Neyshabur, B. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://proceedings.neurips.cc/paper/2021/file/5a4b25aaed25c2ee1b74de72dc03c14e-Paper.pdf.

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., and Katz, B. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper/2019/file/97af07a14cacba681feacf3012730892-Paper.pdf.

bench authors, B. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009. URL https://doi.org/10.1145/1553374.1553380.

Bi, J., Wang, Y., Yan, D., Xiao, X., Hecker, A., Tresp, V., and Ma, Y. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. 2025. URL https://doi.org/10.48550/arXiv.2502.12119.

Hacohen, G. and Weinshall, D. On the power of curriculum learning in training deep networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/hacohen19a.html.

Hacohen, G., Chosen, L., and Weinshall, D. Let's agree to agree: Neural networks share classification order on real datasets. *ICML*, 2020.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021b.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically, 2017. URL https://arxiv.org/abs/1712.00409.

Jiang, Z., Zhang, C., Talwar, K., and Mozer, M. C. Characterizing structural regularities of labeled data in overparameterized models. In *Proceedings of the 38th International Conference on Machine Learning (PMLR)*, 2021. URL https://proceedings.mlr.press/v139/jiang21k.html.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

Kwa, T., West, B., Becker, J., Deng, A., Garcia, K., Hasin, M., Jawhar, S., Kinniment, M., Rush, N., Arx, S. V., Bloom, R., Broadley, T., Du, H., Goodrich, B., Jurkovic, N., Miles, L. H., Nix, S., Lin, T., Parikh, N., Rein, D., Sato, L. J. K., Wijk, H., Ziegler, D. M., Barnes, E., and Chan, L. Measuring ai ability to complete long tasks, 2025. URL https://arxiv.org/abs/2503.14499.

Mayo, D., Cummings, J., Lin, X., Gutfreund, D., Katz, B., and Barbu, A. How hard are computer vision datasets? calibrating dataset difficulty to viewing time. *Advances in Neural Information Processing Systems*, 36:11008–11036, 2023.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Uj7pF-D-YvT.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., and Villalobos, P. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022.

Shirali, A. and Hardt, M. What makes imagenet look unlike laion. 2023.

Sinha, S., Garg, A., and Larochelle, H. Curriculum by smoothing. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f6a673f09493afcd8b129a0bcf1cd5bc-Paper.pdf.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, pp. 19523–19536, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/7b75da9b61eda40fa35453ee5d077df6-Paper-Conference.pdf.

Sprague, Z., Ye, X., Bostrom, K., Chaudhuri, S., and Durrett, G. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. 2024.

Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q. V., Chi, E. H., Zhou, D., , and Wei, J. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020. URL https://aclanthology.org/2020.emnlp-main.746/.

Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Xia, M., Malladi, S., Gururangan, S., Arora, S., and Chen, D. Less: Selecting influential data for targeted instruction tuning. In *ICML*, 2024. URL https://openreview.net/forum?id=PG5fV50maR.

A. Appendix

A.1. Proxy metrics AUC plots

A.2. Future Predictions as Binary Classification

In this section, we treat the task of predicting the future similar to Figure 3 but as a binary classification problem where we only attempt to predict the next $X\%$ of examples.

For each $X \in \{5, 10, 20, 50\}$ we have 4 plots (thus 16 in total) in Figures 5,6, and 7,8. The two plots in the top row are generated using the 37 vision models listed in Appendix A.8 while the two plots in the bottom row are generated using the 26 LLMs also mentioned in Appendix A.8. There is a significant difference between the left column and the right column; while the left column achieves a higher AUC, it does depend on us knowing the ground truth labels of each example (as the ground softmax is used for the prediction) while the right column does not depend on the ground truth label at all, merely on statistics of the outputted set of logits. The significance of this is that the right column tells us that even if neither humans nor the models knew the correct answer, the models are still able to know that the examples are approaching being solved (consistently better than random).

When we refer to "logits total variation" for LLMs we specifically take the logits of each of the n possible multiple choice answers then compute $\|L_i - \frac{1}{n}\|_1$, this metric was chosen as it had achieved the best progress prediction without knowing the ground label. For a comprehensive list of every metric tried, we list all of them in Appendix A.3 along with the AUC for the 5% future prediction task (similar to Figure 5).

A.3. Comprehensive List of Features and ROCs

Here we list a comprehensive list of many metrics we tried along with the AUC ROC for the 5% future prediction task (similar to Figure 5). The metrics are for both vision models and LLMs, and every metric either relies on or does not rely on the ground label. The four tables are Table 1, Table 2, Table 3, and Table 4

Table 1. AUC ROC of metrics from vision models that rely on the ground label

Value	AUC ROC
Targets Softmax	0.793
Targets Logits	0.725

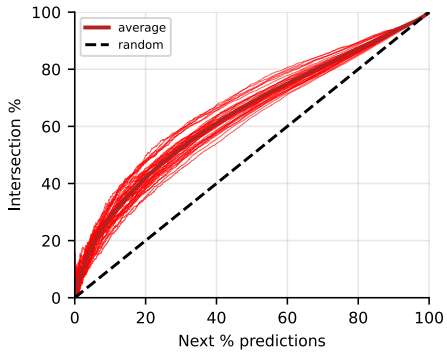
A.4. Derivation of $Q2_{\text{opposite}}$

Here we derive how we obtain $Q2_{\text{opposite}}$ representing the theoretical minimum (opposite ordering), where higher-accuracy models systematically get different examples correct than lower-accuracy models. We define that the theoretical minimum ordering (or opposite ordering) is when the sum of the quantity Q_3 from Figure **??** is maximized across all pairs of models.

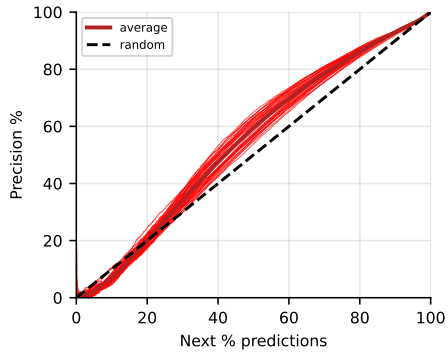
Let $M \in \{0,1\}^{n \times m}$ be a binary matrix with rows $(i = 1, \ldots, n)$ and columns $(k = 1, \ldots, m)$ where $M_{ik} = 1$ implies that model i predicts sample k correctly. Then we fix the row sums of M to match our empirical real model population accuracies such that the sum of the i-th row equals $L(i)$ where $L(i)$ is the number of samples that model i gets correctly. Assume the rows of M are ordered such that the associated accuracies satisfy $L(1) \geq L(2) \geq \cdots \geq L(n)$. The exact sorting of rows isn't required, but for the purposes of this derivation, it makes the equations simpler and easier to follow if the rows are sorted in non-increasing order. We then define the function Q_3 (eq. 2):

$$Q_3(i, j) := |\{k \in \{1, \ldots, m\} : M_{i,k} = 0,\ M_{j,k} = 1\}| \tag{2}$$

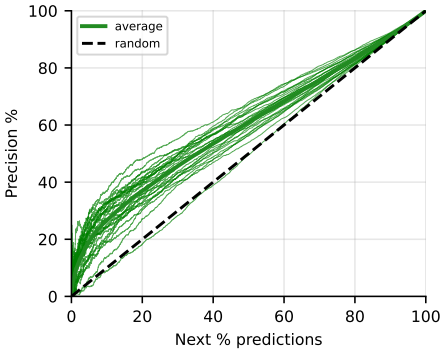
Where $(1 \leq i < j \leq n)$ (i.e., the number of samples that a lower accuracy model j gets right, while the higher accuracy model i gets wrong). Then we note the simple relation:



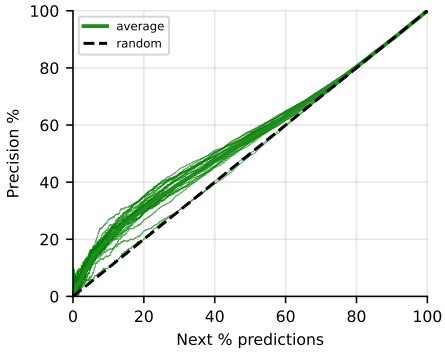
(a) Vision models *progress prediction*.
relying on ground labels (AUC: 0.642)
prediction made using ground truth logit



(b) Vision models *progress prediction*.
without ground labels (AUC: 0.545)
predictions used raw predicted logit



(c) LLM *progress prediction*.
relying on ground labels (AUC: 0.597)
predictions used ground truth logit



(d) LLM *progress prediction*.
without ground labels (AUC: 0.550)
predictions made using entropy of logits

Figure 4. Comparison of different progress prediction methods for next percent prediction. These 4 metrics are used to make a progress prediction, ground truth logit confidence (4a), predicted logit confidence (4b), ground truth logit confidence (4c), and entropy of logits (4d). 26 LLMs and 37 vision models were used for these experiments.

Table 2. AUC ROC of metrics from vision models that do not rely on the ground label

Value	AUC ROC
Logits Cr 2	0.560
Predicted Softmax	0.557
Logits Cr 3	0.537
Logits Mode Value	0.534
Logits Mean	0.531
Logits Std	0.529
Logits Variance	0.529
Logits Renyi Q3	0.528
Logits Renyi Q2 Entropy	0.526
Logits Participation Ratio	0.526
Logits Simpson Index	0.526
Logits Hill Number Q2	0.526
Logits Renyi Q2	0.526
Logits Gini Impurity	0.526
Logits Hhi	0.526
Logits Tsallis Q2 Entropy	0.526
Logits Shannon Entropy	0.525
Logits Kl To Uniform	0.524
Logits Cr 1	0.524
Logits Max Prob	0.524
Logits Min Entropy	0.524
Logits Mode Prob	0.524
Logits Hellinger Distance To Uniform	0.523
Logits Js Divergence To Uniform	0.523
Logits Skewness	0.523
Logits Kurtosis Excess	0.523
Logits Total Variation To Uniform	0.523
Predicted Logits	0.509
Logits Hill Number Q1	0.505
Logits Perplexity	0.505
Logits Cr 5	0.504
Logits Iqr	0.501
Logits Q25	0.501
Logits Median	0.500
Logits Q75	0.500
Logits Min Prob	0.500
Logits Min Value	0.500
Logits Max Value	0.500
Logits Hill Number Q0	0.500

Table 3. AUC ROC of metrics from LLMs that rely on the ground label

Value	AUC ROC
Target Logit	0.708
Target Softmaxed	0.636

Table 4. AUC ROC of metrics from LLMs that do not rely on the ground label

Value	AUC ROC
Logits Cr 2	0.688
Logits Cr 3	0.686
Softmaxed Cr 5	0.678
Logits Renyi Q3	0.677
Logits Participation Ratio	0.677
Logits Hill Number Q2	0.677
Logits Hhi	0.677
Logits Simpson Index	0.677
Logits Renyi Q2 Entropy	0.677
Logits Renyi Q2	0.677
Logits Tsallis Q2 Entropy	0.677
Logits Gini Impurity	0.677
Logits Total Variation To Uniform	0.677
Logits Shannon Entropy	0.676
Logits Perplexity	0.676
Logits Hill Number Q1	0.676
Logits Kl To Uniform	0.676
Logits Js Divergence To Uniform	0.675
Logits Hellinger Distance To Uniform	0.675
Softmaxed Cr 3	0.673
Logits Cr 5	0.670
Logits Mode Prob	0.668
Logits Cr 1	0.668
Logits Max Prob	0.668
Logits Min Entropy	0.668
Softmaxed Min Prob	0.666
Softmaxed Cr 2	0.659
Softmaxed Hellinger Distance To Uniform	0.648
Softmaxed Js Divergence To Uniform	0.640
Softmaxed Total Variation To Uniform	0.636
Softmaxed Kl To Uniform	0.629
Softmaxed Perplexity	0.629
Softmaxed Shannon Entropy	0.629
Softmaxed Hill Number Q1	0.629
Pred Logit	0.627
Logits Min Prob	0.625
Softmaxed Hill Number Q2	0.613
Softmaxed Hhi	0.613
Softmaxed Gini Impurity	0.613
Softmaxed Participation Ratio	0.613
Softmaxed Simpson Index	0.613
Softmaxed Tsallis Q2 Entropy	0.613
Softmaxed Renyi Q2	0.613
Softmaxed Renyi Q2 Entropy	0.613
Softmaxed Renyi Q3	0.607
Pred Softmaxed	0.602
Softmaxed Cr 1	0.599
Softmaxed Mode Prob	0.599
Softmaxed Max Prob	0.599
Softmaxed Min Entropy	0.599
Softmaxed Variance	0.586
Softmaxed Std	0.586
Softmaxed Iqr	0.558
Logits Kurtosis Excess	0.557
Softmaxed Skewness	0.543
Logits Mode Value	0.540
Softmaxed Mode Value	0.540
Softmaxed Kurtosis Excess	0.540
Softmaxed Q25	0.539
Logits Q75	0.538
Logits Std	0.533
Logits Variance	0.533
Logits Mean	0.529
Softmaxed Median	0.526
Logits Iqr	0.525
Logits Skewness	0.523
Logits Q25	0.515
Softmaxed Mean	0.514
Softmaxed Q75	0.502
Logits Median	0.502
Logits Hill Number Q0	0.500

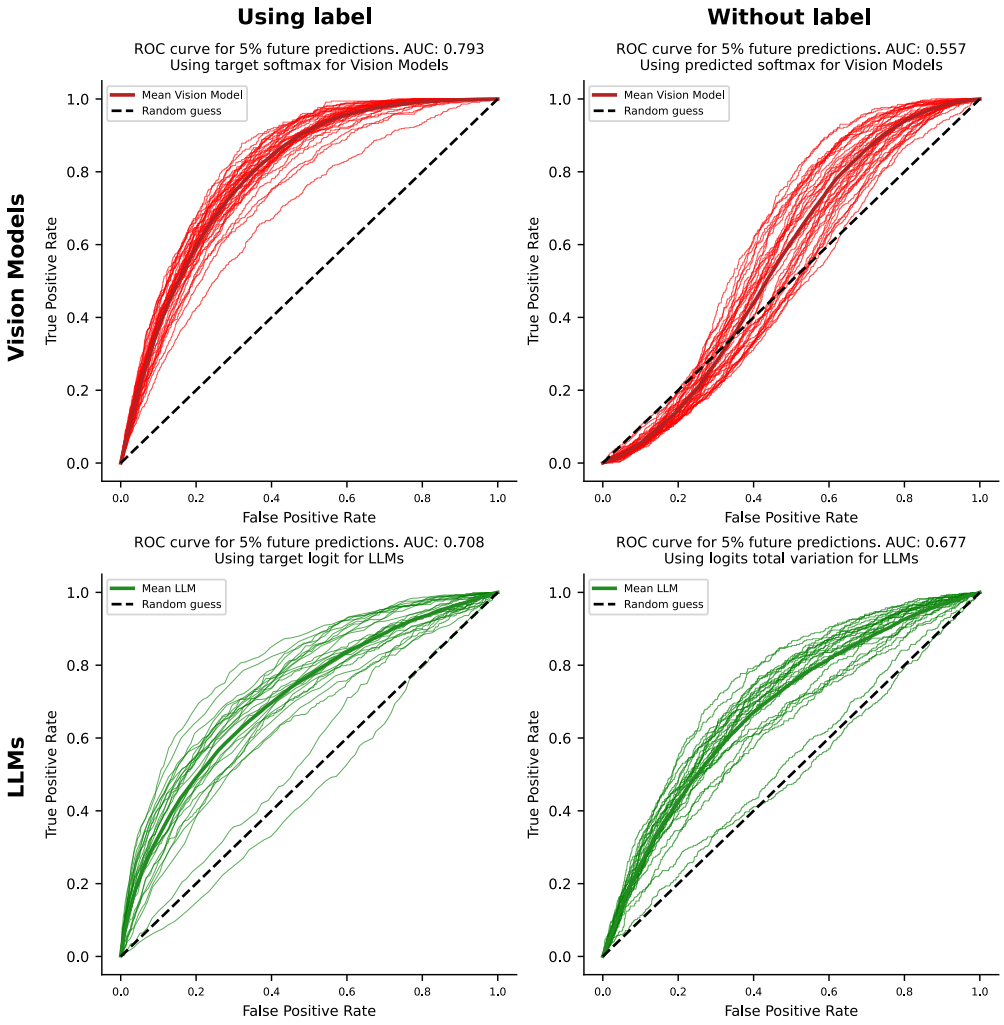


Figure 5. Here we only attempt to predict the next 5% ahead of each model's capabilities. In all plots, we again see the consistent trend that almost all models are capable of predicting the near future beyond what they are capable of solving.

$$L(j) = \sum_{k=1}^{m} \mathbf{1}\{M_{j,k} = 1\} = \sum_{k=1}^{m} \mathbf{1}\{M_{j,k} = 1,\ M_{i,k} = 0\} + \sum_{k=1}^{m} \mathbf{1}\{M_{j,k} = 1,\ M_{i,k} = 1\} \tag{3}$$

$$\therefore \sum_{k=1}^{m} \mathbf{1}\{M_{j,k} = 1,\ M_{i,k} = 0\} = L(j) - \sum_{k=1}^{m} \mathbf{1}\{M_{j,k} = 1,\ M_{i,k} = 1\} \tag{4}$$

Then we define $\mathrm{DIS}(M)$ (eq. 5)

$$\mathrm{DIS}(M) = \sum_{1 \le i < j \le n} Q_3(i, j). \tag{5}$$

Where $\mathrm{DIS}(M)$ is what we want to maximize to obtain $Q2_{\text{opposite}}$ representing our opposite ordering. Given a matrix M, it is trivial to algorithmically calculate $\mathrm{DIS}(M)$, therefore, for our main goal of obtaining $Q2_{\text{opposite}}$ is to obtain the matrix M that maximizes $\mathrm{DIS}(M)$.

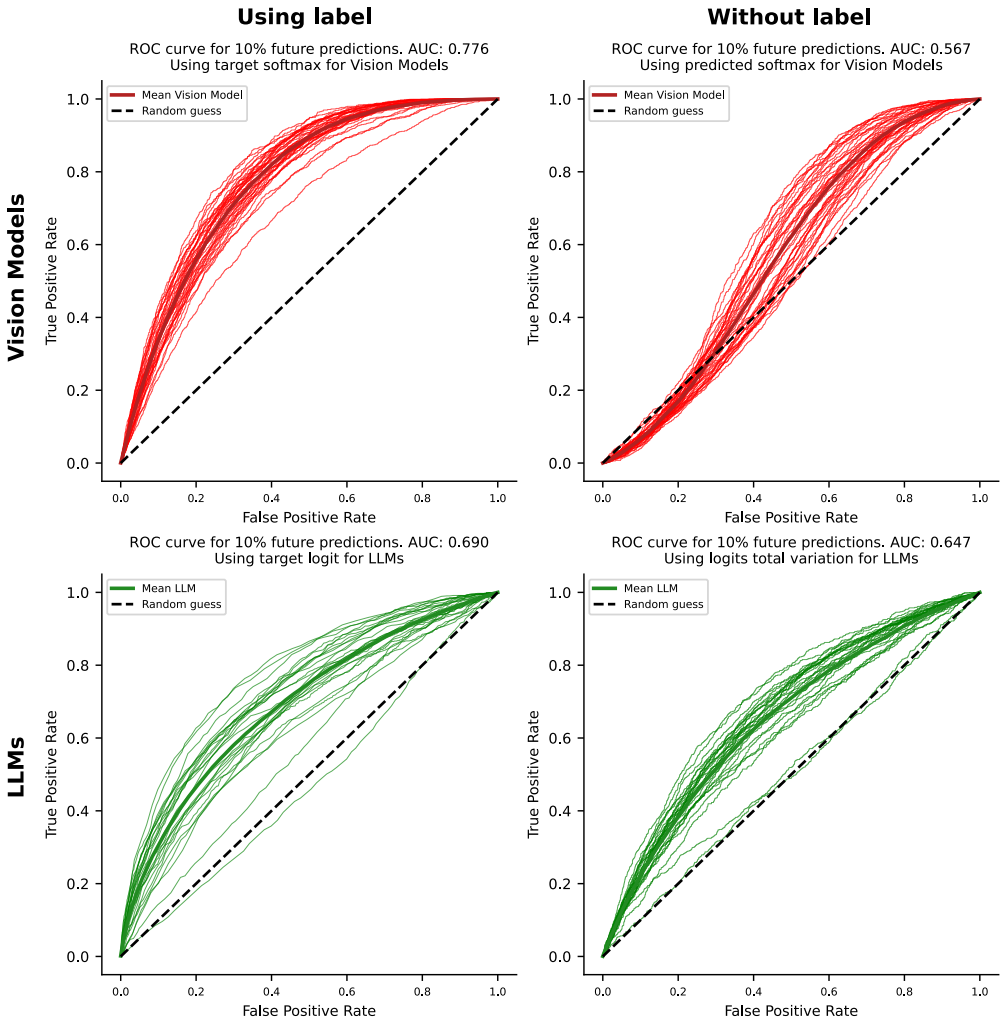


Figure 6. Similar to Figure 5 but this time predicting the next 10% ahead of each model's capabilities.

$$\arg\max_{M \in \{0,1\}^{n \times m}} \quad \mathrm{DIS}(M) \quad \text{s.t.} \sum_{k=1}^{m} M_{ik} = L(i) \;\; \forall i. \tag{6}$$

Thus, we derive the following (the row-sum constraint is omitted for brevity):

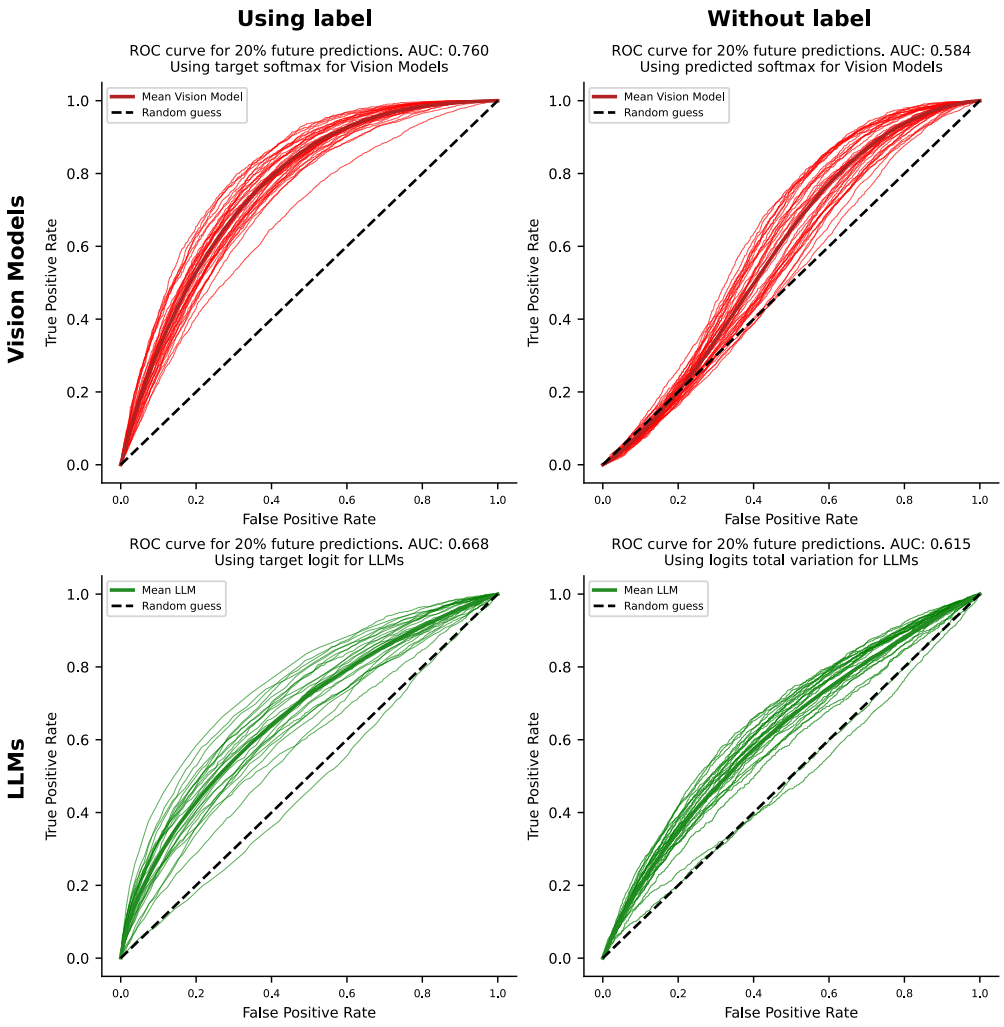


Figure 7. Similar to Figure 5 but this time predicting the next 20% ahead of each model's capabilities.

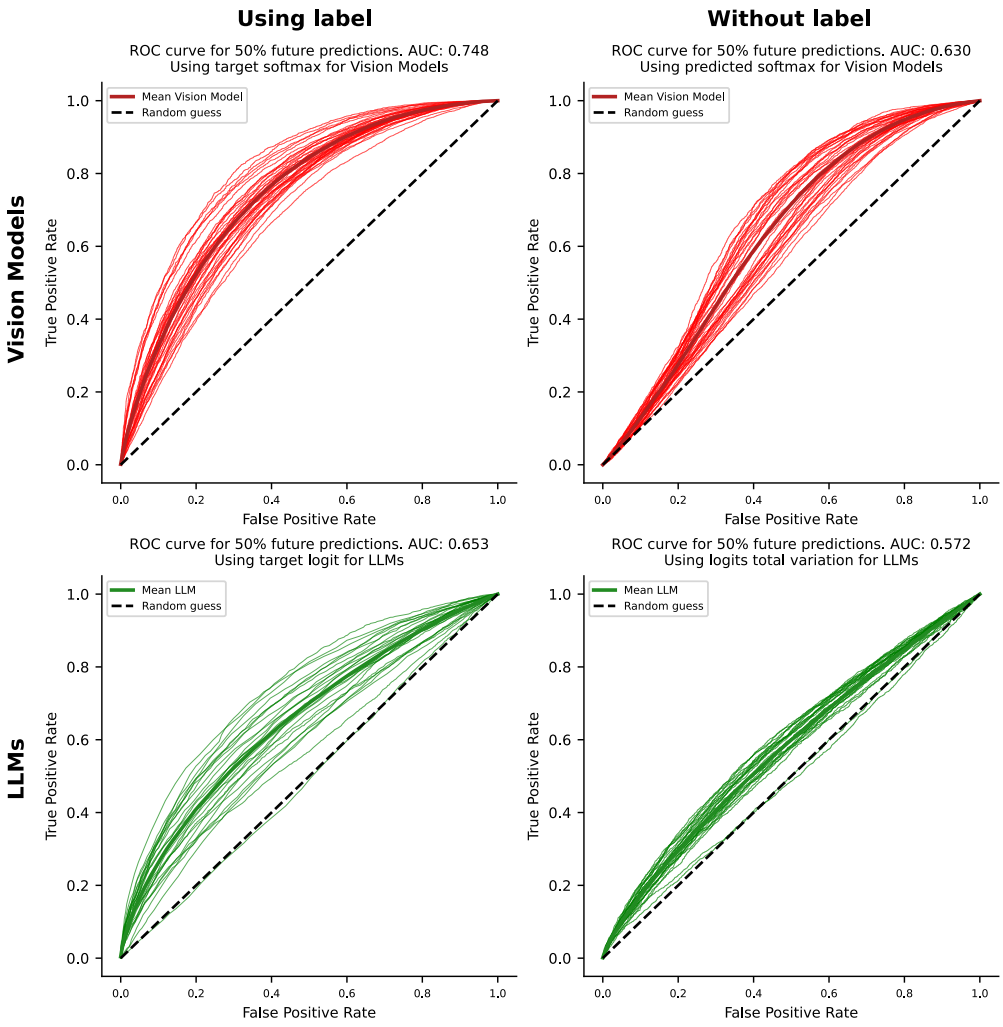


Figure 8. Similar to Figure 5 but this time predicting the next 50% ahead of each model's capabilities.

$$\arg\max_{M\in\{0,1\}^{n\times m}} \mathrm{DIS}(M) = \arg\max_{M} \sum_{k=1}^{m} \sum_{1\le i<j\le n} \mathbf{1}\{M_{j,k}=1,\ M_{i,k}=0\} \tag{7}$$

$$= \arg\max_{M} \sum_{1\le i<j\le n} \left(L(j) - \sum_{k=1}^{m} \mathbf{1}\{M_{j,k}=1,\ M_{i,k}=1\} \right) \tag{8}$$

$$= \arg\max_{M} \left(\sum_{1\le i<j\le n} L(j) \right) - \left(\sum_{k=1}^{m} \sum_{1\le i<j\le n} \mathbf{1}\{M_{j,k}=1,\ M_{i,k}=1\} \right) \tag{9}$$

$$= \arg\max_{M} \underbrace{\left(\sum_{i=1}^{n} (i-1)L(i) \right)}_{\text{const}} - \left(\sum_{k=1}^{m} \binom{t_k}{2} \right) \tag{10}$$

$$= \arg\min_{M} \sum_{k=1}^{m} \binom{t_k}{2} \tag{11}$$

$$= \arg\min_{M} \sum_{k=1}^{m} \left(\frac{1}{2}t_k^2 \right) \underbrace{- \sum_{k=1}^{m} \left(\frac{1}{2}t_k \right)}_{\text{const}} \tag{12}$$

$$= \arg\min_{M} \sum_{k=1}^{m} t_k^2 \tag{13}$$

Where t_k is the column sum of the k-th column. Thus, the matrix M that maximizes $\mathrm{DIS}(M)$ and obtains $Q2_{\text{opposite}}$ is simply obtained by minimizing the sum of squares of the column sums (note that the total number of ones in the matrix is a constant decided by $\sum_{i=1}^{n} L(i)$). Since the column sums are integers and sum to a constant, and by the convexity of the square function, it is trivial to prove via contradiction (as done in A.5) that the minimizer is obtained only when all the elements are at most 1 from each other. Thus, to obtain $Q2_{\text{opposite}}$ we simply create a matrix M that has row sums of $L(\cdot)$ and column sums that are different by no more than 1 from each other and compute $\mathrm{DIS}(M)$.

We construct the optimal M using the following simple algorithm (Algorithm 1). Start the first row at the first column and place $L(1)$ ones contiguously to the right, wrapping around cyclically when you pass the last column. For each subsequent row $i = 2, \ldots, n$, begin one column to the right of where the previous row finished then place $L(i)$ ones contiguously with the same wraparound rule. All remaining entries are zeros. This yields a binary matrix M with the required row sums which maximizes $\mathrm{DIS}(M)$.

Algorithm 1 Construction of M to get $Q2_{\text{opposite}}$

Require: Integers n, m; array $L[1..n]$ with $0 \le L[i] \le m$
Ensure: $M \in \{0,1\}^{n\times m}$ with row i containing exactly $L[i]$ ones

```
M ← zero matrix of size n × m
k ← 1 {indexing starts at 1}
for i ← 1 to n do
    for r ← 1 to L[i] do
        c ← ((k − 1) mod m) + 1
        M[i, c] ← 1
        k ← k + 1
    end for
end forM
```

A.5. Minimizer of $\sum x_i^2$ with fixed $\sum x_i$

Here we quickly prove what vector $\mathbf{x}$ minimizes $\sum x_i^2$ given a fixed $\sum x_i$ and that all x_i are positive integers.

We claim that the vector $\mathbf{x}$ where all elements are at most one away from each other $((\forall\, i, j \in \{1, \ldots, n\})\ |x_i - x_j| \leq 1)$is the minimizer of $\sum x_i^2$ given a constant $\|\mathbf{x}\|_1$. We prove this is true via contradiction, assume

$$\exists\, i, j \in \{1, \ldots, n\} \text{ such that } x_i - x_j \geq 2.$$

$$\text{Define } \mathbf{y} \in \mathbb{N}^n \text{ by} \quad y_i = x_i - 1, \quad y_j = x_j + 1, \quad y_k = x_k \ \ (k \neq i, j).$$

(i.e. we let $\mathbf{y}$ be $\mathbf{x}$ except we bump two elements that have a big gap to be closer to each other) Then $\|\mathbf{y}\|_1 = \|\mathbf{x}\|_1$.

$$\sum_{k=1}^{n} y_k^2 = (x_i - 1)^2 + (x_j + 1)^2 + \sum_{k \neq i,j} x_k^2 = \sum_{k=1}^{n} x_k^2 - 2(x_i - x_j) + 2 \leq \sum_{k=1}^{n} x_k^2 - 2 < \sum_{k=1}^{n} x_k^2,$$

a contradiction to the minimality of $\mathbf{x}$. Thus, the vector where all elements are at most one away from each other is the minimizer of $\sum x_i^2$.

A.6. Derivation of $Q2_{\text{matched}}$

Here we derive how we obtain $Q2_{\text{matched}}$ representing the theoretical maximum (perfect ordering) where higher-accuracy models perfectly subsume lower-accuracy models.

The theoretical perfect ordering is easier to compute compared to the opposite ordering (calculated in Appendix A.4), where the perfect ordering is simply when all the 1's of a column are above all the 0's in the same column. Thus for the i-th row of M (where M, L, and Q_3 are defined in Appendix A.4), we simply place 1's from the first column until the $L(i)$-th column and 0's elsewhere. Since we defined the rows as ordered by non-increasing order of row sums; This construction achieves a $Q_3(i, j) = 0, \forall\, i, j \in \{1, \ldots, n\}\ where\ i < j$. This procedure is shown in Algorithm 2

Algorithm 2 Construction of M to get $Q2_{\text{matched}}$

Require: Integers n, m; array $L[1..n]$ with $0 \leq L[i] \leq m$
Ensure: $M \in \{0, 1\}^{n \times m}$ with row i containing exactly $L[i]$ ones
1: $M \leftarrow$ zero matrix of size $n \times m$
2: **for** $i \leftarrow 1$ **to** n **do**
3: **for** $r \leftarrow 1$ **to** $L[i]$ **do**
4: $M[i, r] \leftarrow 1$
5: **end for**
6: **end for**M

A.7. Derivation of $Q2_{\text{random}}$

Here we provide the derivation of $Q2_{\text{random}}$ which is the value of $Q2$ across all model pairs if the predictions made by each model are independent. To calculate $Q2_{\text{random}}$ we calculate the expected value $Q2$ across all row pairs of a random matrix. We draw rows independently and uniformly from $\{0, 1\}^m$ with exactly $L(i)$ ones. For any column k, $\Pr[M_{i,k} = 1] = L(i)/m$ and for $i \neq j$, rows are independent at fixed k.

We have

$$\sum_{1 \leq i < j \leq n} Q2(i, j) = \sum_{1 \leq i < j \leq n} \sum_{k=1}^{m} \mathbf{1}\{M_{i,k} = 1,\ M_{j,k} = 0\}. \tag{14}$$

By linearity of expectation,

$$\mathbb{E}\left[\sum_{1\le i<j\le n} Q2(i,j)\right] = \sum_{i<j}\sum_{k=1}^{m} \Pr(M_{i,k}=1, M_{j,k}=0) \tag{15}$$

$$= \sum_{i<j} m \cdot \frac{L(i)}{m}\left(1-\frac{L(j)}{m}\right) \tag{16}$$

$$= \sum_{i<j}\left(L(i) - \frac{L(i)L(j)}{m}\right) \tag{17}$$

A.8. Models Used for Future Predictions

In Table 5 we list the 37 vision models which were used for the progress prediction task (depicted in Figure 3 and elsewhere) along with the validation accuracy on ImageNet:

In Table 6 we list the 26 LLMs that were used for the progress prediction task (depicted in Figure 3 and elsewhere) along with the validation accuracy on MMLU-pro (Note that the LLMs were only tasked with outputting a single token which was the answer to the multiple choice question and not given tokens to think, the prompt contained 3 shots of in context examples).

Table 5. Accuracy on Imagenet for Vision Models

	model	Accuracy on ImageNet
0	convnext_large	86.2%
1	convnext_base	85.3%
2	convnext_small	85.1%
3	swin_base_patch4_window7_224	84.8%
4	convnext_tiny	84.1%
5	resnet152	82.6%
6	resnet101	82.0%
7	deit_base_patch16_224	81.9%
8	regnety_032	81.8%
9	wide_resnet50_2	81.5%
10	efficientnetv2_rw_m	81.5%
11	resnext50_32x4d	81.0%
12	vit_base_patch16_224	80.9%
13	swin_tiny_patch4_window7_224	80.9%
14	efficientnetv2_rw_s	80.8%
15	regnety_016	80.6%
16	resnet50	80.1%
17	deit_small_patch16_224	79.4%
18	resnext101_32x8d	79.1%
19	wide_resnet101_2	78.8%
20	regnetx_008	77.5%
21	densenet161	76.8%
22	densenet201	76.3%
23	resnet34	75.9%
24	regnety_008	75.8%
25	mobilenetv3_large_100	75.3%
26	densenet121	75.2%
27	densenet169	75.2%
28	vit_small_patch16_224	74.4%
29	mnasnet_100	74.0%
30	regnetx_006	73.0%
31	mobilenetv2_100	72.6%
32	regnetx_004	71.4%
33	resnet18	70.7%
34	inception_v3	69.6%
35	regnetx_002	67.6%
36	mobilenetv3_small_100	66.6%

Table 6. Accuracy on MMLU-pro for LLMs

	model	Accuracy on MMLU-pro
0	Qwen2.5-14B-Instruct	52.1%
1	Phi-3-medium-4k-instruct	47.3%
2	Qwen3-4B	45.2%
3	Rombos-Qwen-7b	44.6%
4	Qwen2.5-7B	44.5%
5	Gemma-2-9B	43.0%
6	Phi-3.5-mini	41.2%
7	Yi-1.5-9B-Chat	39.7%
8	Qwen2.5-3B	37.5%
9	Mistral-Nemo-Instruct-2407	36.8%
10	Qwen2.5-Coder-7B-Instruct	34.1%
11	Yi-1.5-6B-Chat	33.1%
12	SOLAR-10.7B-Instruct-v1.0	31.7%
13	openchat-3.6-8b-20240522	31.1%
14	Mistral-7B	30.7%
15	OpenHermes-2.5-Mistral-7B	30.2%
16	Nous-Hermes-2-Mistral-7B-DPO	30.0%
17	Llama-3.2-3B-Instruct	29.1%
18	neural-chat-7b-v3-3	28.3%
19	zephyr-7b-beta	28.3%
20	DeepSeek-R1-Qwen-7B	28.1%
21	Qwen2.5-Math-7B-Instruct	27.9%
22	Gemma-2-2B	27.4%
23	aya-23-8B	23.6%
24	Llama-3.2-1B-Instruct	17.6%
25	deepseek-coder-6.7b-instruct	16.7%