The Unreasonable Effectiveness of Easy Training Data for Hard Tasks

Peter Hase[1,2] **Mohit Bansal**[2] **Peter Clark**[1] **Sarah Wiegreffe**[1]
[1]Allen Institute for AI [2]UNC Chapel Hill
{peter, mbansal}@cs.unc.edu, peterc@allenai.org, wiegreffesarah@gmail.com

Abstract

How can we train models to perform well on hard test data when hard training data is by definition difficult to label correctly? This question has been termed the *scalable oversight* problem and has drawn increasing attention as language models have continually improved. In this paper, we present the surprising conclusion that current pretrained language models often generalize relatively well from easy to hard data, even performing as well as oracle models finetuned on hard data. We demonstrate this kind of easy-to-hard generalization using simple finetuning methods like in-context learning, linear classifier heads, and QLoRA for seven different measures of datapoint hardness, including six empirically diverse human hardness measures (like grade level) and one model-based measure (loss-based). Furthermore, we show that even if one cares most about model performance on hard data, it can be better to collect easy data rather than hard data for finetuning, since hard data is generally noisier and costlier to collect. Our experiments use open models up to 70b in size and four publicly available question-answering datasets with questions ranging in difficulty from 3rd grade science questions to college level STEM questions and general-knowledge trivia. We conclude that easy-to-hard generalization in LMs is surprisingly strong for the tasks studied.[1]

1 Introduction

It is difficult to supervise LMs (i.e., train LMs to give correct outputs) in specialized domains of human knowledge, because it is difficult to correctly label data in such domains. Labeling difficulty manifests itself in both time to annotate (and thus cost) and label noise (Lease, 2011; Northcutt et al., 2021). Labeling difficulty becomes severe when specific expertise is required (Sambasivan et al., 2021). For example, for sufficiently specific physics problems, PhD holders and PhD students can make errors on as many as 40% of (objective) problems (Rein et al., 2023). As more NLP benchmarks focus on challenging domain-specific tasks, having access to large human-labeled training corpora may become increasingly infeasible (e.g., existing benchmarks like MMLU (Hendrycks et al., 2021) and GPQA (Rein et al., 2023) do not come with training data). The question arises: how can we train models to solve hard problems when correctly labeling enough hard data for training is difficult? This problem is an example of the scalable oversight problem, which concerns how to give a good reward signal to a model when it is difficult to assess if its outputs are correct (Amodei et al., 2016).

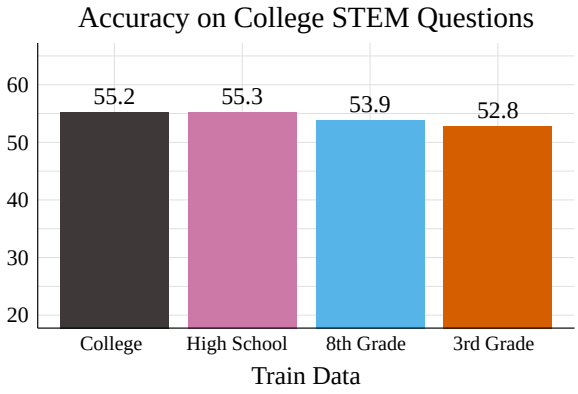

Figure 1: A model prompted with easy data (e.g., 3rd Grade problems) does *almost as well* on a hard task (College problems) as a model prompted with hard data (the College bar). Results shown for Mixtral-8x7B with k=10 prompt examples, averaged over 5 random seeds.

In this paper, we study the problem of **easy-to-hard generalization**. Easy-to-hard generalization refers to model performance on hard test data when finetuned[2] only on easy training data, defined according to some human hardness measure (like

[1]Our code is publicly available at: https://github.com/allenai/easy-to-hard-generalization.

[2]We use "finetuning" interchangeably with "training" and "fitting" to refer to fitting pretrained models to data via in-context learning (ICL), parameter efficient finetuning (QLoRA), or by training a linear classifier head.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7002–7024

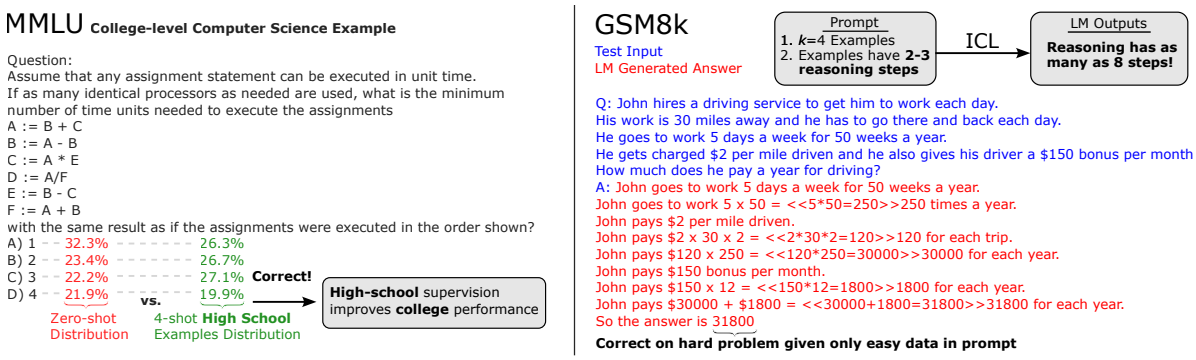

Figure 2: Supervising Llama-2-70b with *easy* data (left: high school level computer science problems; right: math problems with 2-3 reasoning steps) can enable generalization to *hard* data (left: a college level computer science problem; right: a math problem involving 8 reasoning steps). Prompts themselves are omitted for brevity.

grade level). Since gathering data in domains like graduate level STEM fields is expensive and time-consuming, it would clearly be useful if we could improve model performance in these domains by only finetuning models on cleanly labeled data from simpler domains, like high school STEM questions. To assess how well current LMs generalize from easy to hard data, we fit models to easy data and test them on hard data ("easy-to-hard"), then compare them to an oracle upper bound and unsupervised lower bound. The oracle upper bound is a model that has access to labeled hard data for finetuning ("hard-to-hard"), while the unsupervised lower bound is a model that is prompted zero-shot to answer questions ("unsupervised-to-hard"). The metric we are interested in is the **Supervision Gap Recovered (SGR):**

$$\frac{\text{Easy} - \text{Unsupervised}}{\text{Hard} - \text{Unsupervised}}$$

where Easy, Hard, and Unsupervised refer to model performance *on hard test data* when finetuned on easy data, hard data, or no data (zero-shot), respectively. This metric takes a value of 100% when finetuning on easy data is as effective as hard data, and it is 0% when a model finetuned on easy data is no better than prompting a model zero-shot.

Our main result is that pretrained language models generalize surprisingly well from easy to hard data, often performing almost as well as an "oracle" model finetuned on hard data (illustrated in Fig. 1). In experiments with ARC (Clark et al., 2018), MMLU (Hendrycks et al., 2021), GSM8k (Cobbe et al., 2021), and StrategyQA (Geva et al., 2021), we find that the Supervision Gap Recovered is usually **between 70% and 100%**, meaning that easy supervision is at least 70% as good as hard supervision for hard test performance (see Fig. 2 for example problems). These results are robust across (1) model family and scale (between 7b and 70b parameters), (2) six different human hardness measures and a model-based measure, (3) four datasets/tasks, and (4) several finetuning methods including in-context learning with and without chain-of-thought reasoning (Brown et al., 2020; Wei et al., 2022), QLoRA (Dettmers et al., 2023), and linear classifier heads (Belinkov, 2022). Overall, our results suggest that current LMs generalize relatively well to test data across human difficulty levels even when finetuned on data that is measurably easier than the test data. We hypothesize that this occurs because easy data elicits latent knowledge and skills from pretrained models *in a hardness-invariant way*.

We additionally demonstrate that easy supervision can outperform hard supervision when (1) within some data collection budget, a greater quantity of easy data can be collected than hard data, or (2) easy data can be labeled with lower error rates than hard data (Sec. 5.3). Lastly, we study how easy-to-hard generalization changes with model scale and the gap between train and test hardness (Sec. 5.4). The remainder of the paper is organized along the following research questions:

RQ1: How Can We Measure Data Hardness?

RQ2: Can We Do Well on Hard Data by Training on Easy Data?

RQ3: What Are the Cost-Benefit Tradeoffs of Collecting Easy vs. Hard Training Data?

RQ4: Is Easy-To-Hard Generalization Consistent Across Model Scale and Train-Test Hardness Gap Size?

We summarize our main conclusions below:

1. Our six human hardness measures and one model-based measure are empirically diverse

and model performance declines on harder test data for each measure.

2. LMs generalize surprisingly well from easy-to-hard data, closing 70%-100% of the gap between unsupervised and hard train data.
3. We show that it is often better to train on easy data when hard data is more expensive to collect or has noisier labels.
4. The Supervision Gap Recovered is highly robust across model scale. Easy-to-hard performance may begin to decline when the train-test hardness gap is large enough.

2 Related Work

Curriculum Learning. Curriculum learning has historically concerned itself with model performance on hard data points (Bengio et al., 2009). Previous work in this area has argued that learning from easy data first is helpful for models later learning more complex concepts and therefore performing better on hard data (Xu et al., 2020). Whereas curriculum learning aims to improve hard test performance by optimally ordering the training data, we simply aim to investigate how well models generalize to hard data based on the hardness of the training data. Our results suggest that pretrained LMs generalize surprisingly well from easy to hard data, potentially alleviating the need for heavily engineered training curricula.

Compositional Generalization. Work in compositional generalization has previously shown that neural networks struggle to generalize to problems that require combining reasoning steps in ways not seen exactly during training (Lake and Baroni, 2018; Bogin et al., 2022; Zhou et al., 2023b). Further work has begun characterizing the conditions under which models generalize to compositionally more difficult problems. For instance, Transformers will generalize better on classes of algorithmic problems whose solutions can be written in RASP, meaning the programs can be implemented exactly by a Transformer forward pass (Zhou et al., 2023a). Recurrent test-time computation also appears to be quite valuable for generalizing to problems requiring more reasoning steps than those seen during training (Schwarzschild et al., 2021; Bansal et al., 2022). Interestingly, even GPT-3.5 with Chain-of-Thought prompting can struggle to generalize to simple mathematical problems requiring more reasoning steps than seen during finetuning (Dziri et al., 2023). Our results are not inconsistent with these studies, but instead demonstrate that *relative to* an unsupervised-to-hard lower bound and hard-to-hard upper bound, easy-to-hard performance on compositional reasoning problems is often surprisingly good (Sec. 5.2).

Easy-to-Hard Generalization. Amodei et al. (2016) motivate the scalable oversight problem by pointing out how it could be challenging to give a proper reward signal to a model when it is difficult to assess if its outputs are correct. Assessing easy-to-hard generalization provides useful context for understanding the difficulty of the scalable oversight problem, as it tells us how we can expect models to generalize from a setting where we can properly supervise them to one where we cannot. Past work evaluates easy-to-hard generalization in NLP using model-based hardness measures (Swayamdipta et al., 2020) and number of compositional reasoning steps required to solve a problem (Fu et al., 2023). Swayamdipta et al. (2020) show that BERT models perform worse on commonsense reasoning tasks when finetuned on easy data rather than hard data according to a loss-based metric resembling minimum description length (Perez et al., 2021). Fu et al. (2023) show a similar result with GPT3 models for StrategyQA and GSM8k, finding that prompting with "complex" examples does better than "simple" examples, where examples are categorized according to the number of reasoning steps in the annotated human reasoning chain for a problem. Relative to these works, we study easy-to-hard generalization with (1) a greater number of human hardness measures, including grade level, expert rating, required cognitive skills, question length, answer length, and number of reasoning steps, as well as a model-based metric, (2) multiple datasets involving science question answering, compositional reasoning, and mathematical reasoning, and (3) multiple model sizes for understanding scaling trends. In contrast to these works, we show that in a number of settings easy-to-hard generalization is comparable to hard-to-hard generalization.

In concurrent work, Burns et al. (2023) present results on a related "weak-to-strong" generalization problem, where a stronger model is finetuned using labels from a weaker model. They also explore easy-to-hard generalization for NLP tasks using a model-based hardness measure. In contrast to this work, (1) we define our main performance metric (Supervised Gap Recovered) using an unsupervised model as the baseline performance rather than a

weaker model as the baseline performance, which is important when an unsupervised stronger model will greatly outperform a supervised weaker model (as is observed in our experiments); (2) we use human hardness measures in addition to model-based hardness, which is a more realistic and important setting when the two may not correlate strongly (see our Fig. 8); and (3) we use publicly available datasets and open-source models rather than unidentified "NLP tasks" and API-gated models.

3 Measuring Datapoint Hardness

Measuring easy-to-hard generalization requires drawing a distinction between easy and hard data, defined in terms of human ability to correctly label the data. There could be many ways to describe what makes problems harder, including that (1) only people with specialized training and knowledge can solve the problem (Lehman et al., 2019); (2) it takes people longer to solve the problem; (3) people are less certain that their final solution is correct; (4) people with similar expertise naturally disagree about the solution to the problem, while agreeing that there is an objective solution (Dumitrache et al., 2018; Pavlick and Kwiatkowski, 2019; Nie et al., 2020); (5) experts know of a reliable method for obtaining the answer to a problem, but it is costly in terms of time and effort or possibly noisy in its outputs (like conducting scientific experiments). In this paper, we aim to capture the above properties in a number of specific measures we can obtain for each instance in our datasets, including:

1. **Education/Grade Level:** What education level (possibly in a particular domain) would typically lead one to be able to answer the question?
2. **Expert Rating:** How difficult would an expert rate the question, on an ordinal scale?
3. **Required Cognitive Skill:** What cognitive skills are required by the question? This rating is based on Bloom's cognitive skills taxonomy, in order of increasing complexity: (1) Remembering, (2) Understanding, (3) Applying, (4) Analyzing, and (5) Evaluating (Bloom et al., 1956; Adams, 2015).
4. **Question Num. Words:** Question length is a natural proxy for question hardness, as longer questions can involve more premises or a greater number of concepts.
5. **Answer Num. Chars:** We also consider Answer Num. Chars, since longer answers may reflect more specific or more complex problems. Character count provides a measure that is applicable across tasks.
6. **Compositional Steps:** Compositional reasoning is more difficult than executing individual reasoning "primitives." We consider how many individual reasoning steps are involved in answering a question (i.e., the number of subproblems whose solutions must be combined), according to human-annotated reasoning chains.
7. **Minimum Description Length:** A model-based measure of hardness, measuring datapoint loss under a model family. Details for MDL computation are given in Appendix A.

The first six hardness measures are fundamentally human notions of hardness, but we can also measure a model-based metric for datapoint hardness. In this direction, the seventh measure is a minimum-description-length (MDL) metric (Voita and Titov, 2020). In practice, MDL can be measured by computing a *test* datapoint's average label probability across models of identical architecture finetuned on increasing quantities of training data for a task (Perez et al., 2021). Intuitively, MDL captures how hard on average an in-distribution test datapoint is for a model to generalize to given some amount of training data. Ultimately, we use our MDL metric to capture how well a *stronger* model generalizes to data that is hard according to a *weaker* model, in order to simulate a setting where humans cannot label hard problems that they would like for a strong model to solve.

In our experiments, we use four datasets that contain instance-level annotations for some portion of these measures, as shown in Table 1.

- **ARC** (Clark et al., 2018): U.S. gradeschool science questions in multiple-choice format. We combine ARC-Easy and ARC-Challenge splits. Random performance is 25%.
- **MMLU** (Hendrycks et al., 2021): Domain-specific multiple-choice questions for many domains. We subset to high school and college level math, physics, biology, chemistry, and computer science questions (MMLU-STEM-5). Grade level is high school (HS) vs. college. See Figure 2 (left) for an example. Random performance is 25%.

ARC	MMLU-STEM-5	StrategyQA	GSM8k
$n = 4521$	$n = 1746$	$n = 2290$	$n = 8792$
Grade Level (3-8)	Grade Level (HS vs. College)	Grade Level	Grade Level
Difficulty Score (1-3)	Difficulty Score	Difficulty Score	Difficulty Score
Bloom Skill (1-5)	Bloom Skill	Bloom Skill	Bloom Skill
Question Num. Words	Question Num. Words	Question Num. Words	Question Num. Words
Answer Num. Chars	Answer Num. Chars	Answer Num. Chars	Answer Num. Chars
Num. Reasoning Steps	Num. Reasoning Steps	Num. Reasoning Steps	Num. Reasoning Steps
MDL	MDL	MDL	MDL

Table 1: Hardness measures we use for each dataset. Grayed-out options are not present in the dataset's annotations, and thus not used in our experiments.

- **StrategyQA** (Geva et al., 2021): Yes/no general knowledge trivia questions requiring compositional reasoning over individual facts. The "Num. Reasoning Steps" measure is the number of facts that must be combined. Majority-class vote performance is 53.9%.
- **GSM8k** (Cobbe et al., 2021): U.S. grade school math word problems in direct answer format (i.e., no answer choices given). Random performance is 0%. The number of steps in a problem solution is the "Num. Reasoning Steps" measure and is obtained from the human-annotated reasoning chain collected for each problem. See Figure 2 (right) for an example.

There are generally fewer hard datapoints than easy datapoints in our datasets, given the relative difficulty of collecting hard data. In MMLU-STEM-5, for example, there are 603 college level questions and 1143 high school questions. We show histograms for each hardness measure distribution in Appendix Fig. 20. See Appendix C for further dataset information.

4 Experiment Setup

Models. Apart from Fig. 1, we report results in the main paper on the Llama-2 70b base model (Touvron et al., 2023b). In Appendix B, we show results for Llama-2 7b and 13b, an RLHF version of Llama-2-70b ("Llama-2-chat"), Qwen-72b (Bai et al., 2023), and Mixtral-8x7b (Jiang et al., 2024).

Data Hardness Stratification. To separate datasets into easy and hard data (with leftover data being medium data), we define easy/hard cutoffs as follows: for Question Num. Words, Answer Num. Chars, and MDL, we automatically define these values to be at the 30th and 70th percentiles of the variable range. Other variable cutoffs are defined manually: For **ARC**, *Grade Level* is easy (3-5), medium (6-7), hard (8); *Difficulty Score* is easy (1), medium (2), hard (3); *Bloom Skill* is easy (1-2), medium (3), hard (4-5). For **MMLU**, *Grade Level* is easy (high school) and hard (college) with no medium. For **StrategyQA**, *Num. Reasoning Steps* is easy (1-2), medium (3), hard (4-5). For **GSM8k**, *Num. Reasoning Steps* is easy (2-3), medium (4-5), hard (6-11). We show histograms for each hardness measure distribution in Appendix Fig. 20.

Finetuning Methods. We fit models to data with in-context learning (ICL; Brown et al., 2020), linear classifiers trained on frozen model hidden states (Belinkov, 2022), or QLoRA (Dettmers et al., 2023). StrategyQA and GSM8k benefit heavily from utilizing chain-of-thought reasoning (CoT; Wei et al., 2022), so we primarily conduct experiments for these datasets with ICL+CoT and QLoRA+CoT (using reasoning chains from the datasets for supervision). See descriptions of each method below, with full detail in Appendix D.

1. **ICL**: For in-context learning (ICL), we use k=10 prompt examples for ARC and MMLU and k=8 examples for StrategyQA and GSM8k (we see diminishing returns for larger k). When scoring multiple choice questions (no CoT), we get a model prediction by computing the answer probability for each answer choice given the test input and the prompt. When generating outputs with CoT, we greedily generate up to $t = 100$ tokens for StrategyQA and $t = 300$ tokens for GSM8k. Accuracy is computed as exact match between predicted answer and label.
2. **Linear Probing**: We train a linear classifier on frozen LM hidden states. This is an effective method for performing multiple choice QA using LM representations (Liu et al., 2023), and it does not require any finetuning of the underlying LM. For a given question, we compute one representation per answer choice

by concatenating the question and answer choice as input and extracting the model's final-token representation. Then, we score each representation z by applying the linear probe: $f(z; w) = w^T z$. The answer choice with the highest score is returned as the prediction. The probe weight w is trained using SGD to minimize cross-entropy loss on a dataset of representations $Z = \{\{z_{i,j}\}_{j=1}^{j=|A|}\}_{i=1}^{N}$ derived from N training datapoints with $|A|$ answer choices.

3. **QLoRA**: To finetune our LMs, we execute QLoRA with the LoRA implementation from HuggingFace `peft` (Mangrulkar et al., 2022) and the 8-bit AdamW from `bitsandbytes` (Dettmers et al., 2022). We train the default layers for Llama-2 with rank $r = 16$ adapters, $\alpha = 32$, and dropout $p = 0.1$. Model predictions are obtained in the same manner as for ICL, i.e., by scoring multiple choice options or generating $t = 100/300$ tokens for StrategyQA/GSM8k.

Unsupervised Baseline. Our unsupervised baseline is zero-shot prompting, scoring the answer choice probabilities given the question and taking the highest probability answer as the model prediction. The one exception to this is for GSM8k, which does not have multiple answer choices per question. For this dataset, we use a simple "Let's think step by step" style prompt. See Appendix D.

Training Size Controls. For all experiments with linear probing and QLoRA, we use $n = 160$ train points. While we would prefer to use more finetuning data, the bottleneck we face is that fairly comparing easy-to-hard with hard-to-hard generalization requires both fixing the amount of finetuning data and leaving enough hard data left over for testing. Since we have as few as $n = 603$ hard test points for MMLU, we have to limit finetuning data to $n = 160$ points to leave enough test data for reasonably small confidence intervals.

Statistical Testing. We perform experiments using 5 random seeds, controlling the training data selection (leaving remaining data for testing). To obtain confidence intervals and p-values, we use block bootstrap sampling (Efron and Tibshirani, 1994). See Appendix E for further detail.

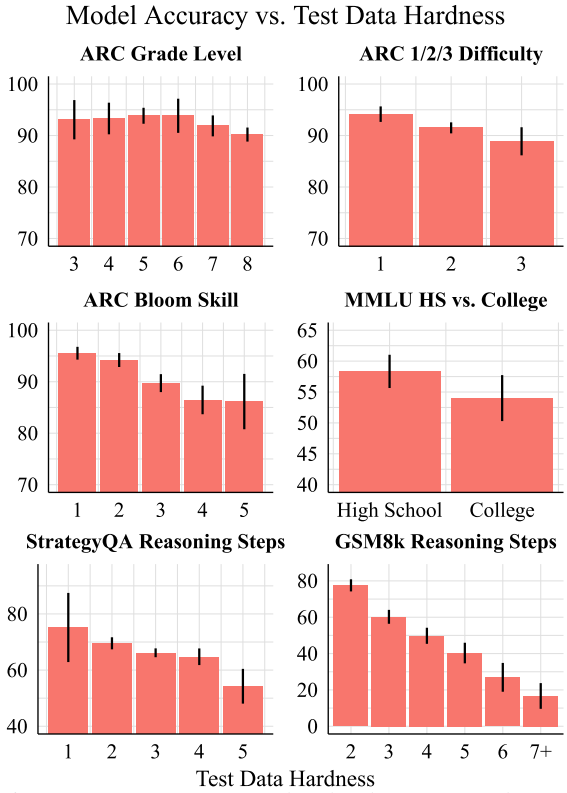

Figure 3: Accuracy vs test data hardness across datasets (using Llama-2-70b with ICL). Data that humans find harder is also harder for LMs. Error bars are 95% CIs showing test sample variance.

5 Experiments

5.1 RQ1: How Can We Measure Data Hardness?

We first explore properties of our hardness measures for each dataset. Here, we focus on hardness measures unique to each dataset, with full results across all measures shown in Appendix B.

Design. While our human hardness measures are direct measurements of data hardness, we validate that each measure is meaningful by assessing model performance across test hardness levels, using Llama-2-70b and ICL with randomly sampled prompts. We also create correlation heatmaps for hardness measures in our datasets, using a Spearman rank-order correlation (Spearman, 1987) between hardness values for each datapoint.

Results. It appears that all of our hardness measures meaningfully capture some aspect of datapoint hardness, as model accuracy declines for harder test data for each of these measures (Fig. 3), including for model-based hardness as measured by an ensemble of 7b-parameter models (Appendix Fig. 9). This also holds when finetuning with QLoRA (Appendix Fig. 10). Next, we find that

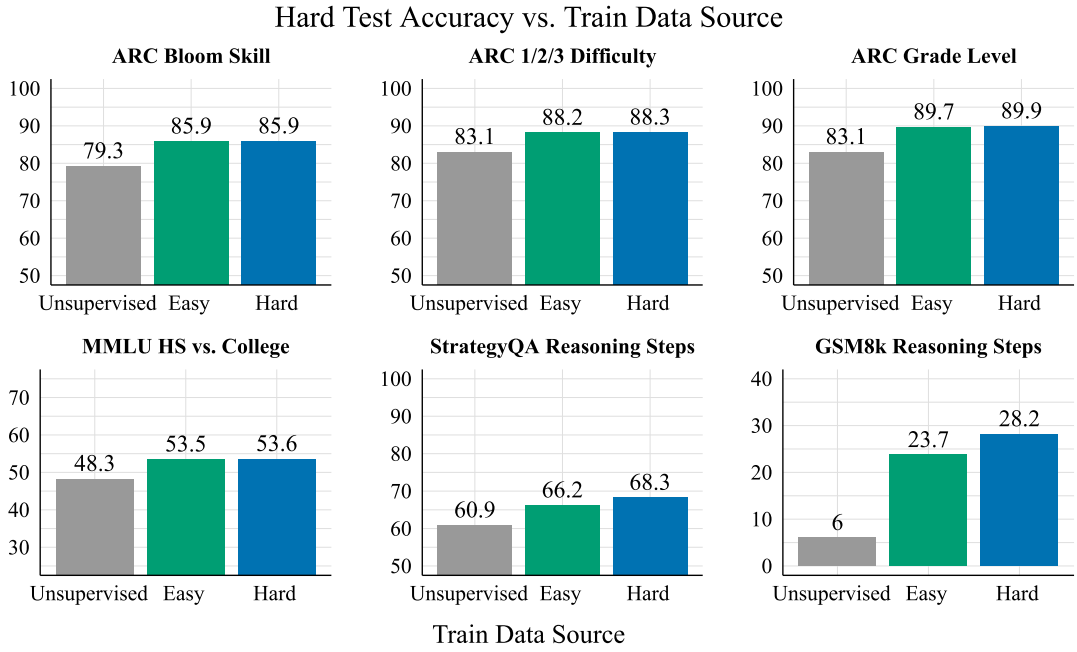

Figure 4: Accuracy on hard test data vs train data hardness (using Llama-2-70b and ICL, averaged over 5 seeds). Models recover 70-100% of the supervision gap (between Unsupervised and Hard) when finetuned on Easy data.

our hardness measures are empirically very diverse (Appendix Fig. 8). Correlations between hardness measures are fairly low, suggesting that these measures capture different possible aspects of datapoint hardness. We conclude that easy-to-hard generalization should be assessed with multiple notions of datapoint hardness, since there are several different available measures and model performance declines for harder test data along each measure.

5.2 RQ2: Can We Do Well on Hard Data by Training on Easy Data?

We now examine how well models generalize from easy training data to hard test data.

Design. For each of our hardness measures, we test models on exclusively hard test data (according to that hardness measure), while varying whether they are finetuned on easy or hard data.[3]

Results. Surprisingly, **Llama-2-70b with ICL shows comparable generalization to hard test data regardless of whether it is fit to easy or hard data** (Fig. 4). In fact, across all six hardness measures, the **Supervision Gap Recovered is between 70% and 100%**. These results are statistically significant, with CIs and p-values shown in Appendix Table 2. Interestingly, for ARC and MMLU, there is *no difference* in easy vs. hard generalization using ICL. Results are also robust across finetuning methods and additional hardness measures (Appendix Figs. 13, 18). With QLoRA, for example, the SGR remains within 70%-100% for ARC, MMLU and StrategyQA. While GSM8k appears to exhibit worse easy-to-hard generalization, we note that easy-to-*all* generalization is actually equally good to hard-to-*all* generalization (see Fig. 11). Thus it seems like easy data provides surprisingly good supervision for LMs.

These results contrast notably with past work in curriculum learning and compositional generalization (Bengio et al., 2009; Lake and Baroni, 2018). This is likely because models like Llama-2-70b have learned much more during pretraining than models commonly used in work on curriculum learning and compositional generalization. So, it would seem that finetuning these models on relatively small amounts of easy data successfully elicits the relevant task knowledge from the models in a way that is largely invariant to datapoint hardness.

5.3 RQ3: What Are the Cost-Benefit Tradeoffs of Collecting Easy vs. Hard Training Data?

One implication of the results from Sec. 5.2 is that if easy data is almost as good as hard data, it could be better to collect and fit to easy data, since hard data can be noisier and costlier to collect (Samba-

[3]We report test accuracy on the full data distribution and the easy test split in Appendix Figs. 11 and 12, respectively.

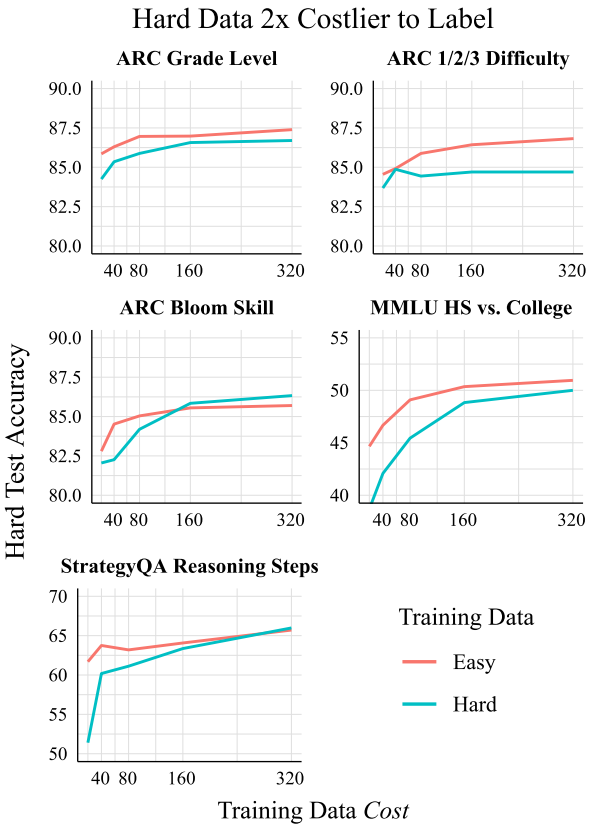

Figure 5: Hard test accuracy is often higher when training on comparable amounts (costs) of easy rather than hard data. Here, we suppose hard data is 2x costlier to collect. Results are for Llama-2-70b with linear probes.

sivan et al., 2021). Hence, we test the hypothesis that finetuning on easy data outperforms hard data under two possible assumptions: (1) that one can collect more easy data than hard data given a fixed budget (i.e., time, money), and (2) that easy data is less noisily labeled than hard data.

Design. For (1) the budget assumption, we fit linear probes on either easy or hard data using datasets of sizes in {10, 20, 40, 80, 160, 320}. We then show hard data test performance vs. training *cost*, assuming hard data costs twice as much as easy data to collect, meaning labeling 40 easy training points is equivalent in cost to labeling 20 hard training points. For (2) the noise assumption, we assume that easy data is mislabeled $p\%$ of the time, while hard data is mislabeled $2p\%$ of the time. Here, we measure test performance on hard data given different values of p. Note the 1:2 data collection cost ratio is almost exactly the ratio observed in MMLU-STEM-5, which contains 603 college level questions and 1143 high school questions, and a 1:2 labeling error ratio is plausible as well given expert human accuracy on datasets like MMLU (estimated

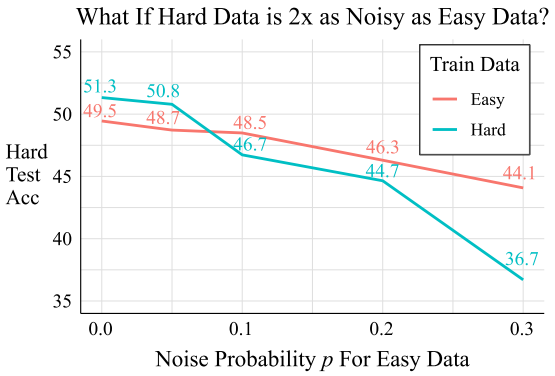

Figure 6: When hard data has noisier labels than easy data, finetuning on easy data can give better hard test performance (shown for MMLU-STEM-5 using Llama-2-70b with a linear probe).

at 89.8%) and GPQA (estimated at $\leq$72% for difficult graduate level STEM questions) (Hendrycks et al., 2021; Rein et al., 2023).

Results. We show results for the data budget assumption in Fig. 5. In terms of hard test accuracy, there is more often than not an advantage to fitting a model to easy data rather than hard data when the cost ratio between them is 1:2.

For the noise assumption, we draw a similar conclusion based on the results for MMLU in Fig. 6. For instance, easy data is preferable when its labeling error rate is 10% (or higher), meaning the error rate for hard data is 20% (or higher). Since high error rates are possible for difficult domain questions (Rein et al., 2023), there are plausible settings where it is better to finetune on easy data than hard data due to label noise.

5.4 RQ4: Is Easy-To-Hard Generalization Consistent Across Model Scale and Train-Test Hardness Gap Size?

We consider two questions likely to be relevant as models become more capable: (1) how does the Supervision Gap Recovered change as models scale in size, and (2) how does hard test performance change as the gap between train and test hardness grows? We are interested in these questions because in various settings AI performance may exceed human expert performance, and we want to know whether it will become more and more difficult to supervise models as this occurs.

Design. For question (1), we measure easy-to-hard and hard-to-hard performance on MMLU for Llama-2 at three different model sizes: 7b, 13b, and 70b. For question (2), we test models on hard MMLU data (college STEM questions), while

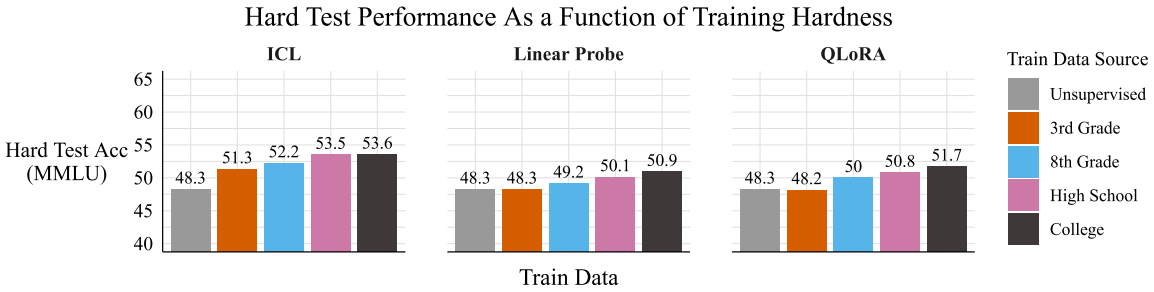

Figure 7: Performance on hard test data may begin to decline when finetuning on very easy data. Results shown for college-level STEM questions using Llama-2-70b (see other hardness measures, models in Appendix Figs. 16, 17).

finetuning them on *even easier* datasets than high school MMLU: 3rd and 8th grade ARC questions.

Results. First, we find that **models show similar levels of Supervision Gap Recovered across sizes** (Appendix Fig. 15). SGR is near 100% for model sizes between 7b and 70b. For our second question, we find that the difference between train and test hardness may have an effect on test performance (Fig. 7). Across methods, we see some decline in generalization once the gap between train and test hardness becomes sufficiently large. However, even 3rd grade supervision can be surprisingly effective for college STEM questions (e.g. SGR falls from 74% to 57% when fitting to 8th grade vs. 3rd grade questions using ICL). In an evaluation across our other hardness measures, easy training data is only marginally worse than medium training data (see Appendix Fig. 16). Together these results suggest that, while (1) easy supervision remains effective as models scale up, (2) easy-to-hard performance may begin to decline when the gap between train and test hardness becomes sufficiently large.

6 Discussion

Are Our Tasks Hard Enough to Provide Generalizable Easy-To-Hard Results? Benchmark datasets for LMs now require specialized domain expertise (Hendrycks et al., 2021). The largest difficulty gap that we test in this paper is between 3rd grade and college level STEM questions (using ARC and MMLU). Concurrent work has called for work studying gaps as large as 3rd grade to 12th grade (described as "huge leaps in generalization"; Burns et al., 2023). Therefore, we see our results as relevant for future work that may operationalize "easy" and "hard" differently.

How Do LMs Solve Hard Problems From As Few As Ten Easy Examples? Our results suggest that finetuning on even small amounts of easy data successfully elicits relevant knowledge from LMs in a way that is largely invariant to datapoint hardness (we do not conclude that we are teaching LMs entirely new skills). This could be because this kind of finetuning encourages models to answer questions based on "truthfulness" representations of text, which should be invariant across domain and data hardness (see Marks and Tegmark, 2023). We emphasize that we do not interpret our results as models merely "learning the task format" as opposed to true generalization: we also fit models using ICL prompts that are trivially simple but match the task format for MMLU and StrategyQA, and find that model performance varies based on prompt data hardness and not simply prompt task format (see Appendix Fig. 19). Hence it appears that fitting to easy data encourages models to give correct outputs for hard questions.

7 Conclusion

We study the problem of easy-to-hard generalization, showing that (1) several meaningful human and model-based hardness measures disagree about which data is hardest; (2) LMs trained on easy data often perform nearly as well as those trained on hard data, recovering 70-100% of the Supervision Gap between an unsupervised lower bound and hard-to-hard upper bound; (3) practically, one can perform better on hard test data by collecting and training on easy data rather than hard data when the hard data is noisier or costlier to collect; and (4) SGR may begin to decline when the gap between train and test hardness becomes sufficiently large. These results are robust across datasets, training methods, hardness measures, and model size. Our findings suggest that the scalable oversight problem may be easier than previously thought.

Limitations

We aim to study how models generalize from settings where humans can easily label data to those where humans have difficulty labeling data. Since we require ground truth labels to evaluate model generalization, this limits our evaluations to datasets where humans *have* reliably labeled the data. While we aim to test a broad range of train and test difficulties (like 3rd grade training data and college-level test data), our results may not generalize to settings with difficulty levels besides those we test in this paper, especially test settings where almost no amount of human effort can reliably produce accurate data labels (like unsolved scientific questions).

Additionally, we are not able to verify that test questions for ARC, MMLU, StrategyQA, and GSM8k are not in training datasets of the open models used in this paper, including Llama-2-70b and Mixtral 8x-7b. This means we cannot be certain that models are generalizing from easy to hard data, rather than reporting memorized hard question answers. While this concern currently affects all research on open source LLMs, including those with public datasets (since finding test set contamination in pretraining data is an unsolved problem), we note that it would be beneficial for future work to evaluate easy-to-hard generalization on test sets that are either private or collected after pretraining data collection cut-offs.

Ethics Statement

We hope that positive results in easy-to-hard generalization imply that we can train models to perform well in niche domain like biology, chemistry, medicine, law, engineering, etc., without demanding significant amounts of expert time and spending large amounts of money in order to annotate data in these settings. In this way, we might make LLMs more useful for hard tasks while requiring less human effort to supervise their training. At the same time, we note that there are dual use and human labor displacement concerns around improving model capabilities in such domains. Ultimately we hope for LLMs to be deployed responsibly, so that they can be used to further human values and not for any ill intent.

References

Nancy E Adams. 2015. Bloom's taxonomy of cognitive learning objectives. *Journal of the Medical Library Association: JMLA*, 103(3):152.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. 2022. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. *Advances in Neural Information Processing Systems*, 35:20232–20242.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Benjamin S Bloom, Max D Engelhart, Edward J Furst, Walker H Hill, and David R Krathwohl. 1956. *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain*. McKay New York.

Ben Bogin, Shivanshu Gupta, and Jonathan Berant. 2022. Unobserved local structures make compositional generalization hard. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2731–2747, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan

Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *9th International Conference on Learning Representations, ICLR*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Anca Dumitrache, Lora Aroyo, and Chris Welty. 2018. Crowdsourcing ground truth for medical relation extraction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–20.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. 2023. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Bradley Efron and Robert J Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC press.

Erich Elsen, Augustus Odena, Maxwell Nye, Sağnak Taşırlar, Tri Dao, Curtis Hawthorne, Deepak Moparthi, and Arushi Somani. 2023. Releasing Persimmon-8B. Blogpost, ADEPT AI.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Matthew Lease. 2011. On quality control and machine learning in crowdsourcing. In *Workshops at the twenty-fifth AAAI conference on artificial intelligence*. Citeseer.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. In *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)*, pages 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Liu, Stephen Casper, Dylan Hadfield-Menell, and Jacob Andreas. 2023. Cognitive dissonance: Why do language model outputs disagree with internal representations of truthfulness? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4797, Singapore. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Mistral AI. 2023. Announcing Mistral 7B. Blogpost.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. Rissanen data analysis: Examining dataset characteristics via description length. In *International Conference on Machine Learning*, pages 8500–8513. PMLR.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. Can language models teach weaker agents? teacher explanations improve students via theory of mind. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, and Lora M Aroyo. 2021. "everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. 2021. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *Advances in Neural Information Processing Systems*, 34:6695–6706.

Charles Spearman. 1987. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293.

TII. 2023. Falcon LLM. Blogpost.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. 2023a. What algorithms can transformers learn? a study in length generalization. *arXiv preprint arXiv:2310.16028*.

Xiang Zhou, Yichen Jiang, and Mohit Bansal. 2023b. Data factors for better compositional generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14549–14566, Singapore. Association for Computational Linguistics.

A Measuring Minimum Description Length

In addition to our human hardness measures, we employ a model-based metric based on minimum-description-length (Voita and Titov, 2020; Swayamdipta et al., 2020; Perez et al., 2021). Since experiments use up to 70b parameter LMs, we measure MDL with models in the 7b parameter range, including Falcon-7b (TII, 2023), Mistral-7b (Mistral AI, 2023), Persimmon-8b (Elsen et al., 2023), and Llama-1-7b (Touvron et al., 2023a). To get one MDL per datapoint, we average the MDL scores obtained for each of the four models. For a single model, we obtain a score by training N models on training sizes in $n \in \{5, 20, 80, 340, 900\}$ (roughly log-uniform) when fitting models with linear classifier heads or QLoRA and averaging model label confidences across these n per-datapoint

scores. For ICL we *compute MDL using no training data*, i.e. $n = 0$. In this way, our Linear Probe and QLoRA MDL metrics represent MDL according to the theoretical definition which involves increasing amounts of training data, while MDL (ZS Prompt) represents the confidence that 7b models assign to labels for data with no supervision. All metrics are then used for assessing how stronger models will perform on data that weaker models find to be hard. See distributions of MDL scores on each dataset in Fig. 20. We only measure probing-based and QLoRA-based MDL for ARC and GSM8k, where we have sufficient data to set aside $n = 1000$ points (up to 900 for training and 100 for model validation).

B Additional Results

We include a number of additional results in this section.

1. For a table of model accuracies by training method in an all-to-all setup, see Table 4.
2. We show correlations between hardness measures for all data in Fig. 8.
3. We show that test accuracy declines with test data hardness for QLoRA in Fig. 10, and using ICL with additional hardness measures in Fig. 9.
4. We show easy-to-hard generalization as measured on *all* test data (not subsetting to hard test data) in Fig. 11, as well as testing on *easy* test data in Fig. 12.
5. We show easy-to-hard generalization on additional hardness measures in Fig. 18.
6. We show SGR statistics for all training methods in Fig. 13.
7. We give SGR estimates along with confidence intervals and p-values, obtained by block bootstrap, in Table 2.
8. We show easy-to-hard generalization on StrategyQA across different models in Fig. 14.
9. We give additional results for test performance by multiple training difficulty levels in Fig. 16.
10. We give test performance by training difficulty for multiple ∼70b models in Fig. 17.
11. In Fig. 19, we test the hypothesis that task format alone is taught by training data, as opposed to true generalization. We list examples used for "task format only" prompts, which are trivially simple examples matching multiple-choice or yes/no answer prompts in our data.

C Dataset Details

We provide additional details for each dataset below, including test data subsetting decisions (see final sample sizes in Table 2). All datasets are publicly available, and license information is included via the links provided. All datapoints are in English. See also Table 1 for a list of which hardness measures are available for which datasets, as well as sample sizes. See Fig. 20 for histograms of hardness measures per dataset.

- **ARC** (Clark et al., 2018):[4] U.S. grade-school science questions in multiple-choice format. We combine the ARC-Easy and ARC-Challenge splits. We release metadata including human hardness metadata accompanying the original source of the questions in our codebase at https://github.com/allenai/easy-to-hard-generalization. We set aside 1000 points for MDL computation, so experiments are conducted on $n = 3521$ test points.
- **MMLU** (Hendrycks et al., 2021):[5] Domain-specific multiple-choice questions for a large number of domains. We subset to high school and college level math, physics, biology, chemistry, and computer science questions (termed MMLU-STEM-5). Here, grade level is high school (HS) vs. college. See Figure 2 (left) for an example.
- **StrategyQA** (Geva et al., 2021):[6] General knowledge trivia questions requiring compositional reasoning over individual facts. The number of facts that must be combined forms the "Num. Reasoning Steps" measure.
- **GSM8k** (Cobbe et al., 2021):[7] U.S. grade school math word problems. The number of steps in the solution to the problem forms the "Num. Reasoning Steps" measure and

[4] https://huggingface.co/datasets/ai2_arc
[5] https://huggingface.co/datasets/tasksource/mmlu
[6] https://huggingface.co/datasets/wics/strategy-qa
[7] https://huggingface.co/datasets/gsm8k

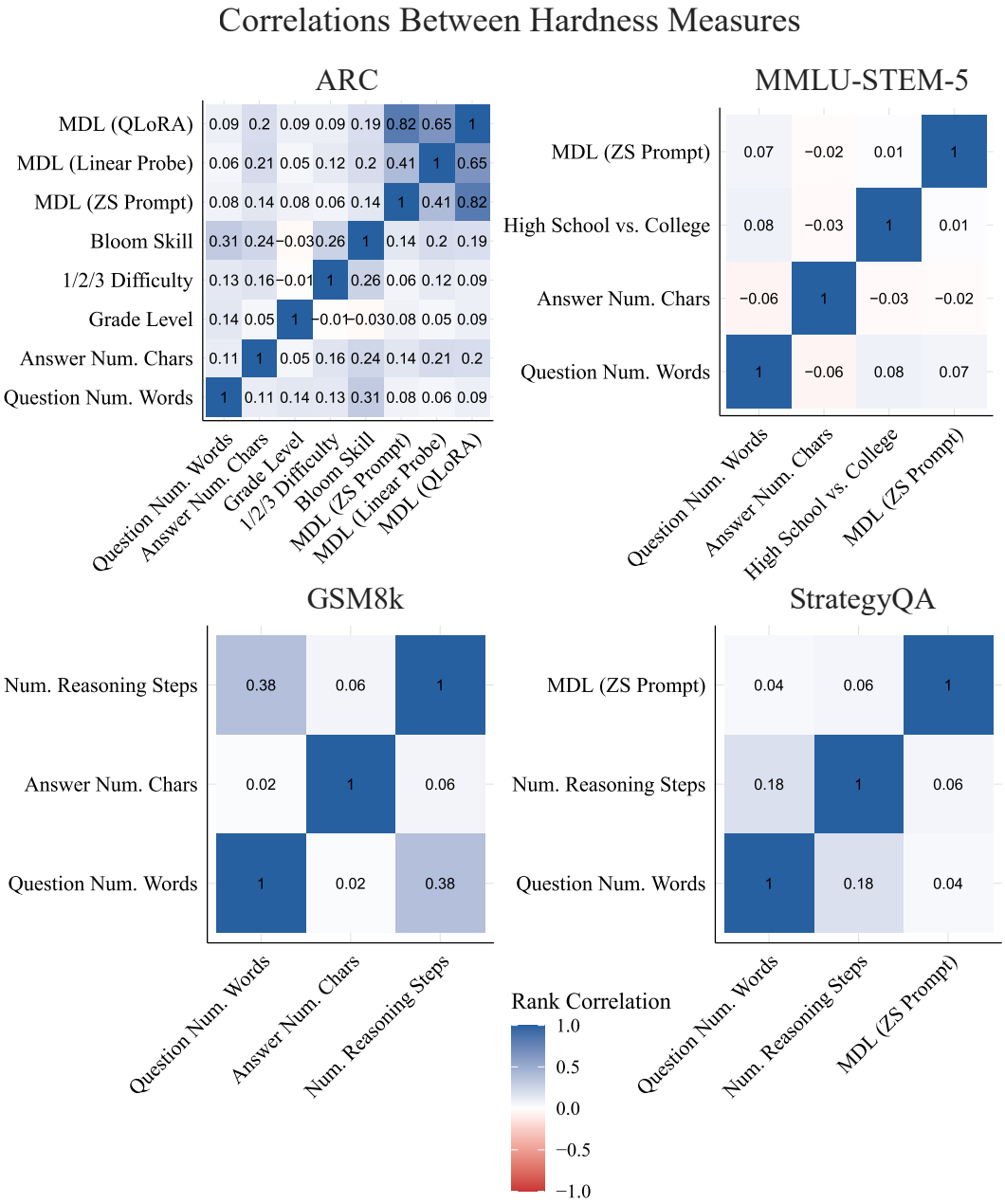

Figure 8: Correlations between hardness measures across datasets (Spearman rank order correlation). We omit MDL for GSM8k because 7b parameter models obtain extremely high loss on GSM8k problems, and MDL is valid as a metric only when using a reasonably good model of the data.

is obtained from the human-annotated reasoning chain collected for each problem. We set aside 1000 points for MDL computation, then further subset to about $n = 2000$ test points given the extreme expense of sampling CoTs with $t = 300$ tokens for 70b parameter models. See Figure 2 (right) for an example datapoint.

Distributions for hardness measures for each dataset and hardness measure (from Table. 1) are shown in Fig. 20.

D Modeling and Tuning Details

We provide additional information around model tuning for each training method here.

GPU Cost. We run experiments on NVIDIA A6000 GPUs. A typical experiment setting is running Llama-2-70b over $n = 2000$ datapoints, quantized 8bit, with ICL with $k = 8$, with batch size 1, using 5 random seeds for prompt data selection. Such an experiment requires 4 GPUs and takes about 10 hours to complete. Experiment with CoT are more expensive, taking about one hour per $n = 200$ test points on GSM8k, using a decoding

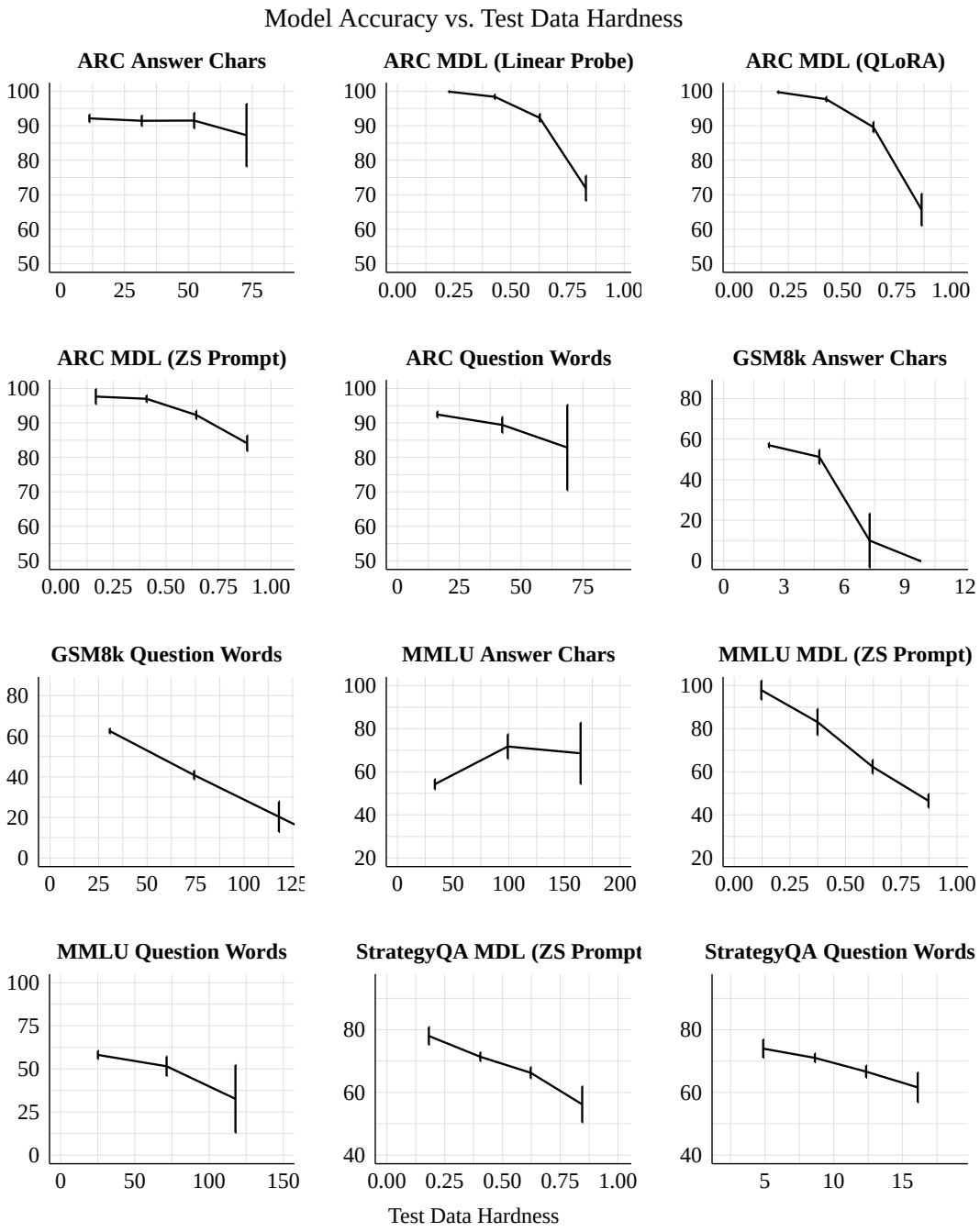

Figure 9: Test accuracy declines as test data hardness increases (shown for **additional hardness measures**), with the exception of MMLU Answer Chars. Error bars are 95% CIs showing test sample variance.

length of $t = 300$ tokens. Linear probing experiments have negligble runtime as we save the hidden states to file, avoiding the need to rerun model forward passes, while QLoRA experiments take about as long as ICL experiments (slightly faster for CoT settings).

Prompt Templates. All training methods use the same prompts, one per dataset, that were selected based on their success in past work (Saha et al., 2023). We show prompts for ARC and MMLU in Table 3. In this template, the {} placeholders are filled with the question, four answer choices, and a single answer choice (explained next). We use this prompt for multiple choice scoring of the four answer choices *for all methods*, meaning that we run four forward passes to either (1) compute answer choice probabilities for each answer choice for ICL and QLoRA, or (2) collect final answer choice token representations for each answer choice for linear probing. Thus the final answer slot, "A: {}" is populated by each answer choice once. When prompting with k in-context examples for ICL, we

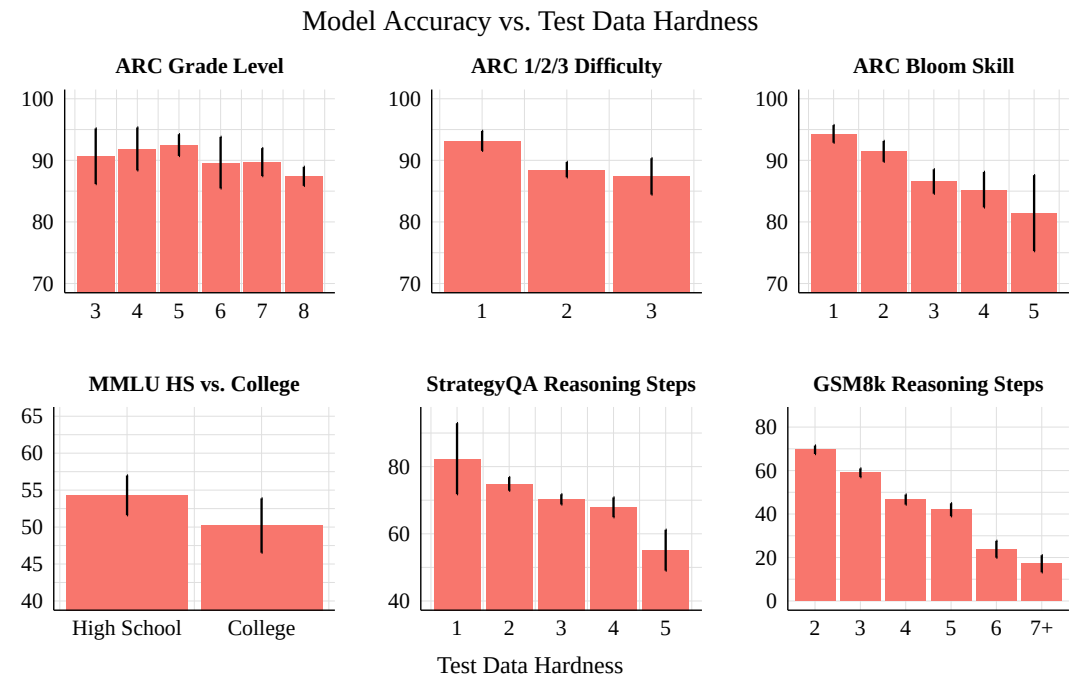

Figure 10: Test accuracy declines as test data hardness increases (shown for **QLoRA with Llama-2-70b**). Error bars are 95% CIs showing test sample variance.

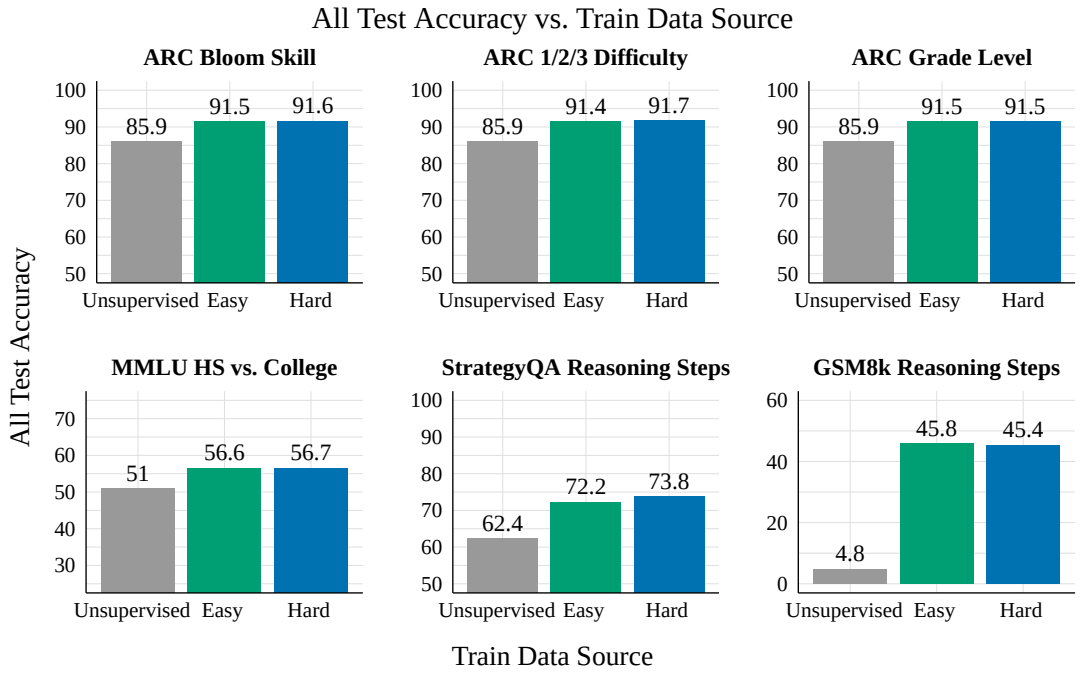

Figure 11: Easy-to-hard generalization measured on **all test data** (not subsetting to hard test data), while training on easy/hard data defined according to each hardness measure (using Llama-2-70b prompted with $k \leq 10$ examples). Results are similar to testing on hard data, except for GSM8k, where accuracy on the whole data distribution becomes comparable (training on easy data outperforms hard data on easy/medium test data).

separate examples with a line break (one line between each pair of examples).

For StrategyQA and GSM8k, we use a different prompt format for CoT, shown in Table 3. In this template, the {} placeholders are filled with the question, the human reasoning chain, and the answer choice (*only for in-context examples*). This prompt is used to generate new reasoning chains and answers at test time, so there is no text included after "A:" for the test input. When prompting with

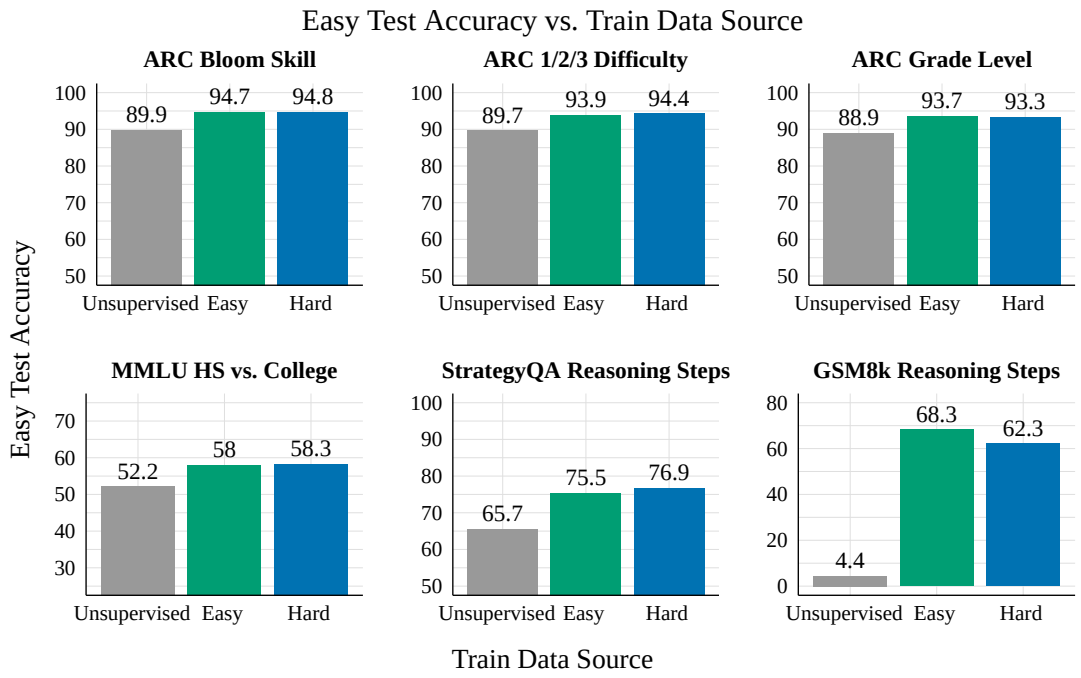

Figure 12: Easy-to-hard generalization measured on **easy test data**, while training on easy/hard data defined according to each hardness measure (using Llama-2-70b prompted with $k \leq 10$ examples). This plot shows hard-to-easy generalization for each dataset, compared to easy-to-easy genearlization. On some datasets, hard data makes for better training data, while for others, easy training data is better for easy test performance.

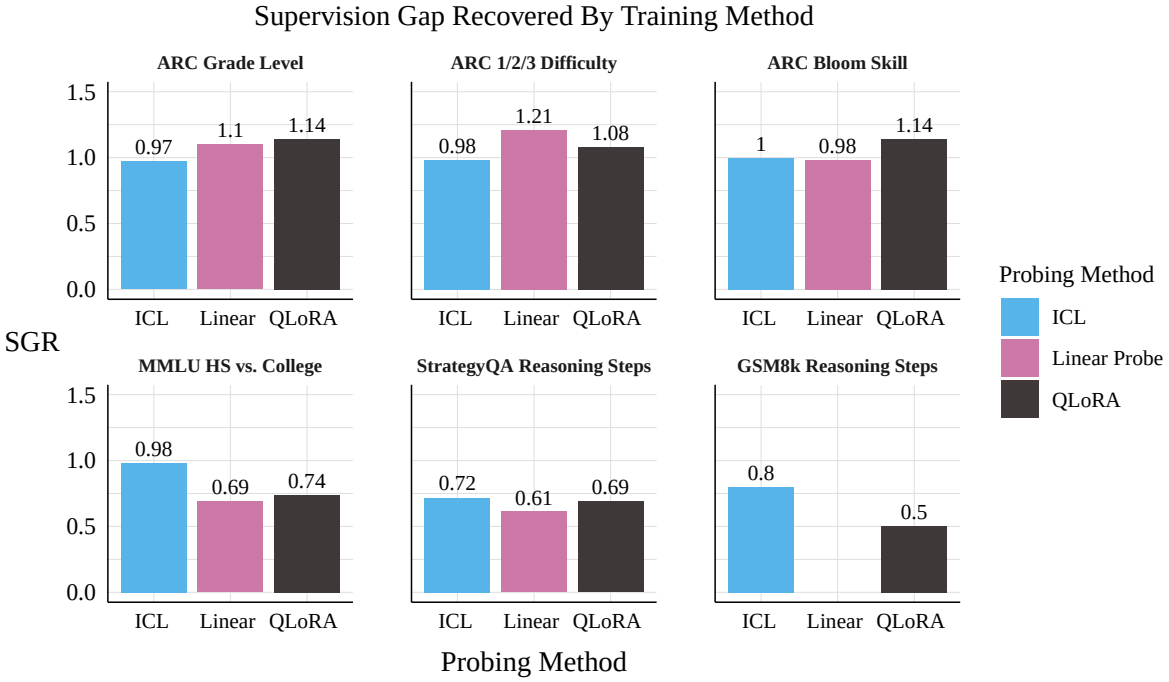

Figure 13: The Supervision Gap Recovered (SGR) shown by training method. Easy-to-hard generalization varies somewhat by training method used, but SGR remains surprisingly high across datasets for the two most effective training methods, ICL and QLoRA.

k in-context examples for ICL, we separate examples with a line break (one line between each pair of examples). The exception to this formatting is for GSM8k's Unsupervised Baseline, which uses a "Let's think step-by-step" prompt (we also considered this for StrategyQA, but zero-shot answer choice scoring worked better). The step-by-step prompt is shown in Table 3. The test input is sup-

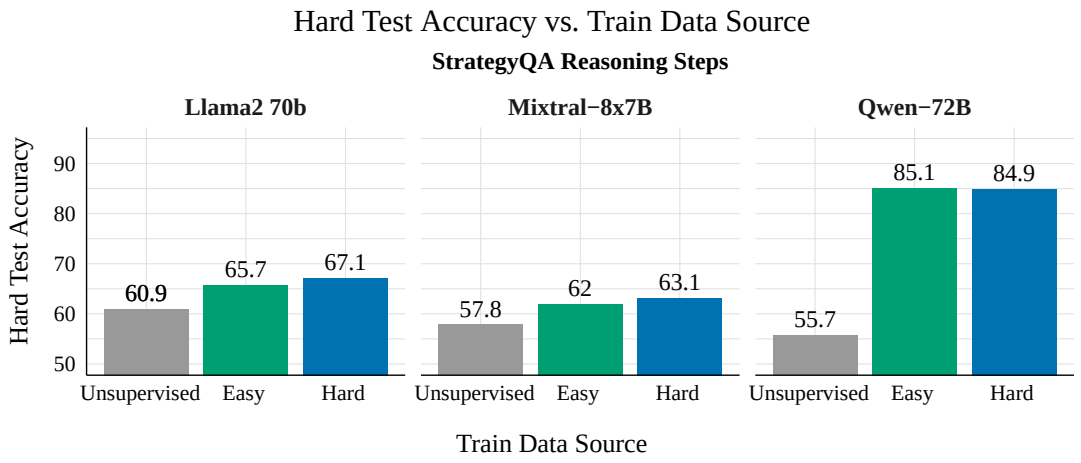

Figure 14: Easy-to-hard generalization results for different models on StrategyQA, using Llama-2-70b prompted with $k = 4$ examples using CoT. Results are similar for Llama-2-70b and Mixtral, while Qwen appears to have been trained on StrategyQA data in a CoT format.

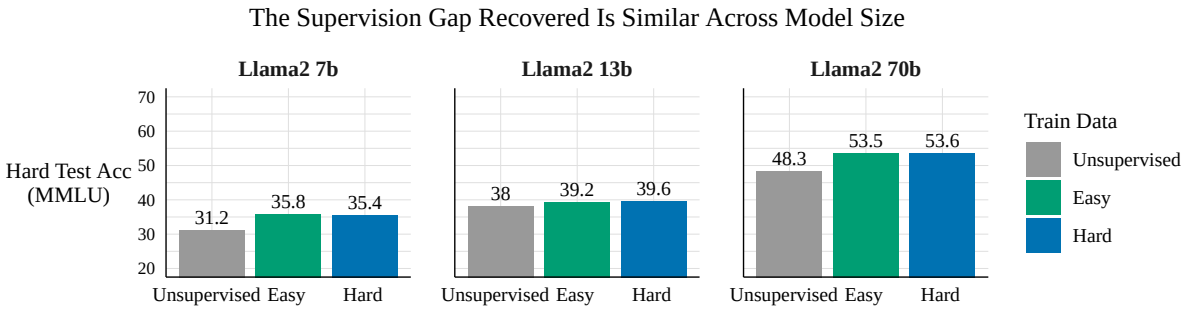

Figure 15: Models perform as well on hard MMLU data when prompted with easy MMLU data as they do when prompted with hard data, regardless of model size ($k = 10$ examples used for ICL).

Figure 16: Test performance on hard data sometimes declines more significantly as the gap between train and test hardness grows, but often the difference between training on Medium and Easy data is relatively small in nature (using Llama-2-70b prompted with $k \leq 10$ examples). MMLU not shown here since there are only two hardness levels for that dataset (high school vs. college). See Fig. 7 for more results training on college vs. high school vs. 8th grade vs. 3rd grade data.

Figure 17: Test performance on hard data declines as the gap between train and test hardness grows for reasoning datasets, **across models**, using ICL with $k = 10$.

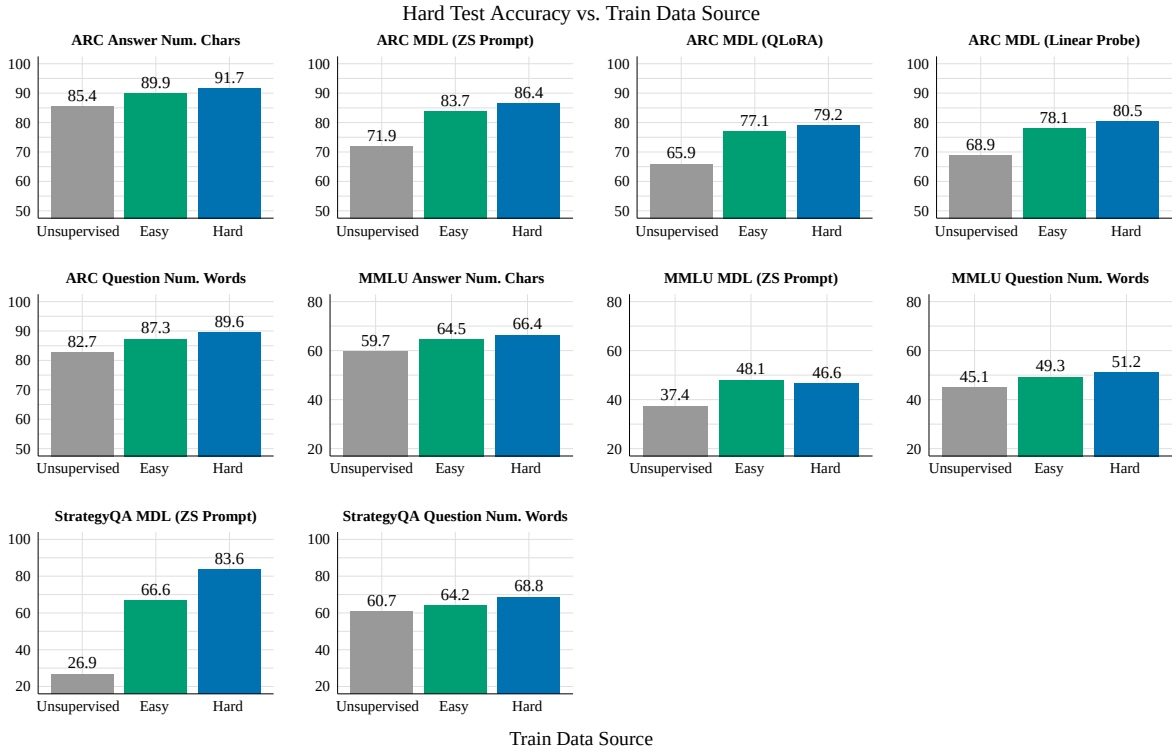

Figure 18: Easy-to-hard generalization for additional hardness measures for each dataset, using Llama-2-70b with ICL. SGR values remain high across possible hardness measures, meaning easy data provides surprisingly good supervision. We do not represent Answer Num Chars. for StrategyQA here because that would cleanly divide the data into 'no' and 'yes' categories. We do not conduct any additional experiments for GSM8k hardness measures as these experiments (involving sampling CoTs with $t = 300$ tokens) are extremely computationally expensive.

plied to the curly brace placeholder.

In-context Learning. For ICL, we select $k = 10$ for ARC and MMLU and $k = 8$ for StrategyQA and GSM8k as we see diminishing returns to accuracy from larger values of k, and using larger k values significantly slows down experiments.

Linear Probing. For Linear Probing, we fit a linear classifier to frozen LM hidden states. For a

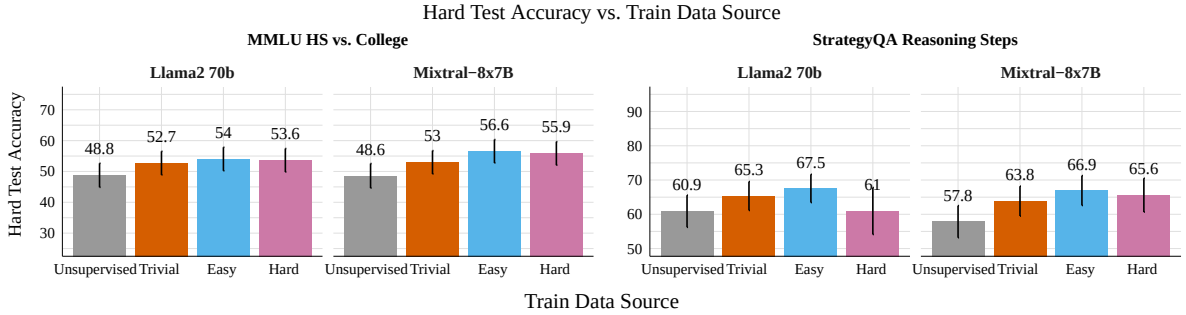

Figure 19: We consider prompts that contain trivially simple problems that match the task format for MMLU and StrategyQA (see Table 5). Results are shown for ICL without CoT, using Llama-2-70b and Mixtral-8x7b. We conclude that task format alone is not sufficient for encouraging generalization to hard data, because hard test performance varies by train data hardness, while all prompts share the task format (the best supervised prompt is better than other supervised prompts at a statistical significance threshold of $p < .05$ for three of four comparisons). For example, Llama-2-70b actually does not generalize at all from hard prompt examples for StrategyQA, suggesting that task format alone is not enough for generalization.

Dataname	Hardness Measure	SGR Estimate	Test Hardness	n
ARC	Grade Level	0.96 ± 0.10 (p < 1e-4)	Hard	1588
ARC	1/2/3 Difficulty	0.98 ± 0.36 (p = 0.0033)	Hard	1588
ARC	Bloom Skill	1.00 ± 0.20 (p < 1e-4)	Hard	1588
MMLU	HS vs. College	0.97 ± 0.59 (p = 0.0158)	Hard	603
StrategyQA	Num Reasoning Steps	0.72 ± 0.93 (p = 0.0788)	Hard	427
GSM8k	Num Reasoning Steps	0.79 ± 0.60 (p = 0.0125)	Hard	333
ARC	Grade Level	1.00 ± 0.09 (p < 1e-4)	All	3521
ARC	1/2/3 Difficulty	0.96 ± 0.08 (p < 1e-4)	All	3521
ARC	Bloom Skill	0.98 ± 0.08 (p < 1e-4)	All	3521
MMLU	HS vs. College	1.00 ± 0.27 (p = 0.0001)	All	1746
StrategyQA	Num Reasoning Steps	0.87 ± 0.32 (p < 1e-4)	All	2290
GSM8k	Num Reasoning Steps	0.98 ± 0.39 (p = 0.0003)	All	2065

Table 2: Supervision Gap Recovered (SGR) statistics for Llama-2-70b with ICL, on hard test data or all test data, defined per dataset and hardness measure. Confidence intervals are 95% CIs estimated by block bootstrap (accounting for test data variance and train data variance), and p-values represent a test for a difference from 0.

given question, we compute one representation per answer choice by concatenating the question and answer choice and feeding it to the model. To get a single representation from the LM forward pass, we concatenate the representations at the last token index (i.e., the last answer token) from the middlemost and last layer. Then, we score each question-answer choice representation z by applying the linear probe: $f(z; w) = w^T z$. The answer choice with the highest score is returned as the prediction. The probe weight w is trained using SGD to minimize cross-entropy loss on a dataset of frozen representations $Z = \{\{z_{i,j}\}_{j=1}^{j=|A|}\}_{i=1}^{N}$ for a dataset of N training datapoints and $|A|$ answer choices. We optimize the weight with the default `SGD` implementation in PyTorch (Paszke et al., 2019) for 100 epochs, without early stopping on any dev data. Based on tuning experiments with Llama-2-13b on ARC, we chose to use SGD rather than AdamW, selecting the number of epochs for convergence, and we chose to use the middlemost and last layer representations (concatenated) rather than either on its own. Note this produces very high dimensional inputs, but by using ℓ_2 regularization with $\lambda = 1$, we are able to stably fit probes to these $2d$-dimensional input representations (where $d = 8192$ for Llama-2-70b) with as few as $n = 10$ training points. The learning rate was fixed at 5e-2.

Model Finetuning with QLoRA. For QLoRA, we selected hyperparameters based on early experiments with Llama-13b on ARC. Based on this setting, we selected an adaptor rank of $r = 16$ rather than $r = 8$, with default hyperparameters otherwise, including default selected layers to optimize. We use AdamW with a a learning rate fixed at 1e-4, and the model is optimized with a batch size of 50. At least 10 gradient updates are performed, or a minimum of three epochs, whichever yields

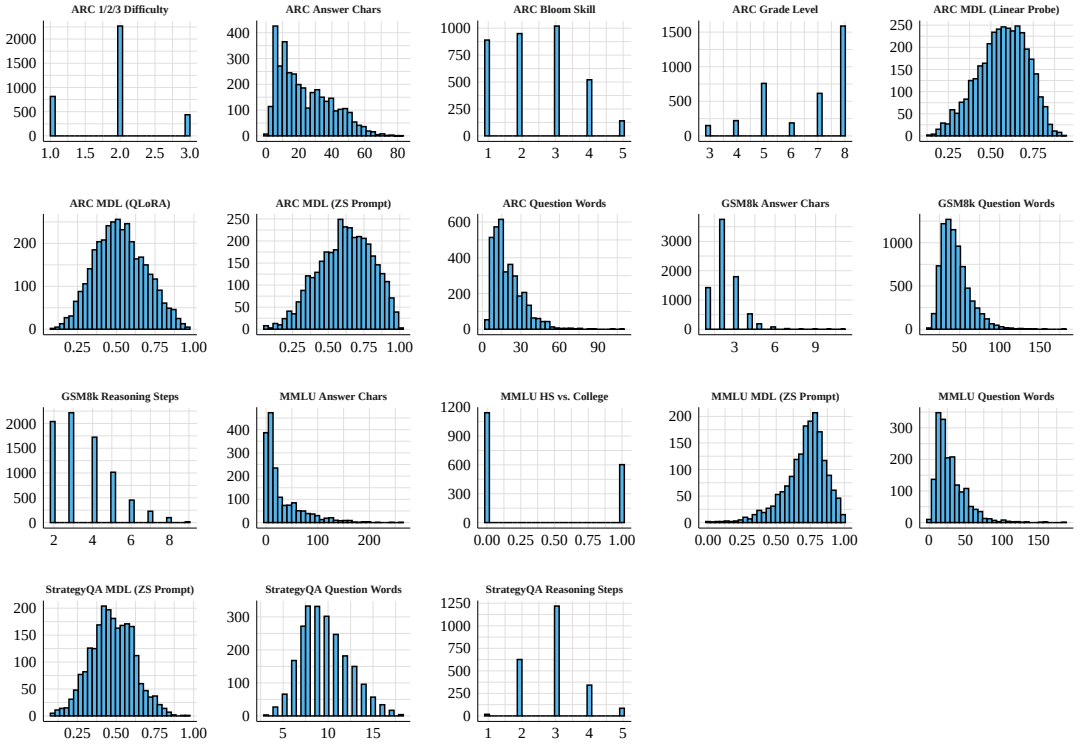

Figure 20: Distributions for hardness measures for each dataset and hardness measure.

ARC+MMLU prompt
Question: {}
A) {}
B) {}
C) {}
D) {}
Answer: {}
StrategyQA+GSM8k prompt
Q: {}
A: {} So the answer is {}
Unsupervised GSM8k prompt
Q: [question text here]
A: Let's think step by step.
1. [step one]
2. [step two]
...
N. [last step]
Therefore, the answer is [answer here].
Now you try!
Q: {}
A: Let's think step by step.
1.

Table 3: Prompt formats used in this paper. Question text, reasoning text, answer choices, and answer text are imputed in curly brackets. The notation "[step one]" is literal, and no variables are imputed in these brackets. When in-context examples are included in the prompt, we separate each example with one empty line.

more gradient updates. This means that for train $n = 160$ points and a batch size of 50, we generally perform 12 gradient update steps (3 epochs) in our experiments.

Decoding Steps. To select the number of decoding steps for each datasets ($t = 100$ for StrategyQA and $t = 300$ for GSM8k), we wanted to make sure that we were generating reasoning chains long enough for Llama-2-70b to solve hard test questions. Therefore, we intentionally selected this parameter based on hard test performance, in order to use as small a valuable as possible (based on experiment efficiency) that did not compromise test performance on hard data.

Quantization. All models are run in 8bit quantization, except for Qwen and Mixtral, which are run in 16bit format, and falcon-7b and persimmon-8b, which are run in full precision. We observe no performance loss from quantization in any experiment.

Training Size Controls. For all experiments with linear probing and QLoRA, we use $n = 160$ train points. While we would prefer to use more training data, the bottleneck we face is that fairly comparing easy-to-hard with hard-to-hard generalization requires both fixing the amount of training data and leaving enough hard data left over for testing. Since we have as few as $n = 603$ hard test

Dataset	Method	CoT	n	Acc (%)
ARC	ICL	No	0	85.94
ARC	ICL	No	10	91.77
ARC	Linear Probe	No	160	89.48
ARC	QLoRA	No	160	89.47
GSM8k	ICL	Yes	0	4.80
GSM8k	ICL	Yes	8	56.24
GSM8k	QLoRA	Yes	160	52.64
MMLU-STEM-5	ICL	No	0	48.30
MMLU-STEM-5	ICL	No	10	56.83
MMLU-STEM-5	Linear Probe	No	160	53.01
MMLU-STEM-5	QLoRA	No	160	52.77
StrategyQA	ICL	No	0	62.40
StrategyQA	ICL	No	8	70.04
StrategyQA	ICL	Yes	8	72.86
StrategyQA	Linear Probe	No	160	68.79
StrategyQA	QLoRA	No	160	66.36
StrategyQA	QLoRA	Yes	160	75.09

Table 4: Model accuracy table when finetuned on randomly selected data from the whole data distribution and tested on the whole data distribution (zero-shot ICL rows included), using Llama-2-70b. Averaged over 5 seeds. Compare to Fig. 11.

points for MMLU, we have to limit training data to $n = 160$ points to leave enough test data for reasonably small confidence intervals. Linear probing and QLoRA demonstrate good sample-efficiency when applied to Llama-2-70b, so we are able to obtain comparable (and sometimes better) performance than ICL across datasets using these methods.

E Statistical Testing

Here, we describe in greater detail how our statistical testing works. We aim to make the most of the data we have, e.g. $n = 603$ hard datapoints for MMLU-STEM-5. Ultimately, we want to use a block bootstrap that resamples (1) test datapoints and (2) models (equivalent to picking which random seed was chosen for training), in order to account for variance due to limited test data, selection of training data, and any random training dynamics. Therefore, we run five random seeds for each experiment, randomly selecting training data and using all remaining data as test data. Each experiment produces a matrix (block) of results, with up to five model predictions per datapoint. Running this matrix through a block bootstrap that resamples rows and columns produces a confidence interval for the statistic of interest (Efron and Tibshirani, 1994). When computing the mean of a resampled matrix, we ignore missing values (which represent that a datapoint was used for training and could not be tested on). We take 100,000 resamples.

We can use a bootstrap to obtain estimates for our SGR statistic too. We aim to estimate the expected value of the random variable

$$\frac{\text{Easy} - \text{Unsupervised}}{\text{Hard} - \text{Unsupervised}}$$

using the samples Unsupervised, Easy, and Hard representing their respective experiment outputs ($n \times 5$ matrices of model predictions). We perform a paired test with respect to test datapoints (resampling the same rows across each matrix), while not assuming random seeds are paired (resampling different columns for each matrix). Note that for the Unsupervised matrix, each column is identical because there is no prompt data. The results of this analysis are given in Table 2. When showing results for hard test data, we subset to just points that are hard according to their respective hardness measure.

The one exception to this kind of bootstrap sampling is for results in Fig. 3 and 9, where we only show CIs derived from test sample variance. In these plots, we average over five random training seeds.

MMLU prompt
Question: Which one is usually yellow? A) Cheese B) Apple C) Carrot D) Bread Answer: Cheese Question: What do we use to wash dishes? A) Broom B) Shovel C) Soap D) Hammer Answer: Soap Question: What color is the sky when it's sunny? A) Grey B) Blue C) Black D) Orange Answer: Blue ...
StrategyQA prompt
Q: Does cheese come from a plant? A: no Q: Can you use a broom to sweep the floor? A: yes Q: Is the sky green when it's sunny? A: no ...

Table 5: Example questions used for testing model performance based on task-format-only prompts. We use ChatGPT to generate 50 such questions for MMLU and 40 for StrategyQA. These are used for experiments using $k = 10$ examples for MMLU and $k = 8$ for StrategyQA, averaging over five different prompts (results shown in Fig. 19).