# BITS THAT COUNT: QUANTIFYING AND PREDICTING THE CAPABILITIES OF LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

1. We investigate the information requirements necessary to elicit language model capabilities, both those which are latent (i.e., not yet manifest or readily demonstrated zero-shot) and those which are already exhibited to some degree, versus to teach new capabilities. We do this in two ways: by examining the minimal number of parameters which must be tuned and the minimal number of examples which must be trained on to surface those capabilities to various levels of performance/proficiency.
2. We find that if capabilities are latent, a very small number of parameters or training examples suffices to recover large fractions of a model's maximum performance relative to its initial baseline. We observe several examples where fine-tuning between ~10-100 *randomly selected* parameters or with fewer than 5 *randomly sampled* examples recovers over 50% of the model's full performance gap and improves performance by over 20 percentage points.
3. We decompose bounds on the minimal information needed to recover/achieve various levels of performance in terms of both the amount of information that initially must be supplied for fine-tuning and the fraction of this information that is subsequently absorbed by the model during training, and we examine differences in these requirements when relevant pretrained knowledge already exists versus is absent in the model.
4. Finally, we introduce *excess description length*, a finite-data analog to surplus description length, and *information utilization*, a measure of how much information a model gains about the true task distribution from its specific training data, and we use these quantities to make quantitative predictions of a model's maximum capability ceiling on a task.

## 1 INTRODUCTION

Large language models (LLMs) acquire diverse capabilities from pretraining on vast amounts of data, many of which are not readily expressed zero-shot. Elicitation aims to surface these latent capabilities, usually with interventions such as prompt engineering or fine-tuning, whereas teaching aims to endow a model with capabilities it lacks. Distinguishing these regimes matters for evaluation, safety, and projecting compute efficiency: elicitation should not fundamentally alter a model's capabilities, and understanding when post-training is effectively teaching informs data and compute budgets as well as capability prediction and forecasting.

Many successes in significantly improving the capabilities of current models have been achieved with minimal or targeted post-training interventions, as opposed to substantial architecture changes. Despite this, a quantitative, knob-invariant method to predict how much information is required to reach a target performance remains lacking, as does a measure of how much of that information becomes or translates to generalizable skill in the model. It remains an open question whether parts of the post-training process, such as high-compute reinforcement learning, teach (or are capable of teaching) models new capabilities, or if they solely work to amplify capabilities that already exist in models.

Our work makes progress towards clarifying these uncertainties by investigating elicitation and teaching through the lens of information-theoretic constraints: what is the minimum amount of information that must be used—both by the developer *and* by the model—to achieve some target

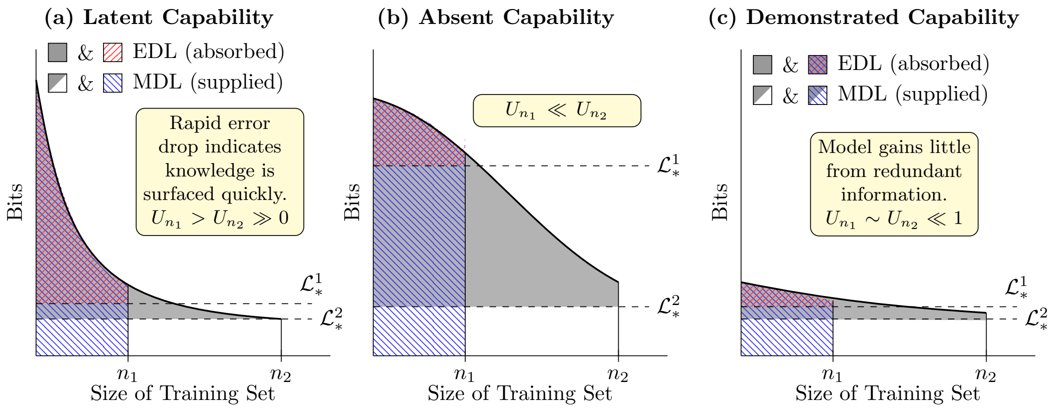

Figure 1: Caption

performance? We measure information *supplied* to the model as well as information *absorbed* by the model during fine-tuning. Minimum description length (MDL) provides a useful metric for measuring the number of bits necessary for a model to accurately encode the labels of a dataset, but it is dataset-specific and scales with the number of data examples, without revealing how many bits of those examples the model either absorbs during training or minimally requires to achieve the target performance. We use prequential MDL to upper-bound the information in the data labels and introduce *excess description length* (EDL)—the area under the first-epoch training curve above the model's test loss $L_*$—as the information in the training data gained by the model during fine-tuning which confers generalized predictive power on the test distribution. Their ratio, information utilization ($U = \mathrm{EDL}/\mathrm{MDL}$), captures how efficiently a model converts label bits into generalizable capability.

We evaluate three pretrained (base) language models within the Llama family (Llama-3.2-1B, -3B and Llama-3.1-8B) as well as a "language-only" pretrained model with the same architecture as Llama-3.2-1B (TinyStories-1B) across parameter-controlled (variable trainable parameter count) and data-controlled (variable training example count) regimes on a variety of common LLM datasets and benchmarks, including BoolQ, ARC-Easy/ARC-Challenge, and SimpleMath (plus additional generative and multiple-choice tasks). We study multiple settings in which the models tested express and contain varying amounts of task-relevant pretrained knowledge (including no task-relevant capability, *latent* capability that is not evident zero-shot, and readily-demonstrated task proficiency).

We find that if capabilities are present but latent, a very small amount of information (in the form of trainable parameters or examples) suffices to recover large fractions of a model's performance gap between its zero-shot baseline and full-weight fine-tuning. In multiple models of varying sizes, we observe several instances where fine-tuning between ~10-100 *random* parameters (uniformly sampled across the entire model) or training on fewer than 5 *randomly sampled* examples recovers over 50% of the model's full performance gap and improves performance by over 20 percentage points.

We additionally use this approach to establish scaling trends for model elicitation and teaching. On tasks for which models already demonstrate significant proficiency in zero-shot settings, performance scales logarithmically with the minimum description length of the labels, independent of the size of the dataset. On tasks which require models to learn entirely new capabilities, the scaling of performance with MDL differs significantly in the low-data and large-data regimes, as marginal returns to compression begin to dominate scaling behavior in the many-example limit.

We find that within a task, performance vs. $U$ collapses across parameter- and data-controlled settings, such that fine-tuning few (many) parameters or using few (many) unique examples are functionally equivalent in terms of the amount of generalizable information a model gains from the training data, regardless of the model's initial capability level or demonstrated task proficiency. For matched accuracy, models with latent capabilities achieve higher $U$ than models which must learn entirely new skills. We also show that, together with sample complexity (the number of unique examples which the model must be trained on to achieve a particular performance), utilization $U$ can be used to estimate capability ceilings.

The main contributions of this work include:

1. We demonstrate that models which have *latent* capabilities that exist, but are not exhibited zero-shot, require a negligible/trivial number of bits (in terms of parameters or examples used for training) to recover large fractions of their maximal performance.
2. We decompose bounds on the minimal information needed to achieve various levels of performance on tasks when models contain and express different amounts of the relevant pretrained knowledge necessary to successfully perform those tasks.
3. We introduce (i) *excess description length* (EDL), a finite-data analog to surplus description length [CITE SDL PAPER] that quantifies the amount of information in the training data that the model gains about the true distribution it is trained to predict, and (ii) *information utilization* ($U$), the fraction of the model's minimum description length of the data that comprises the generalizable information it ultimately absorbs about the overall task distribution during training.
4. We find a relationship between the sample complexity and the fraction of information a model ultimately absorbs from its initial description of the training data which can be used together to predict its maximal performance on the task.

## 2 Methodology

To measure the capacity of post-training procedures to elicit versus teach models relevant capabilities, we train and evaluate models in various settings in which their performance improvements can be attributed to either the application of relevant preexisting abilities or learning of new capabilities.

We elicit pretrained-only base models by fine-tuning on several popular benchmarks and datasets, which include multiple choice and natural language generation tasks (BoolQ, ARC-Easy/ARC-Challenge, GSM8K-CoT-Choice, TinyStories-v2, SimpleMath, Alpaca Instruction Tuning). We consider a model as capable of being elicited on a particular task if we observe a significant difference between its multi-shot and zero-shot performance on that task. Additionally, for multiple choice tasks, we employ logit bias correction for distinguishing elicitation from teaching, in which we subtract the model's prior(s) on the dataset's answer choice distribution to determine its "unbiased" zero-shot performance on the task, using this as one baseline for the minimum performance that can be elicited from the model. Details of how these baselines are implemented and computed can be found in Appendix A.

We enforce information constraints on the models and training process through two methods: restricting the number of trainable parameters and restricting the size of the dataset used for training. We refer to these settings as "parameter-controlled" and "dataset-controlled," respectively. Both methods effectively constrain the amount of information that can be added to the model during training; the former places strict architectural constraints on the amount of (new) information that can be stored in the model as well as the amount by which its original representations can maximally change, and the latter provides a bound defined by the information content of the dataset.

In the parameter-controlled setting, we use low-rank adaptation as a parameter efficient fine-tuning (PEFT) technique to restrict the number of trainable parameters while still aiming to retain as much performance per parameter as possible. We use LoRA (Hu et al., 2021) as for all parameter-controlled experiments, as we find it to yield the best performance per parameter (additional details about parameter-controlled training can be found in subsection B.1). For the data-controlled setting, we truncate the training dataset at varying fractions of the total dataset size and use the same hyperparameters for each (model architecture, dataset) configuration, ensuring that the same examples are seen in the same order by each model during the first epoch so that its training dynamics up to each successive example (batch) are the same, irrespective of dataset size. All training configurations and hyperparameters can be found in Appendix B.

To distinguish between teaching and elicitation, we perform comparative experiments on pretrained and randomly initialized models which share the same architecture, applying the same training procedures (supervised fine-tuning) on identical datasets. Because the randomly initialized model variants contain no preexisting learned representations, any and all improvements on the tasks can be directly and unambiguously attributed to teaching entirely new capabilities from scratch.

To measure teaching and elicitation in models with a preexisting knowledge base, we first pretrain a randomly initialized Llama-3.2-1B model on a corpus of simple English-language short stories which use only a small vocabulary (TinyStories-v2) to teach isolated language skills without introducing additional capabilities, such as arithmetic or science proficiency. We then post-train these narrow, "language-only" models on the same tasks (mentioned previously) with the same fine-tuning procedures as the corresponding pretrained base models, comparing performance with scaling of MDL and trainable parameters.

## 3 Information-theoretic background and definitions

**Notation.** Let $\mathcal{D} = \{(x_i, y_i)\}$ be a dataset where $x_i$ are the inputs to the model (i.e., examples from the dataset) and $y_i$ are the corresponding data labels for some language modeling task. Let us imagine that Alice (who has all $(x_i, y_i)$ pairs in $\mathcal{D}$) wants to communicate a model $\mathcal{M}'$ (trained to a final performance level $\mathcal{L}_*$ on all data pairs $(x_i, y_i) \in \mathcal{D}$) to Bob, who only has the inputs $x_i$ in $\mathcal{D}$.

**Setting.** Instead of sending Bob the model weights themselves, which may be very difficult for large models, Alice can alternatively send the labels of the dataset she used to train her model $\mathcal{M}'$, such that Bob may train an identical model $\mathcal{M}'$ with the same performance $\mathcal{L}_*$ using the same base model $\mathcal{M}$ that she used. Alice's task is then to optimally encode the full set of labels $y_{i=1,\ldots,n}$ such that she uses the minimum number of bits necessary to transmit them to Bob. This codelength, the minimum description length, corresponds to the minimum amount of information needed for Bob to train a copy of the same (initial) base model $\mathcal{M}$ to an equivalent performance level.

To make Bob's fine-tuning process more efficient, Alice and Bob both agree to use identical copies of $\mathcal{M}$ (the base model Alice used for training $\mathcal{M}'$) as the algorithm used to compress, encode, and decode the data labels. They agree on a choice of learning algorithm $\mathcal{A}$ to use, including any variables (such as hyperparameters, seeds, and optimizer) that affect the training dynamics, such that they are able to train identical models given the same training data.

**Prequential MDL and online coding.** Using the same model as Bob, Alice encodes and sends the labels one (batch) at a time, with the cross-entropy between the current model's prediction and the true label for each example (batch) determining the minimum number of bits Alice must use to transmit it. After encoding each label (batch of labels) using the model, Alice sends it to Bob, who then uses his identical copy of the current model to decode the correct label (batch). Once Bob has decoded the label(s), he and Alice then each train their respective copy of the model on the label(s) Bob received according to $\mathcal{A}$, such that their models remain identical at each step.

After training on each new label (batch), their models successively improve at predicting subsequent labels, such that each new label Alice encodes has a smaller cross-entropy (log-loss) with the prediction from the current, slightly better model she and Bob each have an exact copy of. Accordingly, the number of bits required for Alice to transmit the next label decreases as she and Bob simultaneously train a model that better predicts the data. After all labels have been sent, the total (minimal) number of bits needed to transmit them is the minimum description length, given by the sum of the cross-entropy losses for all of the individual labels in the dataset, or equivalently, the area under the training curve for the first epoch.

Once Bob receives the full set of labels, he can continue to train his copy of the model for additional epochs on those same labels to further improve its performance (following the directions of Alice to train an identical copy of $\mathcal{M}'$). Because Alice no longer needs to transmit anything additional for Bob to continue training his model (we assume the set of instructions for how to train $\mathcal{M}'$ is known in advance and included in $\mathcal{A}$), the total information required for her to communicate her fine-tuned model is merely the number of bits needed to transmit the necessary labels for Bob to be able to train an identical copy of it.

**Excess description length and bits absorbed during training.** Once Bob trains his model on all data to convergence (as judged by performance on some validation metric), its cross-entropy loss when evaluated on the test set[1] reflects the average number of bits this best, final model $\mathcal{M}'$ requires

[1] We assume that the test set is sampled I.I.D. from and accurately reflects performance on the true data distribution the model is trained to predict.

to encode labels for examples drawn from the same distribution. This is the remaining information in the data that that the model cannot compress further by training for additional epochs on the current train set (i.e., the minimum bits still required to communicate the correct labels even knowing the correct model parameters) reflecting the residual codelength of the data once the model parameters are known.

Because the minimum description length represents the smallest amount of information necessary to represent both the model *and* the data, the difference between the cross-entropy of the converged, best model $\mathcal{M}'$ trained on all data and the online codelength (the cumulative train set loss during the model's first pass over the data/area under the first-epoch train set loss curve) corresponds to the cost of the model. This "excess description length" beyond what is necessary to reproduce the correct labels using the converged model is the information in the data that the model *does* fully compress during training, reducing its loss from its zero-shot value $\mathcal{L}_0$ to $\mathcal{L}_*$ as a result of fine-tuning. Alternatively stated, the excess description length is the total *generalizable* information in the data that is absorbed by the model during training—the information in the data which contributes to better performance on the test set and which its parameters alone are sufficient to reproduce/represent—given by the difference between the prequential MDL and the remaining cross-entropy of the best model trained on that data.

**Information utilization.** Minimum description length quantifies how much information is supplied to a model, as measured by the model itself, by employing the model's intrinsic ability to generate its own most compact, optimal description of that information. MDL grows linearly with the number of examples in the dataset, with asymptotic behavior determined by the irreducible error in the dataset which can't be compressed; as more information is supplied to the model, the model's description length of the data correspondingly increases to account for the additional information, with a lower bound on the minimum additional information per example set by the irreducible error. In practice, this is approximated by the test loss of the best trained model—a loss "floor" that the model converges to in the asymptotic data limit, which describes the bits of the fine-tuning distribution that the model cannot absorb. It is the best achievable loss on the true distribution given the training data (for the specific hypothesis class).

However, the amount of information that a model can store in its parameters is bounded, with a strict upper limit on the number of bits that can be stored in any single parameter enforced by its numerical precision. During fine-tuning [TO DO: FINISH SECTION]

# 4 Results

In this section, we present the empirical results of our information-constrained fine-tuning experiments. We first demonstrate that latent capabilities can be surfaced with a markedly small amount of information. We then contrast the scaling behaviors of elicitation and teaching, and finally, we introduce information utilization ($U$) as a metric that reveals a fundamental "capability gap" between these two training regimes.

## 4.1 Elicitation surfaces latent capabilities with minimal information

A central finding of our work is that when a capability is *latent*—present in the model but not exhibited zero-shot—a surprisingly small amount of targeted information is sufficient to elicit strong performance. We observe this in both parameter-controlled and data-controlled settings.

In our parameter-controlled experiments, we find that fine-tuning a tiny, randomly selected subset of the model's parameters can yield a substantial fraction of the total possible performance gain. Across multiple tasks, we observe that fine-tuning between ~10-100 parameters recovers over 50% of the performance gap between the model's zero-shot baseline and full fine-tuning. Similarly, in data-controlled settings, training on as few as three unique examples (comprising fewer than 5 bits of compressed labels) can be enough to achieve a similar performance improvement.

Given the small amount of information contained in such few parameters or data, which should be insufficient to teach the technical knowledge necessary for proficiency in subjects such as high-school-level chemistry, this suggests that the fine-tuning process in these instances acts as a mechanism to surface or activate a preexisting, developed capability, rather than to teach an entirely new, complex skill. We expect that the information provided by the few parameters or examples instead primarily serves to align the model's existing representations with the specific task format.

### 4.2 Scaling of elicitation and teaching with information supplied in fine-tuning

While diversely pretrained language models exhibit general logarithmic scaling of performance with MDL for tasks which rely on preexisting knowledge, regardless of task (format or domain), randomly-initialized models exhibit distinct regimes of scaling behavior when learning new capabilities as a function of MDL (and correspondingly, the size of the training dataset). When learning new capabilities, the low-data regime is characterized by a relatively larger slope of performance improvement with the MDL of the labels (as compressed by the model), followed by a transition to behavior that asymptotically approaches the performance of a model with identical architecture which has already acquired the necessary capability.

### 4.3 Contrasting elicitation and teaching using information utilization

Our results indicate that elicitation and teaching operate under fundamentally different information dynamics. This is best illustrated by comparing a pretrained model to a randomly initialized model of the same architecture on the same task [FIGURE COMPARING LLAMA 1B AND RANDOM INIT ON TINYSTORIES].

The pretrained Llama-3.2-1B model, which is already initially proficient at the task, occupies a small region of the plot characterized by low test loss and low information utilization, $U$. This is a regime of diminishing returns; the model has already learned the skill, so additional data provides little new generalizable information.

In contrast, the randomly initialized variant of the same Llama-3.2-1B model architecture traces a wide arc across the plot, visualizing the entire process of learning from scratch. It begins with high test loss and low $U$. As increasing dataset sizes improve the effectiveness of fine-tuning to teach the relevant language capabilities necessary for the task, the model's utilization of the information in the training data $U$ significantly increases, reaching a peak before entering the same diminishing returns phase as the pretrained model. This broad, inverted, nearly symmetric "U" or "V" shape appears to be a characteristic signature of teaching a new capability.

This pattern is not unique to generative tasks. We observe similar dynamics across our benchmark suite, including on commonsense reasoning and multiple-choice tasks. For a model to learn a new capability, it requires two to three orders of magnitude more information (in the form of trainable parameters or MDL) than to elicit a preexisting one to achieve the same level of performance.

### 4.4 The capability gap revealed by information utilization versus test loss

The relationship between information utilization and test loss appears to reveal a fundamental property of the model-task interaction, which we term the *capability gap*. This is distinct from the *performance gap* (the difference between zero-shot and maximal accuracy).

As shown in [FIGURE OF U VS. TEST LOSS], the shape of the utilization versus test loss curve reflects how much a model must learn from scratch. A wide curve, like that of the randomly initialized model, signifies a large capability gap. The model must absorb a large amount of information from the data to build the necessary representations for the task. A narrow curve or a cluster of points in the low-loss, low-$U$ region signifies a small capability gap, as the model already possesses the relevant core knowledge.

For example, on SimpleMath, Llama-3.2-1B has a large *performance* gap of over 60 percentage points but a small *capability* gap, as revealed by its utilization curve. Conversely, on BoolQ, the "language-only" TinyStories-1B model has a smaller performance gap of around 15 points, but a comparatively large capability gap, as it must learn the concepts of question-answering and reading comprehension from scratch.

For models which already readily demonstrate a capability, we see only one side of this curve (diminishing returns to additional information/small changes in performance with large amounts of decreasing utilization). Models in which the capability is latent or only minimally demonstrated often reach the maximum utilization with only small changes in the test loss (and often in the performance metric, as well). This largely reflects the sample efficiency of elicitation: there is an associated sample complexity for which a model becomes able to make most efficient use of the information in the training data. Below this, the model lacks enough information to optimally access or make use of the relevant capability, or there is not enough information to reliably surface the capability. Once a sufficient number of examples is reached (this number is small for easily elicited latent capabilities), the information in the training data can be used to reliably and efficiently access the capability.

The trajectory of these curves in the regime of diminishing returns indicates a potential method for estimating a model's performance ceiling. We find that by fitting a curve to this portion of the data, the extrapolated minimum test loss (x-intercept) is remarkably stable and robust for each model and data configuration, even when using suboptimal hyperparameters [FIGURE FITTING EXTRAPOLATED CURVES). This observation suggests that this value may represent an intrinsic limit of the model's hypothesis class on a given dataset, a direction we believe is promising for future studies.

## 5 Discussion

### 5.1 Practical and Safety Applications

**Estimating capability ceilings.** Our finding that a stable, robust lower bound on a model's test loss on a given data distribution can be extrapolated from its information utilization beyond the point of diminishing marginal returns to data suggests that it may be possible to estimate model capability ceilings. Given some data distribution, performing several preliminary fine-tuning runs using varying amounts of data may be able to provide sufficient signal to estimate maximal performance and scaling of returns to training on additional data from that distribution. This might help inform optimal tradeoffs to additional data, predictions of whether a model or hypothesis class is capable of achieving a target performance with a particular amount or composition of training data, determinations of which data are most effective at producing desired capabilities, estimations of the gap between a model's current and maximal performance on some distribution, or improved forecasting and quantitative approximations of the compute "overhang" (the amount by which capabilities could be increased with sudden improvements in efficient usage of computational resources).

**Informing more accurate evaluations.** Our observation that, for complex capabilities, eliciting or refining extant capabilities scales differently than teaching absent ones has implications for model capability and safety evaluation. To avoid unintentionally introducing new capabilities, ensuring that the model's capability gap remains small or that its utilization $U$ benefits from high marginal returns to (small amounts of) information under some scaffold, intervention, or elicitation technique may help reduce the likelihood of inadvertently inducing or benchmarking significantly different capabilities in a model than would be present during deployment.

### 5.2 Limitations

**Distinguishing low sample complexity teaching from elicitation.** Our characterization of elicitation versus teaching is (and is primarily meant to be) descriptive, rather than prescriptive. We expect that teaching skills with low sample complexity (such as how to format answers to multiple-choice questions) looks very similar to (and is effectively indistinguishable from) eliciting latent capabilities. We believe this is also a strength of the framework we propose: if eliciting complex, highly advanced skills becomes indistinguishable from teaching them entirely because both can be achieved with high sample efficiency and small amounts of information, there ceases to be a functionally useful distinction between elicitation and teaching.

**Model and task diversity.** We primarily study a single family of models (Llama 3 models and architecture) in limited sizes (1B, 3B, and 8B parameters) on a limited set of tasks, including several small datasets in multiple choice format to maximize task breadth and diversity given computational constraints. Though our findings are consistent across datasets, model sizes, architectures, and pretraining procedures, it is possible that different capabilities, larger models, other model families exhibit different scaling behavior. While we compare two models (Llama-3.2-1B and TinyStories-1B) on tasks for which we can unambiguously distinguish elicitation and teaching, we are unable to do the same for the 3B and 8B parameter models because of the limited size of the TinyStories-v2 corpus. We hope that future work expands upon the preliminary framework we propose here to study other capabilities of interest in frontier models.

**Capability estimation limitations.** While considering information utilization, in addition to the total information supplied during fine-tuning, suggests a possible method for estimating model capability ceilings, we emphasize that these estimations are dataset-specific within the chosen hypothesis class. Our framework does not predict how maximal capabilities might change if the training data distribution, learning algorithm, or base model were modified.

### Related Work

[TO DO]

## 6 Distinguishing between elicitation and teaching (notes that still need to be integrated into other sections)

1. While both diversely/generally pretrained and "language-only" pretrained models exhibit logarithmic scaling of performance with increasing train dataset size, "language-only" pretrained models behave differently in small-data and large-data training regimes, with initial large returns to increasing dataset size that taper off, accompanied by decreasing information utilization.
2. While diversely pretrained language models exhibit general logarithmic scaling of performance with MDL for tasks which rely on preexisting knowledge, regardless of task (format or domain), randomly-initialized models exhibit distinct regimes of scaling behavior when learning new capabilities as a function of MDL (and correspondingly, the size of the training dataset). When learning new capabilities, the low-data regime is characterized by a relatively larger slope of performance improvement with the MDL of the labels (as compressed by the model), followed by a transition to behavior that asymptotically approaches the performance of a model with identical architecture which has already acquired the necessary capability.
3. While both diversely/generally pretrained and language-only pretrained models can exhibit logarithmic scaling of performance with increasing train dataset size/MDL of the labels, regardless of whether training involves elicitation or teaching the model entirely new skills, the slope of the performance improvement with minimum description length differs between teaching and elicitation settings (i.e., when a model can access a relevant preexisting capability vs. when it must learn new task-related capabilities from scratch), particularly in the low-data regime.
4. To achieve the same/equal performance levels, learning new capabilities requires several orders of magnitude more bits than eliciting preexisting capabilities. We find that this separation is generally two to three orders of magnitude or more. We validate this across several domains and models, while controlling for factors such as model architecture.
5. FIGURE CAPTION: When fine-tuning is successful at eliciting a *latent* capability, a small amount of information is sufficient to surface the capability (with a correspondingly high information utilization). When fine-tuning primarily serves to further improve performance by marginally enhancing an existing capability that is already (partially) demonstrated, we observe logarithmic returns to performance with increasing MDL and constant or decreasing information utilization with increasing dataset size (as well as with further improvements in performance).
6. FIGURE CAPTION: When a large amount of information is used for fine-tuning (in the form of a large training dataset with many examples), it becomes difficult to distinguish between elicitation and teaching, as information consumption is low in both cases; however, for the same information budget (in terms of dataset size or parameters trained), when comparing a model which initially lacks a capability to one for which that capability already exists (whether latent or manifest), information utilization in the large-information limit is higher for the model which had to learn the capability from scratch, as the useful information for amplifying or enhancing a latent capability is amortized over the size of the dataset (number of training examples/labels) such that high information utilization and sample efficiency early on in training result in lower utilization over the full training run. (This is why it's important to consider the *minimal* amount of information

necessary for achieving various performance thresholds to distinguish between elicitation and teaching.)

7. We see a “parabolic” (more precisely, an upside-down “U”-/“V”-shaped curve) between information utilization and test loss for models which do not demonstrate a capability zero-shot (either because such a capability does not exist or because it is latent). The height of this curve depends on the amount of information required for the model to demonstrate the capability, with smaller curves being associated with easier/less information required for elicitation. This is robust to hyperparameters which change the training dynamics (and in particular, how quickly/efficiently the model trains within the first epoch): while suboptimal hyperparameters result in artificially (misleadingly?) high utilization because the model’s initial description length of the data is larger than what is minimally required, the predicted values of the model’s maximum and minimum capabilities (given by the limits of the curve/extrapolating the intercepts) are the same (within error). The width of this curve reflects the size of the model’s “capability” gap: how much preexisting knowledge/relevant capability the model has for the task. Models which demonstrate significant capability gaps lack preexisting relevant capabilities. Regardless of the size of the *performance* gap (the difference between a model’s zero-shot and maximum performance based on a particular chosen metric), which may be large if the model has a latent capability, the *capability* gap remains small if the capability already exists within the model. For example, on SimpleMath, Llama 3.2 1B has a large performance gap of over 60 percentage points (exact match accuracy pass@1: 20.8% zero-shot versus 83.1% full-weight fine-tuned) but a small capability gap (less than 2 nats per token), reflecting its preexisting knowledge base, whereas on BoolQ, TinyStories-1B has a relatively smaller performance gap of 15 percentage points (balanced accuracy: 50% zero-shot versus 65% full-weight fine-tuned), but a relatively larger capability gap of 4 nats per token, corresponding to the model being taught a relevant capability from scratch and reflecting the relatively higher information requirements necessary to accomplish this.
8. For models which already readily demonstrate a capability, we see only one side of this curve (diminishing returns to additional information/small changes in performance with large amounts of decreasing utilization). For models in which the capability is latent or only minimally demonstrated, we often see models reach the point of maximum utilization with only small changes in the test loss (and often in the performance metric, as well). This largely reflects the sample complexity required for elicitation: there is an associated sample complexity for which a model becomes able to make most efficient use of the information in the training data. Below this, the model lacks enough information to optimally access/make use of the relevant capability, or there is not enough information to reliably surface the capability. Once a sufficient number of examples is reached (this number is small for easily elicited latent capabilities), the information in the training data can be used to reliably and efficiently access the capability.

## References

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

# A Model Performance Baselines

As one method for determining whether a model's improvement in performance can be attributed to elicitation of a relevant latent capability, we consider a model as capable of being elicited on a particular task if we observe a significant difference between its multi-shot and zero-shot performance on that task.

[ADD DETAILS ABOUT MULTI-SHOT PROMPTING]

Models pretrained on large, general, diverse corpora can develop biases in their logit distributions as a result of positional, sequential, and semantic biases in data, which result in disproportionately large probabilities predicted for tokens which correspond to affirmative responses (e.g., “True”, “Yes”, “Correct”) or the earliest position in a sequential order (e.g., "A", "1", or the first entry in a list or sequence). Multi-shot prompting can be effective at shrinking or removing these biases, resulting in better performance, particularly in multiple-choice tasks where the task format reinforces these biases. Accordingly, for multiple choice tasks, we also employ logit bias correction for distinguishing elicitation from teaching, in which we directly remove these biases for each task by subtracting the model's prior(s) on the dataset's answer choice distribution to determine its unbiased zero-shot performance on the task, using this as one baseline for the minimum performance that can be elicited from the model.

[ADD DETAILS ABOUT LOGIT BIAS CORRECTION]

### A.0.1 Logit bias-corrected accuracy

As with few-parameter training, in the few-training-example limit, a small number of bits (of labels) is sufficient to remove pretrained models' logit bias, resulting in better elicited performance.

Whereas with highly sparse few-parameter training it is impossible for the model to overfit the train set (because the description length of the compressed labels exceeds the theoretical upper bound of information that can be contained in the parameters, which is determined by their numerical precision), in the few-train-example/low-data limit, the model can overfit on the train set. We observe that elicited performance in the very-low-data regime is limited by the amount of logit bias that can be corrected during training prior to the model overfitting on the train set (which can result in a breakdown of the logarithmic relationship between MDL and performance if the hyperparameters are not correctly/optimally tuned, as elicited performance in this regime is very sensitive to perturbations).

# B Training Details

## B.1 Parameter-Controlled Training

### B.1.1 LoRA and Random Sparse LoRA Training

## B.2 Other PEFT methods

We also evaluated other LoRA variants, such as DoRA and PiSSA, for their parameter efficiency, finding that LoRA was most parameter efficient for our settings and tasks.

Other parameter-efficient fine tuning techniques, such as soft token methods (including prefix tuning, P-tuning, and prompt tuning), were also evaluated for their efficiency in improving capabilities through elicitation and teaching. We find these to be less parameter efficient than LoRA. This is because there is a lower bound on the number of tunable parameters that can be used, determined by the hidden dimension of the model. For the models tested, the hidden dimension $d = \{2048, 3172, 4096\}$ for Llama-3.2-1B, Llama-3.2-3B, and Llama-3.1-8B, respectively, resulting in similar performance as when one to two fewer LoRA parameters are used.

## B.3 Data-Controlled Training

## B.4 “Language-only” Pretraining on TinyStories-v2

Because the TinyStories pretraining corpus contains no specialized knowledge and purely teaches basic, fundamental language skills, this setup provides a testbed for assessing and measuring the information required for learning entirely new capabilities.

### B.5 Hyperparameter configurations, additional training details, and computational requirements

## C Additional Information-Theoretic Details and Derivations

## D Elicitation versus teaching with low sample complexity

We primarily select tasks in which the relevant capabilities are complex (and the sample complexity necessary to teach/learn them from scratch is expected to be large in the absence preexisting knowledge base that includes similar or related skills), such that we can easily separate elicitation and teaching. We believe that it is likely that there are low sample-complexity skills that are easy to teach, and for which, it is more difficult to separate elicitation of an existing latent capability from teaching of an entirely new one. We believe that the framework we present here still can offer useful insight in these cases, as well. While it may be difficult to definitively designate a particular skill requiring few examples to teach as either latent or absent, our proposed method for predicting/estimating a model's capability ceiling from the trend of its information utilization with test loss works in either case.

We think the distinction between teaching and elicitation is meaningful at the present for current SOTA models as well as for capabilities and tasks that are currently of significant interest for model capability and safety reasons (involving complex technical skills, knowledge, and agency). Right now, getting models to demonstrate these capabilities is a substantial undertaking and involves specialized and targeted training efforts, with many active research efforts underway to improve the performance and capability of models to match and exceed that of human experts. Understanding whether post-training interventions such as RLHF and narrow-domain fine-tuning (for advanced, field-specific/specialized usage) are capable of introducing fundamentally new, complex skills which are often difficult for even humans to learn would help us better determine how compute and different training regimes or strategies influence model capability acquisition and expression, which is relevant for the present maturity/understanding of the field. We think the ideas and techniques presented here are most valuable for quantifying and predicting the amount of information required to elicit versus teach current *frontier* capabilities, as well as what the upper bound on those capabilities might be given a particular kind of training data and hypothesis class.

Contrary to becoming obsolete by the point at which elicitation and teaching may no longer distinguishable because complex capabilities can be taught with low sample efficiency, we believe that this framework can be applied to understand when this transition (approximately) occurs. At the point at which models require very few examples to extract most of the useful, generalizable information about a particular subject or skill and can broadly apply that knowledge or understanding to improve predictions/performance across the entire domain those examples are relevant to/drawn from, regardless of discipline/field/subject matter, what defines "elicitation" and "teaching" becomes merely a matter of semantics. In terms of ultimate capabilities and how easy they are to achieve (which is what we believe the notions of "elicitation" and "teaching" currently serve as proxies for), it does not matter whether a capability technically exists already within a model if it is trivial to teach it.

Elicitation and teaching currently seem to serve as useful proxies for whether it is difficult/possible to get a model to demonstrate some behavior/capability or not. This seems to be particularly important for estimating or predicting things like how easy it is for people to get deployed models to competently perform or automate specific kinds of work (often of a malicious or harmful nature) or whether models have the capacity to behave certain (often, "misaligned") ways. Right now, these are complex behaviors and actions that are difficult to get models to demonstrate, and it is an active area of research and focus of large amounts of funding to determine how to (most efficiently) achieve them. Because it is so difficult to get models to do these things, determining whether certain post-training interventions or evaluation techniques are capable of introducing these capabilities if/where they did not previously exist is a relevant question to ask.

If, for example, introducing some capability to a particular model requires more information (from a particular source/of a particular kind) than can be compressed into the size of the model's context window, then the model cannot learn the capability via in-context learning on that dataset. Furthermore, if the model has restricted or no fine-tuning access (e.g. the company that developed it does not support fine-tuning), then evaluators can rule out the possibility that the model is able to gain that capability with a high degree of confidence. If evaluators were to try to elicit this model using fine-tuning as a proxy for prompt-based elicitation methods (because fine-tuning is often easier and faster to implement and achieves similar or better performance than prompting with less effort and time), they might inadvertently introduce the capability and mischaracterize the model as having the capability, when in reality, it is extremely difficult to get the model to demonstrate it using means available to the public.

Our method can help estimate what a model's maximum capability might be when trained using some amount of information derived from a particular data distribution, which seems like the relevant detail we actually want to understand when trying to determine whether a model can exhibit some capability (i.e., how difficult does it appear to be to get a model to demonstrate a particular capability). If there becomes a point at which we can no longer easily or intuitively determine what distinguishes aspects of a latent capability that's easy to elicit versus an absent capability that is easy to teach for highly advanced capabilities that matter for understanding the frontier of AI progress and safety, this is also the point at which any distinction between the two is all but meaningless, and we should be concerned about the potential for unexpected, unintended, or unwanted capabilities to be trivial to introduce in deployed models, even with limited access.