Position Paper: Uncover Scaling Laws for Large Language Models via Inverse Problems

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are large-scale pretrained models that have achieved remarkable success across diverse domains. These successes have been driven by unprecedented complexity and scale in both data and computations. However, due to the high costs of training such models, brute-force trial-and-error approaches to improve LLMs are not feasible. Inspired by the success of inverse problems in uncovering fundamental scientific laws, this position paper advocates that inverse problems can also be used to efficiently uncover scaling laws that guide the building of LLMs to achieve a desirable performance with significantly better cost-effectiveness.

1 Introduction

LLMs represent a paradigm shift in artificial intelligence, embodied by their unprecedented levels of complexity and scale in both data and computations, and their demonstrated generalization capabilities across a wide array of tasks and domains, such as natural language processing, computer vision, coding, gaming, among many others (Bommasani et al., 2021; Anthropic, 2023; OpenAI, 2023; Nijkamp et al., 2023; Dubey et al., 2024; Reid et al., 2024). These remarkable successes result from the amalgamation of several *input* ingredients, including high-quality and diverse training data, advanced modeling techniques, skillfully designed training procedures, and effective inference schemes (Antropic, 2024b; Davis, 2024; Wei et al., 2022b). The intricate interactions among these ingredients are not fully understood, yet they collectively influence the performance of large language models. To advance the development of high-performance and cost-effective models further, it is essential to uncover the underlying scaling laws that govern these *interactions*. More importantly, designing an LLM that achieves desirable performance *under resource constraint* is a highly complex challenge, as it requires careful selection and combination of data, model architecture, training procedures, and inference strategies.

As an example, when building an LLM specifically for GSM8K (i.e., grade school math benchmark), several design principles must be considered: (i) The training data should contain ample examples that foster language understanding and reasoning capabilities to ensure that the LLM can learn the nuances of math problems presented in natural language; (ii) the model architecture should be complex enough to process sequential inputs (since each question in GSM8K is described in natural language) and generate the required output formats, such as multiple-choice questions or detailed natural language explanations; (iii) the training procedure should be designed to allow the model to effectively acquire task-specific knowledge from the data (e.g., suitably defined loss functions tailored for solving math problems); and (iv) the inference scheme should guide the LLM toward generating accurate and desired outputs, as demonstrated by techniques like Chain of Thought (CoT) (Wei et al., 2022b) and ReAct (Yao et al., 2023a).

Due to the scale of the required data and modern model architectures, creating an LLM instance is an extremely costly process, e.g., GPT-4 costs over $100 million (Knight, 2023) while the cost for Gemini Ultra is estimated at over $191 million (HAI, 2024). This high expense makes building better LLMs through brute-force trial and error prohibitively costly. In contrast, DeepSeek V3 achieved state-of-the-art performance with just $5.6 million by optimizing training protocols and architecture (Liu et al., 2024a; ApX, 2025). Thus, it becomes necessary to *uncover underlying scaling laws* (e.g., the required composition and minimum size of training data or model architecture) that help *build LLMs with the desired performance and significantly better cost-effectiveness*. To this end, ***we advocate examining the class of inverse***

problems for LLMs. Inverse problems involve determining unknown parameters of an underlying model from observational data, a concept crucial in scientific and engineering domains (Groetsch and Groetsch, 1993; Vogel, 2002; Chadan and Sabatier, 2012; Gazzola et al., 2018). Tackling the inverse problems is a tried-and-true methodology for inferring and uncovering fundamental scientific laws from observations, e.g., Kepler's laws of planetary motion, Newton's law of universal gravitation, and Schrödinger's wave equation. Inspired by these successes in physics, inverse problems offer a powerful approach for uncovering the underlying scaling laws behind the behavior of LLMs.

A typical approach to tackle an inverse problem involves using a ***forward process*** to obtain observation data given some specified input and latent parameter values. However, this forward process is often costly. The ***inverse problem***, which involves identifying the latent parameter values that are consistent with a given set of observation data, is inherently very challenging due to the complexity of the search space and the lack of solution uniqueness. In the LLM context, the inverse problem requires finding the optimal combination of ingredients (i.e., data, model architecture, training procedures, and inference schemes) to build the LLMs with desirable performance, while forward processes refer to the costly training of LLMs and running model inference for task execution and evaluation.

Formally, let $\mathcal{T}$ denote the training ingredients, such as the dataset, model architecture, and training procedure, and let $\mathcal{I}$ represent the ingredients of the inference scheme (e.g., prompting method). Note that $\mathcal{T}$ includes *both* pretraining and fine-tuning, and it affects the LLM's model parameters, whereas $\mathcal{I}$ typically does not alter these parameters. Let $\boldsymbol{F}(\mathcal{T}) \rightarrow \text{LLM}$ denote the process of creating an LLM by executing the computation following the specified ingredients $\mathcal{T}$ (i.e., forward process). Let $\boldsymbol{T}(\text{LLM}, \mathcal{I}) \rightarrow C$ represent the evaluation of the LLM on a task using the inference scheme $\mathcal{I}$, resulting in a performance metric C. Therefore, we have the following two forward processes:

$$\boldsymbol{F}(\mathcal{T}) \rightarrow \text{LLM}\,, \tag{1a}$$

$$\boldsymbol{T}(\boldsymbol{F}(\mathcal{T})\,, \mathcal{I}) \rightarrow C\,. \tag{1b}$$

These two forward processes are illustrated in Fig. 1. To understand how these forward processes function, consider the above example of building an LLM for the GSM8K task, which assesses various design principles related to data (i.e., pretraining and fine-tuning datasets), model architecture, and training procedure. These ingredients are included within $\mathcal{T}$ and used in the creation process of an LLM as $\boldsymbol{F}(\mathcal{T}) \rightarrow \text{LLM}$. Subsequently, the trained LLM, along with the inference ingredients $\mathcal{I}$, is evaluated on the GSM8K task as $\boldsymbol{T}(\text{LLM}, \mathcal{I}) \rightarrow C$. Here, $\boldsymbol{T}(\cdot)$ includes both the evaluation metric (e.g., accuracy) and the evaluation dataset (i.e., GSM8K questions), and C is thus representing the accuracy of the trained LLM on the GSM8K benchmark.

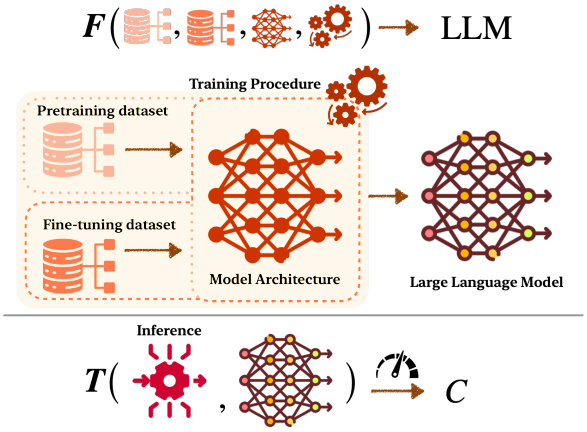

Figure 1: The forward process generates an LLM from key input ingredients and components: datasets, model architecture, and the training procedure. During inference time, other ingredients such as the prompt examples would affect the desired performance metric C.

Given practical constraints such as limited data and computational resources, tackling the inverse problems to uncover end-to-end scaling laws may be overly ambitious. Thus, as a first step, we consider simplified versions of these problems by fixing certain ingredients or focusing on a manageable subset of the problem space. Specifically, this position paper frames the following classes of inverse problems in the context of LLMs:

- In Section 2, we frame **Data Selection** as an inverse problem, focusing on integrating multiple data modalities, exploiting commonly used yet non-differentiable metrics, and enhancing selection efficiency. Solving this problem is expected to improve performance on downstream tasks while reducing the need for extensive human feedback.
- In Section 3, we frame **Inference Optimization** as an inverse problem and focus on the inference scheme used in conjunction with trained models. Solving this problem ensures trained models are adapted to underlying downstream tasks using minimal resources, without needing to modify their parameters.

- In Section 4, we frame **Machine Unlearning** (MU) verification and MU for LLMs to achieve desired performance metrics as inverse problems. Solving these problems ensures data owners that their deletion requests are fulfilled and assures model owners that harmful data are removed.

2 Data Selection

The recent successes of LLMs have been driven by training on massive and heterogeneous datasets. For example, LLaMA 3 was trained on 15 trillion multilingual tokens (Dubey et al., 2024). Previous works have established scaling laws that link data quantity to model performance (Kaplan et al., 2020; Wu et al., 2024a; Zhai et al., 2022). However, more recent studies (Xia et al., 2024; Wang et al., 2024b) demonstrate that strategically selecting data subsets can improve the performance of both LLMs and multi-modal LLMs (MLLMs) in a way even surpassing the conventional scaling laws, particularly in domains like computer vision (Sorscher et al., 2022)). This naturally raises some key questions: How does model performance scale with data quantity when data selection methods are used for MLLMs? Furthermore, how do the scaling laws vary across different stages of MLLM training, such as pretraining, fine-tuning, and alignment? We formulate data selection as an inverse problem of $\boldsymbol{T}(\boldsymbol{F}(\mathcal{T}), \mathcal{I}) \rightarrow C$. The goal is to understand how the quantity of selected training data (in $\mathcal{T}$) scales with the desired MLLM performance (C). For example, we might want to identify the minimal dataset required to train an MLLM to achieve specific performance metrics under optimal data selection. Therefore, efficient data selection can significantly reduce computational costs by prioritizing informative and representative data, thereby improving training efficiency without sacrificing performance. Furthermore, these scaling laws should be general enough so that they are applicable to a family of data selection methods instead of specific implementations (e.g., the family of influence functions (Koh and Liang, 2017) vs. its implementation DataInf (Kwon et al., 2024)).

2.1 Data Selection for Multi-Model LLMs

The remarkable successes of LLMs have led to the development of MLLMs that integrate advanced visual processing capabilities (Liu et al., 2023; Zhu et al., 2024; Dai et al., 2023). However, the rapid growth of the MLLMs and their multi-modal nature have led to instruction-tuning datasets that often rely on automated or template-based content, resulting in relatively poor-quality and redundant datasets (Liu et al., 2024d). To address this challenge, introducing smaller yet high-quality datasets can potentially maintain or even improve the performance of MLLMs. Traditional data pruning methods often require repeated gradients retrieval (Park et al., 2023) or extensive memory for storage (Yang et al., 2023), both of which become impractical for MLLMs due to their massive model sizes and data volumes. Conventional attribution methods, such as the influence function (Koh and Liang, 2017; Kwon et al., 2024) or TracIn (Pruthi et al., 2020), have not been widely adapted for MLLMs. This naturally raises a question: How to perform effective data selection for MLLMs while considering both image and text features?

Previous efforts have approached the problem as a large-scale data selection challenge, focusing on external evaluators such as established criteria (Wei et al., 2023) or intrinsic features (Liu et al., 2024d; Chen et al., 2024a). For example, Xia et al. (2024) demonstrated using a small subset of textual training data can achieve the same performance as the full dataset. The next step is to propose relatively more compute-friendly methods and generalize them to the large-scale domain of MLLMs, improving upon the standard power law scaling. The core objective of data selection research is to identify the techniques for optimally scaling training with respect to the amount of data used.

In addition, some training data points may rely primarily on a single modality (e.g., cases where images alone suffice to answer the questions). Would the scaling laws of data selection differ across different modalities, and would any particular modality have a stronger impact on the performance? To address these inquiries, one can potentially employ feature attribution methods like Integrated Gradients (Sundararajan et al., 2017) to attribute the score of each training data point to specific modalities. The multi-modal nature of data introduces an additional layer of complexity, rendering the adaptation more challenging than its conventional application in computer vision tasks. Analyzing these modality-specific scores will help better understand the relative importance of each modality and how these modalities influence the performance and, hence, uncover a universal scaling law for all modalities.

2.2 Data Selection for LLM Fine-tuning with Non-differentiable Performance Metrics

Commonly used data selection methods in LLMs are often the gradient-based data attribution methods (Han et al., 2020; Yeh et al., 2022; Schioppa et al., 2022; Grosse et al., 2023; Wang et al., 2024a), such as influence functions (Kwon et al., 2024) and TracIn (Xia et al., 2024), which quantify the impact of each data point on model parameters and next-token prediction loss. However, non-differentiable metrics C, such as semantic similarity with the ground truth (Cer et al., 2017), BLEU score (Papineni et al., 2002; Sellam et al., 2020), reward models (Ouyang et al., 2022), and LLM-as-a-judge (Zheng et al., 2023), are commonly used to evaluate the LLM performance in practice. This discrepancy between the metrics used for data selection and the metrics employed for evaluating the LLM performance can result in sub-optimal performance. Therefore, we advocate for research on how to select data for LLM fine-tuning when optimizing for commonly used but non-differentiable evaluation metrics. This problem is non-trivial because, unlike influence functions, there is no straightforward way to compute the effect or gradient of the non-differentiable evaluation metric with respect to the model parameters and training data.

One promising approach is the integration of non-differentiable evaluation metrics into the data selection method using reinforcement learning techniques, for instance, the policy gradients from the REINFORCE algorithm (Williams, 1992). By serving as a surrogate for "gradients" of the non-differentiable evaluation metrics with respect to the model parameters, these methods can lead to a novel data selection method that facilitates direct optimization towards desired (non-differentiable) evaluation criteria, thereby directly uncovering the underlying scaling laws that link the amount of training data to model performance.

2.3 Data Selection for LLM Alignment

Existing works have shown that LLM responses often do not immediately align with user intent after pretraining or fine-tuning, as LLMs can generate untruthful, unuseful, and even harmful contents (Bai et al., 2022). However, recent successes (Ouyang et al., 2022; Stiennon et al., 2020) in training LLMs using human feedback has improved alignment between user intent and LLM responses (i.e., achieving the desired alignment performance metric C) via methods like Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2024). Achieving the desired alignment depends heavily on obtaining high-quality human feedback (i.e., human labeling), which is *costly* and requires a large amount of feedback to ensure effective alignment training (i.e., RLHF/DPO). This challenge has motivated the development of a heuristic-based approach (Muldrew et al., 2024) that aimed at efficiently selecting a subset of LLM responses for human feedback. However, this heuristic-based approach lacks a principled foundation, leading to the following question: How to actively select the LLM responses for human feedback in a principled way to minimize the amount of feedback required while ensuring effective RLHF/DPO alignment training?

To address this problem, one can consider designing theoretically grounded acquisition functions (Verma et al., 2024) specifically tailored for efficient LLM fine-tuning. Such acquisition functions should account for variations in pretrained data and model architecture, which can lead to potentially different preferences for responses depending on these factors. Specifically, the acquisition functions need to consider the DPO process and quantify the uncertainty for the difference between the latent scores of two prompt-response pairs, where the latent scoring function is defined using the LLM itself (Rafailov et al., 2024). Uncovering scaling laws to efficiently acquire high-quality and diverse training data from LLM users can reduce the budget needed for data collection.

2.4 Joint Optimization for Data Selection

Previous discussions focus on data selection for a single training stage. However, different training stages improve different aspects of the model capability, and combining them can further improve the performance (Ke et al., 2023). Specifically, continued pretraining can be used to keep the knowledge of the model updated (Ke et al., 2023; Jindal et al., 2024) while instruction fine-tuning can improve its ability to follow natural language instructions (Wei et al., 2022a). Thus, a question naturally arises: How to decide the ratio of data points used in different stages under a fixed number of data points? A joint optimization approach can be plausible to find the optimal ratio (Jindal et al., 2024). Finding this optimal ratio helps uncover the underlying scaling laws of optimal data selection across different

training stages, changing the scaling law of model performance C with respect to the dataset size.

Recent results from training LLMs for low-resource languages such as SEA-LION (Singapore, 2024) demonstrate that combining continued pre-training with instruction fine-tuning achieves superior performance. On the other hand, selecting the best training data also depends on the LLM/MLLM architecture. Existing model selection works (Xia et al., 2024; Raschka, 2018; Wang et al., 2021) typically seek to find the optimal model architecture given fixed training data or the other way around. Therefore, producing the best-performing LLM/MLLM requires us to jointly select the most appropriate data and model architecture. Hence, an important research direction will be to develop algorithms that jointly select data and model architecture (Hemachandra et al., 2023) in order to optimize an LLM/MLLM's performance metric C. By doing so, deeper insights into the underlying scaling laws governing how model architecture and data selection jointly influence the LLM/MLLM's performance metric C can be developed.

3 Inference Optimization

Optimizations carried out at the inference stage significantly affect the performance of LLMs. For example, given a trained LLM, it is common practice to provide a prompt (i.e., a snippet of text) that the LLM uses to generate further text conditioned on the snippet. This represents a forward process $\boldsymbol{T}(\boldsymbol{F}(\mathcal{T}), \mathcal{I}) \rightarrow C$ in Eq. (1b), where the prompt is a component of inference ingredient $\mathcal{I}$, and inverting the process to carefully construct prompts that can instruct the LLM to perform a specific downstream task, hence achieving a desired performance measured by the metric C, is challenging. Thus, inference optimization can be viewed as an inverse problem of $\boldsymbol{T}(\boldsymbol{F}(\mathcal{T}), \mathcal{I}) \rightarrow C$ in Eq. (1b), where the goal is to design inference schemes in $\mathcal{I}$ that, when combined with a model trained on $\mathcal{T}$, achieves the desired performance metric C. Furthermore, one can also aim to uncover the underlying scaling laws at inference time with respect to optimized data, model, and compute.

3.1 Data Optimization at Inference Time

Prompts are key components of $\mathcal{I}$ during the LLM inference. A widely adopted popular prompting structure consists of instructions and few-shot demonstrations (data samples), also known as exemplars. This approach leverages LLMs' ability for *in-context learning*, which has emerged with the rapid scaling of LLMs in terms of the number of parameters, particularly since the advent of GPT-3 (Brown et al., 2020). Specifically, the LLMs can understand and perform tasks based on exemplars and instructions provided only in the context of the prompt, without relying on conventional training methods like fine-tuning on specific datasets (Liu et al., 2022). It is widely observed that the design of instructions and the selection of exemplars in the prompt significantly influence the LLM performance (Albalak et al., 2024; Rubin et al., 2022).

Prompting techniques have been introduced to steer the LLM responses towards better accuracy, tailored tone, improved focus (Antropic, 2024b,a), and reduced hallucinations (Davis, 2024; Xu et al., 2024b). In short, prompting is a tool to achieve the desired performance metric C. Despite its benefits, designing instructions and selecting exemplars for prompts typically requires a human-intensive and costly trial-and-error approach (Reynolds and McDonell, 2021; Mishra et al., 2021). Recent works have explored heuristic local search methods (Zhou et al., 2023) and evolutionary strategies (Prasad et al., 2023; Guo et al., 2024) to identify the best instructions and retrieval-based methods to find the most relevant exemplars (Liu et al., 2022; Rubin et al., 2022). However, these methods can still be costly and sub-optimal, raising the question: How can the prompts be efficiently optimized when subjected to resource constraints, such as limited computational resources or fewer queries?

Viewing the research question as an inverse problem, one can formulate the prompt optimization problem as a black-box optimization problem where the inputs are the prompts (comprising instructions and exemplars) and the output is the prompt's performance. Then, optimization techniques such as the NeuralUCB algorithm can be applied to optimize the prompt for the best performance under resource constraints (Zhou et al., 2020; Dai et al., 2022). Specifically, in the NeuralUCB algorithm, a neural network is trained on past observations to predict the LLM performance for different combinations of instructions and exemplars. This approach will help uncover underlying scaling laws and understand the effect of instructions and exemplars on LLM performance. Moreover, finding the exemplars (given a fixed budget) and instructions to achieve the best LLM performance helps to uncover the scaling law of LLM performance with respect to the number of exemplars.

This scaling law will inform the real applications to choose the least number of exemplars to achieve a target performance metric C.

Since both the data in $\mathcal{T}$ and the data in $\mathcal{I}$ affect the final LLM performance, optimizing an LLM's performance requires the *joint optimization* of in-context data in $\mathcal{I}$ and training data in $\mathcal{T}$. To efficiently solve this optimization problem, we further advocate for research into developing algorithms that automatically select the optimal combination of in-context and training data for an LLM. This approach will help us to uncover fundamental scaling laws governing the combined impact of both training $\mathcal{T}$'s and inference $\mathcal{I}$'s ingredients on the performance metric C.

Additionally, we can consider the problem of prompt optimization with human feedback, aiming to minimize the amount of human feedback required to find the best prompt that maximizes LLM performance. Specifically, we consider the inverse problem in which the performance metric C is defined as the alignment of LLM responses with human values, such as helpfulness. The goal is to optimize the prompt to improve the alignment. Recent works have shown that humans are better at providing preference feedback than giving a score, which has been the focus of prior prompt optimization works (Lin et al., 2024b; Hu et al., 2024; Wu et al., 2024b; Zhou et al., 2024b). To address this, recent works propose a framework of prompt optimization that relies solely on human preference feedback on the LLM responses (Lin et al., 2024a), demonstrating superior performance compared to prior results on prompt optimization.

3.2 Model Optimization at Inference Time

When deploying resource-efficient LLMs, understanding the scaling laws for determining optimal model configurations is crucial for effective and efficient usage (Devvrit et al., 2024). Selecting the best model configuration during inference is a critical inverse problem that aims to identify an LLM setup capable of achieving a target performance metric C with minimal computational resources. Formally, the inference-time model configuration should be considered as part of the inference ingredients $\mathcal{I}$ in Eq. (1b). The goal is to identify a model configuration that minimizes computational requirements while achieving the desired performance metric C. As model sizes increase, they require proportionately more compute and memory per generation, making them impractical in resource-constrained settings. Moreover, simply scaling model parameters does not guarantee better performance, especially in scenarios constrained by data variety and quality (Allen-Zhu and Li, 2020).

This challenge can be addressed from two perspectives: (1) selecting the optimal model at inference time from LLMs of varying sizes and capacities using methods like model valuation and selection (Xu et al., 2024a), and (2) determining the optimal number of activated routes in Mixture-of-Expert LLMs during inference time to balance efficiency and performance. Through a structured exploration of model size scaling, it is possible to determine how to adjust the model size to meet the demands of specific tasks during inference. Ultimately, uncovering the scaling laws behind model scaling at inference allows us to trade off between computational efficiency and performance.

3.3 Compute Optimization at Inference Time

The recent introduction of OpenAI's o1 model and DeepSeek R1, which are designed to facilitate CoT (Wei et al., 2022b) reasoning during inference, has induced increasing interest in scaling computational resources at inference time to improve model performance (Wu et al., 2024a; Snell et al., 2024). Existing work (Chen et al., 2024b) has demonstrated a scaling law that relates model performance to the computational resources used in inference. However, this work focuses on a single inference scheme, where the inference scheme (e.g., CoT) is an inference ingredient $\mathcal{I}$ in Eq. (1b). Besides CoT, other inference schemes, such as prompt optimization, optimization with human feedback, retrieval-augmented generation (Gao et al., 2024; Shao et al., 2024), repeated sampling (Brown et al., 2024; Gui et al., 2024), and ensemble models (Allen-Zhu and Li, 2020), have also been explored to scale inference-time compute for improving performance.

An exciting area of research is to optimize a mix of these inference schemes within a fixed computational budget, uncovering more effective model scaling behavior. Specifically, computational resources can be quantified by the number of responses generated by each of these inference schemes. Optimally allocating resources across schemes and then selecting and merging these responses improves LLM performance. Studying how the scaling law changes when inference schemes are optimally combined will provide better insight into the computational requirements neces-

sary to achieve a target performance C.

3.4 Joint Optimization at Inference Time

LLM performance is influenced by a complex interplay between data, model, and compute. Given a fixed computational cost specified by the performance metric C, it is crucial to identify the optimal combination of model configuration and inference schemes when user prompts (i.e., data) are fixed. Thus, jointly optimizing the model configuration and inference schemes can help to approach optimal LLM performance. Specifically, exploring how to allocate computational resources across different inference schemes and models should be a key focus. This approach will help uncover the underlying scaling laws that characterize how models, inference schemes, and computational budgets collectively impact LLM performance. These scaling laws can help to decide minimal model parameters and computational resources needed for LLMs to achieve desired performance, reducing the serving cost of these models in real-life applications.

4 Unlearning

Machine unlearning (MU) is the process of removing the influence of a set of training data (i.e., erased data) from a trained model to either comply with data owners' deletion requests (GDPR, 2016; CCPA, 2018) or erase harmful data to improve the model performance (Fore et al., 2024; Liu et al., 2024c; Zhou et al., 2024a). We consider two inverse problems. **Verification of MU** is an inverse problem of $\boldsymbol{F}(\mathcal{T}) \rightarrow$ LLM as given any "unlearned" model, aiming to identify if the erased data is present in the training ingredients $\mathcal{T}$. **MU techniques** can also be viewed as an inverse problem of $\boldsymbol{T}(\boldsymbol{F}(\mathcal{T}), \mathcal{I}) \rightarrow C$ as given some performance metrics (e.g., poor knowledge on weapons of mass destruction (Li et al., 2024), similar performance on the retained data as before unlearning), the goal is to design the inference ingredients (e.g., unlearning prompts) in $\mathcal{I}$, or the datasets and training procedure (e.g., use of training checkpoints, model architecture that facilitates unlearning without retraining) in $\mathcal{T}$ to achieve the desired metrics.

4.1 MU Verification

Despite the growing interest in MU for LLMs (Eldan and Russinovich, 2023; Chen and Yang, 2023; Liu et al., 2024b), one major challenge remains: How to efficiently verify whether the requested data is not present in an unlearned LLM? At first glance, we can compare the similarity of an unlearned LLM with the model trained only on the retained data (without the erased data) (Nguyen et al., 2022; Maini et al., 2024). However, such an approach requires obtaining the LLMs retrained only on the retained data, which is computationally expensive (Yao et al., 2024) or infeasible when there are computational hardware constraints. Other MU metrics try to address the challenge empirically. For example, the Membership Inference Attack (MIA) metric (Shokri et al., 2017) expects low accuracy on the erased data when assessed by an adversarial model trained to classify whether data points were members of the training dataset. These metrics fall short as they either require white-box access to the LLM (Duan et al., 2024), which is often unfeasible, or require training shadow models, which are computationally expensive (Shokri et al., 2017). Furthermore, the MIA metric depends on the adversarial model's ability to distinguish between membership and non-membership (Duan et al., 2024), which can be limited when similar data points are present in both erased and retained data (e.g., multiple news sources reporting on the same event). Thus, such a situation raises the following question: How can an efficient MU verification metric for LLMs not requiring model retraining be designed? Can the metric be intuitive and effective despite the presence of similar data?

Answering these open questions is non-trivial. One potential approach is to leverage related work on scalable and robust watermarking (Lau et al., 2024) for text data. By embedding unique watermarks into each data owner's text content before LLM training. Such watermarks should remain detectable and verifiable in LLM predictions after fine-tuning and, hence, be used to test the effectiveness of unlearning. Research on this metric could help support the scaling law that retraining-free metrics require data attribution to trace the impact of individual data points during initial training, thereby *improving unlearning procedures without the need for complete retraining*.

MU metrics can help define scaling laws governing the difficulty of unlearning erased data. Previous work (Zhao et al., 2024) explored how the tug-of-war (ToW) verification metric, which compares the accuracies of the unlearned and retrained models, is influenced by the properties of erased and retained data. It also examined how certain properties of erased data, like high memorization score, may require different MU techniques to achieve a

better ToW score. Building on these works, one can further explore how this new retraining-free metric and other MU metrics are influenced by various dataset properties, such as dataset size, watermark count, and the similarity between erased and retained data. These insights will uncover underlying scaling law that guides the selection of MU techniques and improve the reliability of metrics used for evaluating unlearning techniques.

4.2 MU Techniques

Many existing MU techniques modify the model weights (Chen and Yang, 2023; Yao et al., 2023b; Jang et al., 2023), making them unsuitable for black-box LLMs or when fine-tuning is expensive due to computational constraints. While recent approaches like offset unlearning (Huang et al., 2024) may work for black-box models, they often cause an unacceptable performance drop in the retained data (Huang et al., 2024). Moreover, prior LLM work (Pawelczyk et al., 2023) on in-context unlearning is restricted to sentiment classification and does not scale to generative tasks. Existing MU techniques may perform well on metrics like MIA but risk unlearning some retained data that are similar to the erased data (Jin et al., 2024). This situation raises a critical question: Is post-hoc unlearning (i.e., only modifying $\mathcal{I}$) for text generation feasible without compromising the performance of the retained data or introducing unintended biases?

The target performance C of an LLM is defined as minimizing the generation of harmful data or weak watermark strength based on the watermarking-based MU metric while retaining its performance on other metrics, such as the validation loss. How to achieve C efficiently by modifying the inference process $\mathcal{I}$? We advocate for research that identifies the private or harmful data (e.g., by identifying the watermarks present in generated text) and modifies $\mathcal{I}$ during inference to suppress the data influence and prevent them from being generated. Alternatively, can C be achieved efficiently by modifying the model architecture in $\mathcal{T}$ such that it is easier to unlearn?

Using the intrinsic sparsity of Mixture-of-Experts transformer paradigm (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022) to isolate the influence of data to only a few experts and thereby perform unlearning more efficiently on fewer model parameters. Overall, the aim should be to improve LLM performance on the given metric C and uncover underlying scaling laws for unlearning during inference. Specifically, this involves identifying how the metric C, like the loss on the erased and retained data, varies with the size of these datasets, computation cost, and model's ability to unlearn during inference. These scaling laws can identify the most suitable MU techniques for removing harmful knowledge from LLMs and determine how much data can be erased before performance metrics drop below a predefined threshold, beyond which retraining becomes necessary.

5 Conclusion and Future Outlook

This position paper highlights the significance of improving our understanding of the scaling laws that govern the behavior of LLMs, such as data requirements and compute scaling laws. To uncover the underlying scaling laws, we advocate for research exploring two classes of inverse problems for LLMs (i.e., Eq. (1a) and Eq. (1b)): Identifying optimal input ingredients and achieving desired performance metrics by adjusting both training and inference ingredients. Specifically, we frame data selection, inference optimization, and machine unlearning as inverse problems, each presenting unique challenges to solve. Yet, jointly optimizing them (including data, model architecture, training procedures, inference scheme, and unlearning techniques) holds great potential for advancing the development and deployment of LLMs.

Instead of iterating over the engineering efforts to further improve the empirical performance, we aim to uncover the fundamental scaling laws governing the training and inference of LLMs via inverse problems, which can lay the foundations for building better LLMs. These scaling laws can improve specific applications by providing better selection methods for training data, flexible unlearning techniques, methods with improved inference efficiency, and optimized inference schemes.

Looking ahead, future research should further explore these scaling laws and investigate how the interplay among various components and ingredients impacts performance. Additionally, advancements in machine unlearning will be crucial as models become more complex, ensuring they can adapt without compromising functionality or privacy standards. Emerging technologies and methodologies from fields like optimization theory can also offer novel tools for tackling inverse problems in LLMs. By integrating these approaches, we may uncover innovative solutions that improve the efficiency and cost-effectiveness of LLM development.

Limitations

While the inverse problem formulation offers a promising perspective for studying large language models (LLMs), it is important to recognize that not all problems in LLMs have well-defined inverse formulations. Analogous to how the inverse for a many-to-one function is ill-defined mathematically, many forward problems in LLMs, such as data aggregation or input-to-output mappings, are inherently many-to-one. This leads to potential ambiguity or ill-posedness in their inverse counterparts. Addressing these challenges will require further theoretical and methodological advancements.

Additionally, this paper focuses on a limited set of illustrative problems, such as data selection, inference optimization, and machine unlearning for LLMs, to demonstrate the potential of the inverse problem framework. A comprehensive exploration of its applicability across the broader and rapidly evolving landscape of LLM research remains an open direction. We encourage future work to uncover additional problem domains where inverse formulations may offer meaningful insights.

Ethic Statement

LLMs are largely trained on data scraped from the Internet, which may include dangerous, unsafe, biased, or inaccurate content. As a result, LLMs risk reproducing these harmful patterns in their generated outputs. Moreover, the use of scraped data raises both legal and ethical issues. The data may be copyrighted or include sensitive personal information without the consent of the data subjects. In response, we aim to mitigate these risks by improving data selection and developing machine unlearning techniques that support the removal of harmful or sensitive data and machine unlearning verification metrics to verify removal.

References

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *arXiv:2402.16827*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv:2012.09816*.

Anthropic. 2023. Claude 3.5 sonnet.

Antropic. 2024a. Prompt engineering: Giving claude a role with a system prompt.

Antropic. 2024b. Release notes: System prompts.

ApX. 2025. DeepSeek V3 training cost: Here's how it compares to Llama 3.1 (405B).

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv:2204.05862*.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, and 95 others. 2021. On the opportunities and risks of foundation models. *arXiv:2108.07258*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv:2407.21787*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proc. NeurIPS*, pages 1877–1901.

CCPA. 2018. California consumer privacy act of 2018. California Civil Code Title 1.81.5.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv:1708.00055*.

Khosrow Chadan and Pierre C Sabatier. 2012. *Inverse problems in quantum scattering theory*. Springer Science & Business Media.

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for LLMs. In *Proc. EMNLP*.

Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. 2024a. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. *arXiv:2402.12501*.

Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024b. A simple and provable scaling law for the test-time compute of large language models. *arXiv:2411.19477*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Proc. NeurIPS*, pages 49250 – 49267.

Zhongxiang Dai, Yao Shu, Bryan Kian Hsiang Low, and Patrick Jaillet. 2022. Sample-then-optimize batch neural Thompson sampling. In *Proc. NeurIPS*, pages 23331–23344.

Wes Davis. 2024. "you are a helpful mail assistant," and other apple intelligence instructions.

Fnu Devvrit, Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit S Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham M. Kakade, Ali Farhadi, and Prateek Jain. 2024. Matformer: Nested transformer for elastic inference. In *Proc. NeurIPS*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv:2402.07841*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv:2407.21783*.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? Approximate unlearning in LLMs. *arXiv:2310.02238*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, pages 1–39.

Michael Fore, Simranjit Singh, Chaehong Lee, Amritanshu Pandey, Antonios Anastasopoulos, and Dimitrios Stamoulis. 2024. Unlearning climate misinformation in large language models. *arXiv:2405.19563*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv:2312.10997*.

Mattia Gazzola, Levi H Dudte, Andrew G McCormick, and Lakshminarayanan Mahadevan. 2018. Forward and inverse problems in the mechanics of soft filaments. *Royal Society open science*, page 171628.

GDPR. 2016. General data protection regulation, article 17: Right to erasure ('right to be forgotten'). *Official Journal of the European Union*.

Charles W Groetsch and CW Groetsch. 1993. *Inverse problems in the mathematical sciences*. Springer.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, and 1 others. 2023. Studying large language model generalization with influence functions. *arXiv:2308.03296*.

Lin Gui, Cristina Gârbacea, and Victor Veitch. 2024. BoNBoN alignment for large language models and the sweetness of best-of-n sampling. *arXiv:2406.00832*.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *Proc. ICLR*.

HAI. 2024. Artificial intelligence index report 2024. *HAI - The AI Index*.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proc. ACL*.

Apivich Hemachandra, Zhongxiang Dai, Jasraj Singh, See-Kiong Ng, and Bryan Kian Hsiang Low. 2023. Training-free neural active learning with initialization-robustness guarantees. *arXiv:2306.04454*.

Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiangqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. 2024. Localized zeroth-order prompt optimization. *arXiv:2403.02993*.

James Y Huang, Wenxuan Zhou, Fei Wang, Fred Morstatter, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024. Offset unlearning for large language models. *arXiv:2404.11045*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proc. ACL*, pages 14389–14408.

Zhuoran Jin, Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Rwku: Benchmarking real-world knowledge unlearning for large language models. *arXiv:2406.10890*.

Ishan Jindal, Chandana Badrinath, Pranjal Bharti, Lakkidi Vinay, and Sachin Dev Sharma. 2024. Balancing continuous pre-training and instruction fine-tuning: Optimizing instruction-following in llms. *arXiv:2410.10739*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv:2001.08361*.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *Proc. ICLR*.

Will Knight. 2023. OpenAI's CEO says the age of giant AI models is already over. *WIRED*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proc. ICML*, pages 1885–1894.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. 2024. Datainf: Efficiently estimating data influence in loRA-tuned LLMs and diffusion models. In *Proc. ICLR*.

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2024. Waterfall: Framework for robust and scalable text watermarking. In *ICML 2024 Workshop on Foundation Models in the Wild*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv:2006.16668*.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, and 37 others. 2024. The WMDP benchmark: Measuring and reducing malicious use with unlearning. *arXiv: 2403.03218*.

Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024a. Prompt optimization with human feedback. *arXiv:2405.17346*.

Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024b. Use your INSTINCT: INSTruction optimization for LLMs using neural bandits coupled with transformers. In *Proc. ICML*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024a. Deepseek-v3 technical report. *arXiv:2412.19437*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Proc. NeurIPS*, pages 34892 – 34916.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proc. DeeLIO: Deep Learning Inside Out*, pages 100–114.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Xiaojun Xu, Yuguang Yao, Hang Li, Kush R Varshney, and 1 others. 2024b. Rethinking machine unlearning for large language models. *arXiv:2402.08787*.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024c. Towards safer large language models through machine unlearning. In *ACL Findings*.

Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. 2024d. Less is more: Data value estimation for visual instruction tuning. *arXiv:2403.09559*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. *arXiv:2401.06121*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021. Reframing instructional prompts to GPTk's language. In *ACL Findings*.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. 2024. Active preference learning for large language models. In *Proc. ICML*.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *arXiv:2209.02299*.

Erik Nijkamp, Hiroaki Hayashi, Caiming Xiong, Silvio Savarese, and Yingbo Zhou. 2023. Codegen2: Lessons for training llms on programming and natural languages. In *Proc. ICLR*.

OpenAI. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Proc. NeurIPS*, pages 27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. Trak: Attributing model behavior at scale. In *Proc. ICML*, pages 27074–27113.

Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2023. In-context unlearning: Language models as few shot unlearners. *arXiv:2310.07579*.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, Edit-based Instruction Search for Prompting Large Language Models. In *Proc. ACL*, pages 3827–3846.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Proc. NeurIPS*, pages 19920–19930.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. In *Proc. NeurIPS*.

Sebastian Raschka. 2018. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv:1811.12808*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proc. NAACL*, pages 2655–2671.

Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. 2022. Scaling up influence functions. In *Proc. AAAI*, pages 8179–8186.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proc. ACL*.

Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, and Pang Wei Koh. 2024. Scaling retrieval-based language models with a trillion-token datastore. *arXiv:2407.12854*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv:1701.06538*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proc. IEEE S&P*, pages 3–18.

AI Singapore. 2024. SEA-LION (southeast asian languages in one network): A family of large language models for southeast asia.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv:2408.03314*.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. In *Proc. NeurIPS*, pages 19523–19536.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proc. NeurIPS*, pages 3008 – 3021.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proc. ICML*, pages 3319–3328.

Arun Verma, Zhongxiang Dai, Xiaoqiang Lin, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. Neural dueling bandits. *arXiv:2407.17112*.

C Vogel. 2002. Computational methods for inverse problems. *Frontiers in Applied Mathematics/SIAM*.

Jiachen T Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. 2024a. Data shapley in one training run. *arXiv:2406.11011*.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2024b. Helpful or harmful data? fine-tuning-free Shapley attribution for explaining language model predictions. In *Proc. ICML*.

Ruochen Wang, Minhao Cheng, Xiangning Chen, Xiaocheng Tang, and Cho-Jui Hsieh. 2021. Rethinking architecture selection in differentiable nas. *arXiv:2108.04392*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *Proc. ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. NeurIPS*.

Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. 2023. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *arXiv:2308.12067*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, pages 229–256.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024a. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv:2408.00724*.

Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024b. Prompt optimization with EASE? efficient ordering-aware automated selection of exemplars. *arXiv:2405.16122*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. LESS: Selecting influential data for targeted instruction tuning. In *Proc. ICML*, pages 54104–54132.

Xinyi Xu, Thanh Lam, Chuan Sheng Foo, and Bryan Kian Hsiang Low. 2024a. Model shapley: equitable model valuation with black-box access. In *Proc. NeurIPS*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *arXiv:2401.11817*.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. 2023. Dataset pruning: Reducing training data by examining generalization influence. In *Proc. ICLR*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. *arXiv:2402.15159*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023a. React: Synergizing reasoning and acting in language models. In *Proc. ICLR*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2023b. Large language model unlearning. *arXiv:2310.10683*.

Chih-Kuan Yeh, Ankur Taly, Mukund Sundararajan, Frederick Liu, and Pradeep Ravikumar. 2022. First is better than last for language data influence. In *Proc. NeurIPS*, pages 32285–32298.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2022. Scaling vision transformers. In *Proc. CVPR*, pages 12104–12113.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Bărbulescu, Eleni Triantafillou, and Peter Triantafillou. 2024. What makes unlearning hard and what to do about it. In *Proc. NeurIPS*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proc. NeurIPS*, volume 36.

Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural contextual bandits with UCB-based exploration. In *Proc. ICML*, pages 11492–11502.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024a. Making harmful behaviors unlearnable for large language models. In *Findings of ACL*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large Language Models Are Human-Level Prompt Engineers. In *Proc. ICLR*.

Zijian Zhou, Xiaoqiang Lin, Xinyi Xu, Alok Prakash, Daniela Rus, and Bryan Kian Hsiang Low. 2024b. Detail: Task demonstration attribution for interpretable in-context learning. *arXiv:2405.14899*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *Proc. ICLR*.