

SURVEYING THE EFFECTS OF QUALITY, DIVERSITY, AND COMPLEXITY IN SYNTHETIC DATA FROM LARGE LANGUAGE MODELS

Alex Havrilla¹, Andrew Dai^{4,5}, Laura O’Mahony³, Koen Oostermeijer⁴, Vera Zisler⁴

Alon Albalak⁶, Fabrizio Milo⁷, Sharath Chandra Raparth⁸, Kanishk Gandhi⁹, Baber Abbasi¹⁰

Duy Phung⁶, Maia Iyer¹¹, Dakota Mahan⁶, Chase Blagden⁶, Srishti Gureja¹², Mohammed Hamdy¹²

Wen-Ding Li², Giovanni Paolini¹³, Pawan Sasanka Ammanamanchi⁷, Elliot Meyerson¹⁴

Georgia Tech¹, Cornell University², University of Limerick³, Aleph Alpha @ IPAI⁴, Sakana AI⁵, SynthLabs⁶, Independent⁷, Reka AI⁸, Stanford University⁹, Eleuther AI¹⁰, IBM¹¹, Cohere for AI Community¹², University of Bologna¹³, Cognizant AI Labs¹⁴

ABSTRACT

Synthetic data generation with Large Language Models (**LLMs**) has emerged as a promising paradigm for augmenting natural data over a nearly infinite range of tasks. However, most existing methods are fairly ad-hoc, utilizing a wide range of seed-datasets, LLMs, prompts, filters, and task-specific generation strategies. Given this variety, direct comparisons among synthetic data generation algorithms are scarce, making it difficult to understand where improvement comes from and what bottlenecks exist. To address this, we propose to evaluate algorithms via the makeup of synthetic data generated by each algorithm. In particular, we propose to examine the *quality*, *diversity*, and *complexity* (**QDC**) of resulting synthetic data. We choose these three data characteristics due to their significance in open-ended processes and the impact each has on the capabilities of downstream models. We find quality to be essential for *in-distribution* model generalization, diversity to be essential for *out-of-distribution* generalization, and complexity to be beneficial for both. Further, we emphasize the existence of Quality-Diversity trade-offs in training data and the downstream effects on model performance. We then examine the effect of various components in the synthetic data pipeline on each data characteristic. This examination allows us to taxonomize and compare synthetic data generation algorithms through the components they utilize and the resulting effects on data QDC composition. This analysis extends into a discussion on the importance of balancing QDC in synthetic data for efficient reinforcement learning and self-improvement algorithms. Analogous to the QD trade-offs in training data, often there exist trade-offs between model output quality and output diversity which impact the composition of synthetic data. We observe that many models are currently evaluated and optimized only for output quality, thereby limiting output diversity and the potential for self-improvement. We argue that balancing these trade-offs is essential to the development of future self-improvement algorithms and highlight a number of works making progress in this direction.

CONTENTS

1	Introduction	3
1.1	Related Topics and Surveys	5
2	Defining Data Quality, Diversity, and Complexity	6
2.1	Defining Dataset Quality	7
2.2	Defining Dataset Diversity	9
2.3	Defining Dataset Complexity	11
3	The Effects of Data QDC on Model Performance	14
3.1	The Effect of Quality	14
3.2	The Effect of Diversity	15
3.3	The Effect of Complexity	17
3.4	Trade-offs in QDC and Impacts on Performance	18
4	QDC Promoting Mechanisms in Synthetic Data Algorithms	19
4.1	Quality Promoting Mechanisms	21
4.2	Diversity Promoting Mechanisms	23
4.3	Complexity Promoting Mechanisms	26
4.4	Impacts on Recursive Self-Improvement	28
4.5	QDC-Aware Synthetic Data Generation Algorithms	30
5	QDC Synthetic Data Algorithms Outside Common LLM Tasks	31
6	Conclusions and Open Questions	32
A	Table of QDC Metrics	54
B	Table of QDC Mechanisms in Synthetic Data	57

1 INTRODUCTION

Synthetic data generation has emerged as a promising approach to enhance the capabilities of large language models beyond traditional supervised fine-tuning datasets. This development has led to the creation of a diverse set of synthetic data generation algorithms for a variety of tasks and domains. The majority of these algorithms follow a two-step process: First, leverage existing large language models to gather a large set of task prompts and sample continuations. Second, filter the generated dataset to eliminate “low-quality” samples. Their main goal is to maximize the “quality” and quantity of synthetically generated data. Relatively little effort is spent seeking to carefully understand what intrinsic characteristics of the data most impact downstream generalization. While these algorithms are a natural place to start, this type of approach is inefficient, leading to most synthetically generated data being discarded (Zhou et al., 2023a).

This survey aims to clarify the impact of synthetic data generation on downstream model generalization by analyzing three key data characteristics: *quality*, *diversity*, and *complexity*. Informally, quality measures the “noisiness”, “correctness”, or “alignment” of data with a desired target distribution Q . Diversity measures the “self-similarity” or “coverage” of data. Complexity intuitively captures some notion of the “difficulty” or “compositionality” of data. We choose these characteristics for their importance so far in assessing and building artificial open-ended systems, an emerging paradigm that can be applied to iterative self-improvement of models (Hughes et al., 2024). The field of Quality-Diversity has established quality and diversity measures as effective proxies in encouraging increasingly novel and interesting/learnable/valuable synthetic artifacts, often of increasing complexity, and synthetic data generation is natural application of this framework (Pugh et al., 2016; Cully & Demiris, 2017; Chatzilygeroudis et al., 2021). The importance of data quality, diversity, and complexity is also reflected in many prominent synthetic data generation methods, which explicitly or implicitly aim to maximize at least one of the above (though rarely all three together) (Xu et al., 2023; Gunasekar et al., 2023; Wang et al., 2023c).

Through this quality-diversity-complexity (**QDC**) lens we investigate three closely related research questions:

- **RQ1:** How should quality, diversity, and complexity be defined? How are these quantities measured in LLM literature?
- **RQ2:** How do quality, diversity, and complexity in training data impact model generalization?
- **RQ3:** How do existing synthetic data generation algorithms promote quality, diversity, and complexity?

The answers to these questions can enable the design of more sample-efficient synthetic data generation algorithms with better model generalization and self-improvement abilities.

In Section 2 we investigate **RQ1**. We begin by providing abstract, high-level definitions of quality, diversity and complexity in data. Informally, each characteristic is fairly intuitive: quality measures the “noisiness” or “correctness” of data, diversity measures the “coverage” and “self-similarity” of data, and complexity measures the “difficulty” or “compositionality” of data. Yet, despite these intuitive informal definitions, many different practical measures for each characteristic exist in the literature, and these practical measures vary in their utility. Some are generally applicable, others domain-specific. Some correlate with downstream metrics of interest, while others do not (depending on the task).

Armed with a better understanding of how data quality, diversity, and complexity are measured in practice, in Section 3 we survey the effects of each characteristic on model performance. We come away with three key takeaways in answer to **RQ2**:

- Data quality is essential for *in-distribution* generalization.
- Data diversity is essential for *out-of-distribution* (**OOD**) generalization.

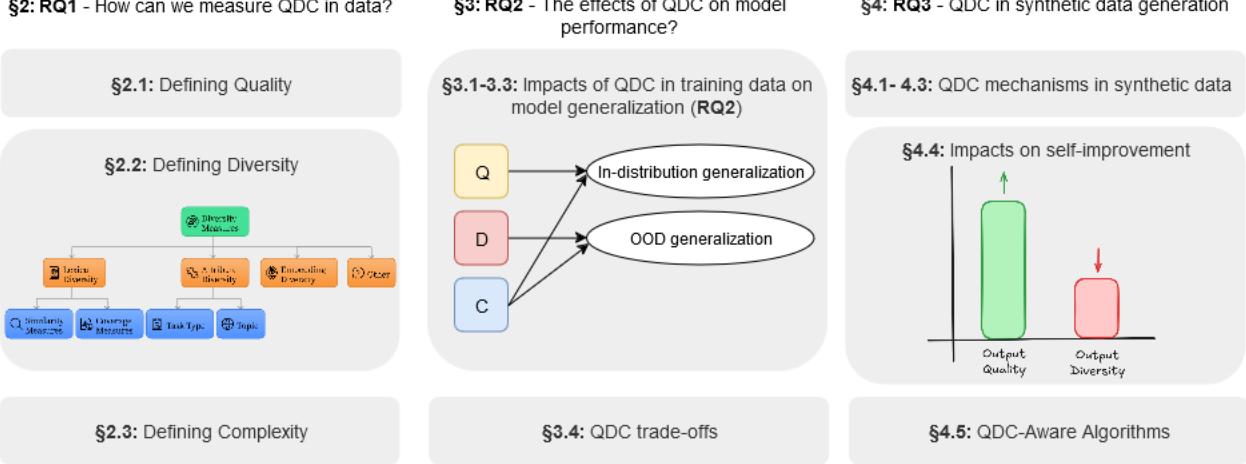


Figure 1: A summary of our research questions, and key findings discussed in greater detail in the relevant sections and subsections.

- Appropriate levels of data complexity benefit both in-distribution and OOD generalization.

Further, trade-offs often arise between the quality and diversity of training data. In such situations, decisions must be made on how to prioritize the three characteristics. This gives rise to a potential QDC *generalization frontier* as different mixtures of quality, diversity, and complexity change how downstream models generalize.

Finally, we move to investigate **RQ3** in Section 4. We begin by taxonomizing existing synthetic data generation approaches through the QDC perspective. This is done by classifying common components of synthetic data pipelines as “quality-promoting”, “diversity-promoting”, or “complexity-promoting”. This results in a continuum of methods which mix and match various components, resulting in synthetic data with varying degrees of quality, diversity, and complexity. We find the majority of algorithms employ relatively simple methods of promoting quality, often by sampling from a large SOTA model. Similarly, many methods promote diversity simply by initializing sampling using a large seed-dataset. Often complexity is not explicitly considered at all. We then discuss the **impact of QDC data characteristics on the synthetic data generation process itself** with applications to model self-improvement. Analogous to the QDC trade-offs found in Section 3, we find several works suggesting a trade-off between models generating high-quality data and models generating highly diverse data, i.e., a trade-off between model output quality and model output diversity. We argue that, due to this trade-off, future algorithms for synthetic data generation must carefully balance QDC mixtures of synthetic training data for optimal self-improvement. However, the majority of algorithms and benchmarks today optimize for quality alone. As a result, model output diversity and the potential for bigger self-improvement gains suffer. Finally, we highlight a few approaches directly inspired by more classical quality diversity (QD) search algorithms (Lehman & Stanley, 2011b; Mouret & Clune, 2015) (cf. QD paragraph in section 1.1), which attempt to more explicitly control the quality and diversity of generated data. These *QD synthetic data generation algorithms* explicitly aim to generate data which has both maximal quality and diversity in a sample efficient way, thus receiving the benefits of both characteristics.

In Section 5 we survey evolutionary/quality-diversity algorithms for synthetic data generation with LLMs outside of common benchmarks tasks. We then conclude the survey in Section 6 by reviewing the key takeaways highlighted at the end of earlier sections. Notable takeaways include:

Takeaways

- Quality largely improves in-distribution generalization and diversity largely improves OOD generalization. Appropriate levels of complexity can improve both.
- Quality and diversity often trade-off in training data.
- Many existing models/methods are heavily optimized and evaluated for model output quality, thereby limiting synthetic data diversity.

We also summarize the list of collected open problems highlighted at the end of earlier sections. Notable open questions include:

Open Questions

- Establishing benchmarks jointly measuring the quality *and* diversity of model output and synthetic data.
- Designing better algorithms explicitly controlling for trade-offs between model output quality and output diversity.
- Better understanding of the trade-offs between complexity and the other two characteristics.

See Figure 1 for a visual outline of the survey’s organization and key-takeaways.

1.1 RELATED TOPICS AND SURVEYS

Synthetic Data Generation Synthetic data generation algorithms utilize generative models to create “synthetic” data points which can be used downstream for training, benchmarking, etc. A few recent surveys have investigated synthetic data generation (Bauer et al., 2024; Guo & Chen, 2024; Liu et al., 2024a; Long et al., 2024). Bauer et al. (2024) provide a broad overview of synthetic data generation across both vision and language throughout the past decade. Specifically, they highlight the difficulty of benchmarking existing algorithms. Guo & Chen (2024) and Liu et al. (2024a) focus their surveys on synthetic data generation practices that have developed more recently, with a primary spotlight on LLMs. Discussion is centered around applications in various domains (e.g., reasoning and multi-modality). Less emphasis is put on comparing characteristics of the data generated by different algorithms in the same domain. Long et al. (2024) look at LLM-driven synthetic data generation, curation, and evaluation of synthetic data without as much emphasis on downstream impacts.

Data Selection Data selection is the task of selecting a susbet of desirable training samples from a larger training dataset \mathcal{D} . It plays an important role in a number of synthetic data generation pipelines, and is an important topic that has been previously surveyed (Albalak et al., 2024; Qin et al., 2024; Wang et al., 2024c). Albalak et al. (2024) present a systematic survey of data selection methods focused on language model pre-training, and of particular importance to the current work, they point out that data selection methods generally fall under one of two categories: *distribution matching* and *distribution diversification* methods, which are closely related to quality and diversity, respectively. Qin et al. (2024) present a survey of data selection methods for instruction tuning, finding that methods can be categorized into three groups: quality-based, diversity-based, and importance-based. Wang et al. (2024c) also present a survey on data selection for instruction tuning; however, their work focuses on describing how a sample of popular datasets were created.

Quality-diversity (QD) and open-endedness Quality-diversity (QD) algorithms (Pugh et al., 2016; Cully & Demiris, 2017; Chatzilygeroudis et al., 2021) are a class of search algorithms originating from evolutionary computation (Lehman & Stanley, 2011b; Mouret & Clune, 2015) that aim for both quality and diversity among a population of solutions and artifacts, two of the key dataset attributes covered in this survey. Such methods are inspired by the creativity of natural evolution in discovering diverse solutions (e.g. species) that also excel within the diverse niches they fill in the environment, and evolve into increasingly diverse and fit species in the population. QD combines traditional objective optimization with insights from novelty search (Lehman & Stanley, 2011a), an open-ended algorithm that overcomes local optimality through the continual accumulation of novel solutions. By generating and maintaining a collection of diverse solutions, and then selecting for the next generation of solutions that are either increasingly novel, or optimized improvements to existing solutions within similar niches, QD leverages this growing collection to discover more diverse, high-quality solutions without having to trade off between quality and diversity. QD methods have been applied recently for their illuminating search capabilities towards generating diverse, high-quality synthetic data for training models (cf. Section 4). QD research is aligned with the study of open-ended systems, or open-endedness (OE) (Soros et al., 2017; Song, 2022), a broader term for the field that emerged from the study of open-ended evolution (Packard et al., 2019). OE studies systems designed to generate and discover continually “novel” and “interesting” outcomes, and takes inspiration from real-world open-ended processes such as natural evolution, and human collective innovation. OE has become a key subject for providing new ways to tackle challenges in AI research, for example, towards generating open-ended synthetic data from which models can learn (Jiang et al., 2023; Sigaud et al., 2023; Hughes et al., 2024; Samvelyan et al., 2024b). LLM-based tools may provide new opportunities to advance surveyed methods in synthetic data generation, as OE and evolutionary methods are becoming more integrated with LLM components (Lehman et al., 2022; Meyerson et al., 2023; Zhang et al., 2023; Wu et al., 2024a; Chao et al., 2024).

This survey Our survey complements the above perspectives on synthetic data generation, open-endedness, and quality diversity (QD). We unify these findings to form a broader view of how future work in data generation and selection can emerge from distinct fields. This is done by providing a taxonomy for synthetic data through the lens of quality, diversity, and complexity, thus providing a better framework for understanding trade-offs and inefficiencies in the synthetic data generation process. We ground this framework with concrete takeaways and best practices in popular domains, including pre-training, instruction-tuning, and reasoning. We conclude by offering a list of open-problems and promising future research directions to better understand the intersection of synthetic data generation and QDC. The above summary of existing works and surveys highlights the important gaps that this survey fills.

2 DEFINING DATA QUALITY, DIVERSITY, AND COMPLEXITY

Suppose we have some sample space $\Omega = \mathcal{X} \times \mathcal{Y}$ where each $\omega = (x, y) \in \Omega$ is an input-output sample pair. Further suppose we have a set of tasks τ_1, \dots, τ_k defined as probability measures on Ω . Finally, suppose we are given a large n -sample training dataset $\mathcal{D} \in \Omega^n$ and a model \mathbf{M}_θ . Note that \mathcal{D} need not necessarily be sampled depending on the tasks τ_1, \dots, τ_k . Given some objective/loss l , it is often of interest to find characteristics of \mathcal{D} which can be used to predict the downstream performance of \mathbf{M}_θ on tasks τ_1, \dots, τ_k . In this survey we are interested in understanding the impact of three intuitive characteristics: dataset quality $Q(\mathcal{D})$, dataset diversity $D(\mathcal{D})$, and dataset complexity $C(\mathcal{D})$.

Despite being intuitive, widely used terms in ML, defining exactly what is meant by dataset quality, diversity, and complexity can be a surprisingly difficult task. Numerous implementations of proxy measures exist which attempt to capture our intuitive notions of these characteristics. To further complicate matters, different measures correlate better with downstream metrics of interest (such as model performance) in different settings. This makes choosing a single definitive measure practically impossible. Instead, we attempt to define abstract notions of quality, diversity, and complexity that line up with our intuitions as best as possible.

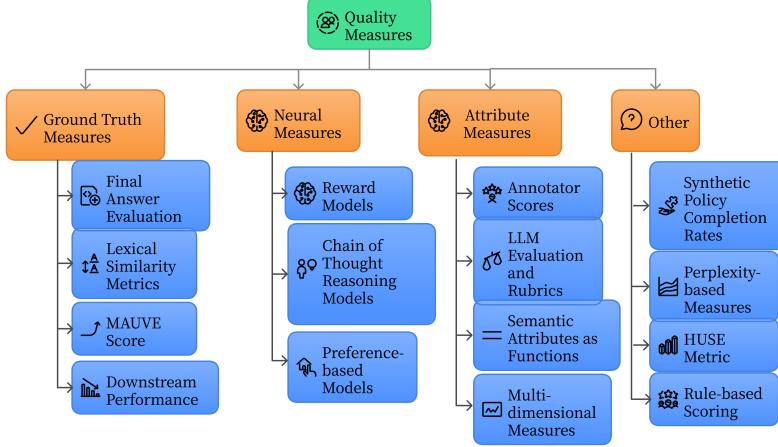


Figure 2: Quality Metrics

With these formalisms in place, we survey the many different practical implementations of each characteristic found in the literature. Along the way, we discuss trade-offs of different types of implementations and their utility in predicting and improving downstream metrics of interest such as model performance. We determine the overall utility of a particular metric by assessing three key qualities: (1) *applicability*, how widely applicable the metric is across domains, (2) *cost*, how expensive is the metric to compute, and (3) *performance*, how closely does the metric correlate with downstream model performance. Note: all the metrics we discuss are collected in tables in Appendix Section A.

2.1 DEFINING DATASET QUALITY

Fix a target task τ as a probability distribution on the sample space Ω . Informally, we say the quality $Q(\mathcal{D})$ of dataset \mathcal{D} aims to measure the “noisiness” or “correctness” of samples in \mathcal{D} with respect to τ . High-quality datasets with respect to τ should be entirely contained inside the support of τ in the sample space Ω . Low-quality datasets will contain many samples far out of distribution of τ . Often, quality measures are defined at the sample level $Q_\Omega : \Omega \rightarrow \mathbb{R}$. For example, the “quality” of a piece of code could be determined by how many unit tests it passes. In these common cases we could regard the corresponding task distribution τ which concentrates on the set of code samples passing all unit tests. A quality measure for the entire dataset can then be extracted by averaging the sample level quality: $Q(\mathcal{D}) = \frac{1}{n} \sum_{\omega \in \mathcal{D}} Q_\Omega(\omega)$.

Implementations of Q can be categorized into four groups: ground truth measures, neural measures, attribute measures, and hybrid / other.

Ground truth reference measures: Most often the data quality of a sample ω is measured by comparing to a corresponding ground truth sample ω^* . The sample level quality $Q_\Omega(\omega)$ is then the similarity of ω to ω^* . In domains where there is an intended final answer A^* , e.g., math (Yu et al., 2024; Toshniwal et al., 2024; Singh et al., 2024), coding (HumanEval, Pourcel et al. (2024)), and other reasoning settings, the quality of a sample ω can be easily measured as

$$Q(\omega) = \begin{cases} 1 & A_\omega = A^* \\ 0 & \text{otherwise.} \end{cases}$$

However, this comes with the obvious drawback of ignoring any intermediate steps in the chain of thought used to produce the final answer A_ω . Further, more complex, open-ended tasks such as theorem proving and creative writing do not have a final answer. In these cases, another measure must be used to assess quality. When a ground truth sample ω^* is available, lexical measures of similarity to ω^* , such as Rouge or Bleu/SacreBleu (Samvelyan et al., 2024b) can be used. While these measures are popular for less complex tasks such as paragraph-level summarization (Stiennon et al., 2022), lexical similarity is often insufficient for evaluating correctness on more complex tasks such as instruction following (Liu et al., 2024b). In such cases, alternative metrics have been proposed. MAUVE (Pillutla et al., 2021; Ye et al., 2022a) proposes to approximate the area under the divergence curve between the given data and a reference dataset. Other works simply define quality as performance on a downstream benchmark: Zhou et al. (2023a); Viswanathan et al. (2023); Gandhi et al. (2024a).

Neural measures: When training data is available, *reward models* can be trained and generalize to unseen samples. The simplest example of a reward model is a classifier trained using labeled data (e.g., a toxicity classifier (Chakrabarty, 2019)). A number of works have trained binary classification models using data that is known to be high-quality as the positive class examples, and unfiltered data as the negative class examples (Brown et al., 2020b; Gao et al., 2020; Du et al., 2022; Xie et al., 2023; Li et al., 2024b). In domains such as reasoning, specialized reward models such as Outcome-based reward models (ORMs) (Uesato et al., 2022) are used, which are trained with a classification objective at the step level supervised by the the final answer (Pang et al., 2024; Tian et al., 2024; Havrilla et al., 2024b). The resulting reward model outperforms final answer classifiers and (somewhat) generalizes to intermediate steps. Process-based reward models (PRMs) (Lightman et al., 2023) further improve on ORM by training a step-level classifier with supervised labels at each step. The PRM can then be used as a quality measure of each step of a solution ω in addition to the full trace. However, collecting human annotations for training PRMs is expensive. Recent work (Havrilla et al., 2024b; Wang et al., 2024d) has investigated automating this process with synthetic data. Ye et al. (2024b) use synthetic natural language critiques generated by LLMs to provide additional feedback and then train a reward model that predicts a scalar reward on top of them.

A related line of work trains so called Bradley-Terry reward models contrastively on ordered preference data of the form (ω_-, ω_+) where ω_+ is the accepted sample and ω_- is the rejected sample (Ziegler et al., 2020; Ouyang et al., 2022; Bai et al., 2022a; Kirk et al., 2024; Bukharin & Zhao, 2024; Havrilla et al., 2023; Bai et al., 2022a; Dubey et al., 2024b). Classifiers can also be used in this setting by conditioning on a ground truth reference, as in the Stanford Human Preferences Dataset (Ethayarajh et al., 2022). A quickly growing line of work (Ankner et al., 2024; Zhang et al., 2024c; Mahan et al., 2024) trains reward models with the ubiquitous next-token prediction objective instead of a discriminative objective to generate critiques or explanations before providing a quality score. Such approaches appear to generalize further out of distribution than their purely discriminative counterparts due to their CoT reasoning capability and ability to effectively utilize more inference-time compute. Also related is the zero-shot or few-shot use of LLMs-as-a-judge (Zheng et al., 2023) to perform in-context preference selection over multiple candidates.

Attribute measures: When ground truth data is not available to assess data quality another option is to rely on data annotators to assess sample *attributes* relevant to quality. Abstractly, a sample attribute is a generic function $T : \Omega \rightarrow [0, 1]$ measuring some semantic property of the sample. $T(\omega) = 1$ signifies that ω has the semantic property, $T(\omega) = 0$ signifies that ω does not have the semantic property, and $0 < T(\omega) < 1$ signifies that ω somewhat has the property. A quality score Q for a sample can then be recovered via some combination f of the attributes $Q(\omega) = f(T_1(\omega), \dots, T_k(\omega))$.

Each attribute function T is often implemented as a discrete Likert score (Likert, 1932) over the integers 1-N (which can then be normalized into the range $[0, 1]$). For example, Bai et al. (2022a) recruit annotators to measure helpfulness, harmlessness, and honesty of AI assistant responses on a scale of 1-5, and Stiennon et al. (2022) recruit annotators to measure the coverage, clarity, and fidelity of a summary. LLMs themselves are increasingly being used as annotators to judge how well a sample adheres to a written constitution (Bai

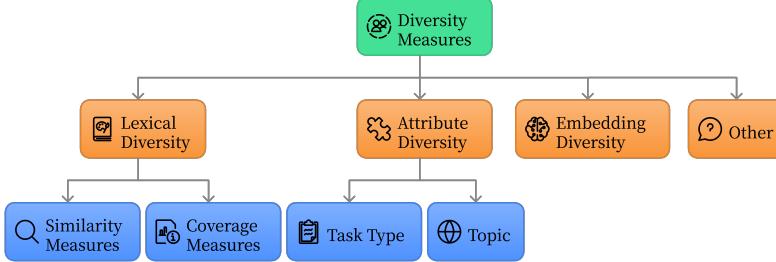


Figure 3: Diversity Metrics

et al., 2022b; Samvelyan et al., 2024b)), to measure the difficulty of a problem (Gandhi et al., 2024b), or to compare data examples across various dimensions of quality (Wettig et al., 2024).

A related approach to evaluating quality, uses sample *rubrics* to assess the correctness of open-ended reasoning and language tasks. Sawada et al. (2023) use GPT-4 to write solution rubrics for hard STEM problems given example ground truth answers. While previous reference solution metrics were based on Bleu/Rouge, rubric based approaches generalize better to measuring important attribute based sample features. For example, EvolQuality evolved solutions of increasing quality in a manner similar to WizardLM (Luo et al., 2023b) and then applies GPT-4 as a judge to assess the quality of each sample of the sequence in the same shared context (Liu et al., 2024b). Zhong et al. (2022) propose UniEval, a method that assigns multidimensional quality measures $T_i \in [0, 1]$, $i \in \{1, \dots, k\}$, where T_i is defined as

$$T_i(\omega) = \frac{P(\text{Yes}|\omega, q_i)}{P(\text{Yes}|\omega, q_i) + P(\text{No}|\omega, q_i)}, \quad (1)$$

where $P(\cdot)$ is the probability of the model to generate a specific word, and the input q_i is a question specific to the quality being measured. For instance, the question to measure coherence could be “Is this a coherent extraction of the document?”. The same process can be repeated for other dimensions including fluency, understandability, naturalness, consistency, groundedness, engagingness, and others. (Li et al., 2024e) introduce an automated, rule-based framework for filtering for high-quality training data. They generate a diverse set of rules using an LLM, before rating a batch of data based on these rules. Following this, they use the determinantal point process (DPP) from random matrix theory to select the most orthogonal score vectors, thereby identifying a set of independent rules used to evaluate all data to select for training. This rule evaluation metric is designed to promote low correlation and high diversity among rules.

Hybrid/other measures: Several other miscellaneous measures of dataset quality can be found in the literature. Fontaine et al. (2021b) generates synthetic game levels and uses the completion rate of an expert policy as a measure of quality. A similar idea is explored in Havrilla et al. (2024a) by evaluating the quality of synthetically generated questions by the solve rate of a fixed student policy. Other works use perplexity as a measure for quality (Wenzek et al., 2020; Sharma et al., 2024). Hashimoto et al. (2019) introduces the HUSE metric for jointly measuring the quality and diversity of machine generated text. HUSE utilizes averaged annotator quality ratings for a per-sample quality measure and length normalized model log-likelihood to measure diversity.

2.2 DEFINING DATASET DIVERSITY

A diversity measure $D(\mathcal{D})$ aims to capture the “self-similarity” and “coverage” of \mathcal{D} over the entire sample space Ω . A highly diverse dataset should be uniformly distributed on Ω . This motivates us to consider formally defining the diversity of \mathcal{D} as the distance between \mathcal{D} and the uniform measure on Ω under some

chosen probability metric. Importantly, this means the diversity of \mathcal{D} is not necessarily dependent on a target task τ . However, in some settings where we have a good prior over the sample space Ω it may make more sense to adapt our notions of diversity to be uniform with respect to the prior instead of uniform over the entire sample space. For example, in the language setting a more sensible uniform prior may be the uniform distribution over all natural language sentences instead of uniform over all possible strings up to a fixed finite length.

Unlike quality, it is difficult to define diversity at the sample level. We categorize practical diversity measures into four groups: lexical, attribute, embedding, and miscellaneous as summarized in Figure 3.

Ground truth measures: **Lexical similarity measures** can be used to capture a relatively simple type of diversity in text data by comparing n-grams between source and target datasets. Scores like ROUGE (Lin, 2004; Gandhi et al., 2024a), Bleu (Papineni et al., 2002a; Samvelyan et al., 2024b; Ye et al., 2022b; Zhu et al., 2018; O’Mahony et al., 2024), and INGF (Yu et al., 2023b; Kirk et al., 2024) measure pairwise similarity between samples in a dataset \mathcal{D} by pairing n-gram overlap between samples. A diversity measure for \mathcal{D} is then assigned by averaging all pairwise similarity measures. We refer to these kinds of diversity measures as *similarity measures*. Dataset vocabulary size, i.e., the number of unique words in a dataset, can also be used as a simple diversity measure which does not rely on pairwise lexical comparisons between samples (Yu et al., 2023b). We refer these types of diversity measures as *coverage measures*. Measure of Textual Lexcial Diversity (**MTLD**) (McCarthy & Jarvis, 2009; Cao et al., 2024) is another measure of lexical diversity which analyzes the average number of words in a row that maintain a certain Type-Token Ratio (TTR). The “distinct” metric (Li et al., 2016a) measures diversity of a collection of texts as the ratio of distinct tokens to the total number of tokens in the text. The expectation adjusted distinct metrict (**EAD**) additionally normalizes by the expected number of distinct tokens under some prior.

Lexical diversity measures are general-purpose, requiring no special domain knowledge for application. However, for the same reason they often do not capture more relevant/interesting notions of diversity in domain specific settings: e.g., the type of skill used to solve a programming problem (Pourcel et al., 2024).

Attribute diversity: Similar to attribute-based measures of quality, attribute-based measures of diversity also require domain-specific knowledge to define attribute functions $T_i : \Omega \rightarrow [0, 1]$. Each attribute function T_i measures where a given sample ω falls in the semantic attribute T_i . The collection of attribute measures $T(\omega) = (T_1(\omega), \dots, T_k(\omega))$ defines an attribute description for each sample ω . Concretely, the attributes computed for a dataset vary from task to task. Many instruction tuning papers (Wang et al., 2022; Li et al., 2024d; Zhang et al., 2024a; Lu et al., 2023b) categorize instruction by task type (summarization, question answering, translation, etc.) to measure instruction diversity. Some papers (Zhou et al., 2023a; Wang et al., 2022) also annotate the topic, e.g., politics, health, sports, etc., of each instruction. Wang et al. (2022) additionally classifies instructions by language to measure instruction diversity along three axes: type, topic, and language. Other works describe attributes using *behavioral descriptors*. Samvelyan et al. (2024b) classifies red-teaming prompts by what constitutional rules are violated. Fontaine et al. (2021b) counts the number of attributes contained in a description of a tile-based Mario level. Pourcel et al. (2024) categorizes solutions to programming puzzles by tracking what skills are used in the solution (e.g., arrays, graphs, dynamic programming, etc.). Tian et al. (2024); Havrilla et al. (2024a) identify solutions to math problems by their orders of operations. Gandhi et al. (2024a) measures the diversity of learned search algorithms playing the Game of 24 by the number of unique states they visit while finding a solution.

The main difficulty with measuring attribute-based diversity is in accurately measuring the desired attributes of an unstructured text sample ω . Implementations of these measures fall largely into two approaches: (1) relying on human annotators or (2) using LLMs as judges to automatically annotate samples. Earlier works (Zhou et al., 2023a; Wang et al., 2022) utilize attribute labels coming exclusively from human annotators. The marked improvement in LLM capability made automatic annotation of attributes only recently possible. Several works have already experimented with this approach (Samvelyan et al., 2024b; Pourcel et al., 2024;

Bradley et al., 2023; Bai et al., 2022b), with results varying depending on the complexity of the attribute. Yu et al. (2023b) experimented with GPT for automated attribute discovery. Instag (Lu et al., 2023b) uses GPT-4 to generate sets of skill/topic labels for instruction tuning data.

Many different diversity measures can be implemented once a set of attributes $\{T(\omega) : \omega \in \mathcal{D}\}$ has been measured for every sample in the dataset. Similarity-based measures can be implemented by computing the pairwise similarities between attribute profiles. Pourcel et al. (2024) implement a similarity-based measure for python puzzle solutions by computing binary indicator attributes for programming skills. Hamming distance can then be used to measure distance between the binary attribute profiles. A coverage-based approach can also be taken by measuring the total number cells in the boolean grid occupied by a sample. Samvelyan et al. (2024b) use a similar coverage-based approach to measure diversity of red-teamed LLM responses based on what constitutional rules are violated. QD-score (Tjanaka, 2022) jointly measures the quality and diversity of solutions in a grid by summing the fittest sample in each cell: $\sum_{c \in \mathcal{G}} f(c)$.

In general, attribute-based measures of diversity are effective at measuring interesting, domain-specific features which capture useful notions of diversity. However, measuring such attributes can be difficult and expensive. LLM-as-a-judge approaches have made automating attribute labeling more feasible, but struggle on more complex tasks/attributes.

Embedding diversity: Another set of popular dataset diversity measures are computed by embedding the dataset \mathcal{D} as latent space embeddings (Reimers & Gurevych, 2019; Zhao et al., 2019). Pourcel et al. (2024) take a similar approach by embedding solutions to programming puzzles. Many other works (Yu et al., 2023b; Kirk et al., 2024; Yu et al., 2024) embed natural text using Sentence-BERT (Reimers & Gurevych, 2019) and measure average pairwise cosine similarity (**APS**) between the embeddings. Cao et al. (2024) measure diversity using i th-Nearest neighbor distance in SentenceBERT embedding space. Similarly, SemDeDup (Abbas et al., 2023) and D4 (Tirumala et al., 2023) measure the diversity of data points by embedding data, however they use a generative model, and their goal is to remove semantically duplicated data. Ding et al. (2024) train an embedding model to measure sample diversity using a triplet-based contrastive objective. They do this by collecting triplet preferences of the form $y_1 \sim y_2$ or $y_1 \sim y_3$ where $y_1 \sim y_2$ indicates y_1 is more similar to y_2 than y_3 . Lee et al. (2023) measure the *diversity coefficient* of pre-training data by computing the Fisher information matrix of gpt-2 gradients. Bukharin & Zhao (2024) is a type of coverage measure which measures the diversity of instruction following datasets using the *facility location function* $d(A) = \sum_{v \in V} \max_{a \in A} sim(a, v)$ which relies on a background dataset V to measure the diversity of A . Cosine similarity between embeddings is used as the similarity measure.

Measures of diversity using latent space embeddings do not require domain knowledge and thus serve as general purpose diversity measures (provided a general enough embedding model). However, for more domain specific/complex tasks embeddings may fail to capture certain attribute-based features essential to comparing samples in the task. This may result in a weaker correlation with downstream metrics, when compared to attribute measures, depending on the domain.

Other diversity measures: Many other miscellaneous measures of dataset diversity can be found in the literature. Kirk et al. (2024) measures the diversity of summaries for a fixed passage by computing the average number of pairwise contradictions via a natural language inference model. Singh et al. (2024); Havrilla et al. (2024a); Toshniwal et al. (2024) measure the solution diversity of math problems via the pass@ n score on the test set i.e., the number of problems that can be solved with an n -solution sample budget.

2.3 DEFINING DATASET COMPLEXITY

Complexity is a third essential characteristic of data that intuitively measures some notion of the “difficulty” or “compositionality” of a sample. While less commonly considered than quality and diversity, improving

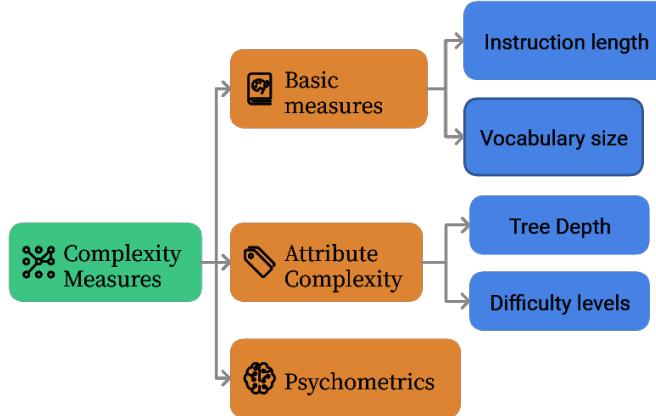


Figure 4: Complexity metrics.

output complexity is essential for any algorithm for recursive self-improvement through synthetic data. This is because the output complexity of a model reflects its capability to compose distinct concepts across domains. A model with high-quality and highly-diverse output may reliably recall the data it was trained on across a variety of tasks. However, if the model has low output complexity, it may completely fail to learn how to combine pieces of knowledge it has already learned. Learning how to do this type of composition is essential for intelligence.

Despite its intuitiveness, complexity is a notoriously difficult concept to formalize (Mitchell, 2009). *Kolmogorov complexity* (Li & Vitányi, 1997) proposes to measure complexity of a sample as the length of the shortest program generating the sample ω with respect to a fixed programming language. *Logical depth* (Bennett, 1988) is another notion of complexity which instead measures the minimum amount of *time* needed to compute the sample ω when minimizing over all programs in a fixed programming language. While such notions are quite general, they are intractable to compute in common situations, necessitating practical alternatives. However, they do point to the common theme that most measures of complexity implicitly depend on a fixed background set of primitives used to construct the sample ω .

To help express the foundations of the nature of complexity that could be studied under different measures, we give a more actionable definition for complexity. We define complexity $C(\omega)$ of a sample $\omega \in \Omega$ as its size under a fixed representation scheme. This also aims to align closely with the broader definition of complex systems as described in Mitchell (2009):

A system in which large networks of components with no central control and simple rules of operation give rise to complex collective behavior, sophisticated information processing, and adaptation via learning or evolution.

This notion of complex systems could be important for future developments in dataset creation: learning increasingly complex knowledge is important for the development of more powerful models, and the evolution of more complex data is important for models to be able to have more learning opportunities in the first place.

Concretely, we denote an n -samples complexity measure as function $C : \Omega^n \rightarrow \mathbb{R}$, which intuitively measures data “difficulty”. Similarly to quality, complexity can also be defined on a sample level as $C_\Omega \rightarrow \mathbb{R}$

with C being recovered as the average over samples. We now survey commonly used measures of complexity in text data and summarize them in Figure 4.

Basic measures of complexity: We start by reviewing some simple measures of textual data complexity. Token length (Liu et al., 2024b) is broadly applicable across all text but lacks any notions of semantic complexity. Sometimes human generated labels can also be used to form complexity hierarchies as in Hendryck’s MATH which groups problems into increasing levels of difficulty (Hendrycks et al., 2021b). Some other works measure the complexity of data as its compressibility (Pandey, 2024) or intrinsic dimension (Sharma & Kaplan, 2020; Havrilla & Liao, 2024).

Attribute complexity: Similar to attribute based measures of quality and diversity, one can define an attribute function T , which measures for each sample its complexity in this case. One such attribute complexity metric is Tree instruct Zhao et al. (2024), which is defined as the number of nodes in the semantic tree of an instruction. The authors show empirically that for this specific complexity metric, the number of added tree nodes in the prompt, with increasing complexity performance increases. Moreover using less, but more complex data over performs more, but less complex data. In this case, it is shown that the better performance does not come due to more tokens. Another complexity metric falling into this category is InsTag (Lu et al., 2023b), where $T(\omega) = \#$ semantic tags (x) assigned to sample ω . Each set of tags is assigned using a prompted LLM.(Sharma et al., 2024) also propose to measure complexity based on the depth and structure of a parse tree. Other works directly prompt GPT-4 to score the complexity of samples on a 1-5 scale (Chen et al., 2024b). EvolComplexity (Liu et al., 2024b) computes complexity scores by evolving instruction responses in a manner similar to WizardLM (Luo et al., 2023b) and then applying GPT-4 as a judge to assess the complexity of each sequence of evolved samples in-context.

Other model based measures: Model perplexity (Liu et al., 2024a) can also be regarded as a measure of complexity, with higher perplexity samples being more complex. Note however there is some overlap with using perplexity as a measure of data quality. The Instruction Following Difficulty (IFD) score (Li et al., 2024c) applies perplexity to measuring complexity of instruction following samples as

$$IFD(x, y) = \frac{s(x|y)}{s(x)}, \quad (2)$$

where s denotes the cross entropy

$$s(x) = -\frac{1}{N} \sum_{i=1}^N \log P(x_i|x_1, \dots, x_{i-1}; \theta)) \quad (3)$$

and θ denotes the weights of the pretrained base LLM. Albalak et al. (2023a) similarly measure complexity as the information gain of a sample.

Psychometric measures: Psychometric approaches like item response theory (IRT)(Rasch, 1993; DeMars, 2010), traditionally used to assess human cognitive abilities and the difficulty of questions, could be adapted to measure the difficulty or complexity of datasets for LLMs. IRT allows the difficulty of datasets to be captured through parameters in a probabilistic model aimed at explaining model performance by sampling from a diverse population of LLMs as respondents. This method could involve presenting samples from the dataset to a variety of LLMs with different sizes and strengths. Measures such as the perplexity of data points could then be analyzed using IRT (Thissen, 1982) to generate standardized difficulty scores for each sample. Such measures have been used to adaptively evaluate models with fewer examples (Polo et al., 2024; Zhuang et al., 2023), and to compare model responses to a population of humans (He-Yueya et al., 2024) — but could be extended to measure the complexity of the training data. This would provide a nuanced measure tailored specifically to LLMs, which could be valuable for curating datasets of varying difficulty levels for training and testing.

Recap We now summarize the main takeaways from the last three sections covering various measures of quality, diversity, and complexity:

Takeaways

- General purpose textual similarity measures are broadly applicable but often fail to capture domain-specific features of interest.
- Attribute-based measures must be handcrafted for specific domains but can be chosen to capture important features of interest.

These observations are reinforced by findings in Liu et al. (2024b) who compare several different measures of quality, diversity, and complexity in instruction tuning data. They found model-based measures relying on GPT-4 as a judge to outperformed purely lexical quality and complexity measures. This leads to the following question:

Open Questions

- How can we develop better domain agnostic measures capable of adapting to domain specific features of interest?

3 THE EFFECTS OF DATA QDC ON MODEL PERFORMANCE

In Section 2 we defined notions of quality, diversity, and complexity and surveyed numerous practical implementations used in the literature. In this section we now seek to understand the effects of quality, diversity, and complexity in training data on model performance and generalization. Overall, we find that quality tends to most benefit in-distribution generalization, diversity most benefits OOD generalization, and *appropriate levels* of complexity can benefit both. Additionally, we examine often occurring trade-offs between quality and diversity in training data. As a result, this often produces corresponding trade-offs between in-distribution and OOD generalization.

A note on in-distribution vs. OOD generalization Let P be probability measure on task space $(\mathcal{X}, \mathcal{Y})$ and $\mathcal{D}_{\text{train}} = \{(X_i, Y_i)\}_{i=1}^n \sim P$ a set of training data independently identically sampled (**i.i.d**) from P . In this survey, when we say a model \mathbf{M}_θ has good **in-distribution generalization**, we mean that \mathbf{M}_θ trained on $\mathcal{D}_{\text{train}}$ generalizes well to test data $\{(X_j, Y_j)\}_{j=1}^t$ sampled from P . We say \mathbf{M}_θ has good OOD generalization if it generalizes well to a related but new distribution Q on $(\mathcal{X}, \mathcal{Y})$. See Miller et al. (2021) for more discussion on in-distribution versus OOD generalization.

3.1 THE EFFECT OF QUALITY

Data quality on its own has been a huge topic of interest recently for both model pre-training and post-training (Albalak et al., 2024; Zhou et al., 2023a; Chen et al., 2024b; Cao et al., 2024; Sharma et al., 2024). For this reason we break our discussion into two parts: one for pre-training and one for post-training.

Data quality in pre-training SOTA pre-training data pipelines include several iterative rounds of quality filtering (Albalak et al., 2024). The initial round applies of set of heuristic filters aimed at removing

noise coming from malformed webtext (Penedo et al., 2024) and maximizing “educational content” (Yue et al., 2023). Sharma et al. (2024) apply over a dozen quality-based filters to OpenWebText to improve LLM pre-training efficiency. They find pruning 40% of the available tokens improves model benchmark performance. Zhang et al. (2024d); Shao et al. (2024) both propose measuring quality of mathematical texts with an in-context LLM classifier. They see up to two times increase in pre-training efficiency gains on common reasoning tasks (GSM8K, BBH, MATH) when training on the filtered datasets. Xie et al. (2023) samples high-quality data for pre-training via importance resampling with similarity to a target distribution computed using a bag-of-words n-gram model. The Phi series of models (Gunasekar et al., 2023; Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024) explores the improvement from increasing data quality as opposed to scale. Gunasekar et al. (2023) trains a model on high quality data filtered by ChatGPT and achieves benchmark performance comparable to significantly larger models. Subsequent iterations (Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024) rely heavily on high-quality synthetic data, demonstrating impressive benchmark gains on targeted reasoning tasks. Several works (Penedo et al., 2024; Abdin et al., 2024; Dubey et al., 2024a) show training synthetic data to train model quality classifiers works extremely well in filtering out low-quality data.

Data quality in post-training Much recent work also demonstrates the importance of high-quality fine-tuning data (Zhou et al., 2023a; Chen et al., 2024b; Cao et al., 2024). Zhou et al. (2023a) finds fine-tuning even on just 1000 highest quality samples improves instruction following performance. However, they also note lower quality samples has a negative effect unless more diverse samples are also included. Quality is measured using a Likert scale with scores assigned by both humans and GPT. Diversity is measured via topic tagging. Most of their evaluation is done in-distribution. Alpaganus (Chen et al., 2024b) filters the Alpaca instruction tuning dataset from 52k samples to 9k samples via Likert-scale quality annotations from ChatGPT. Fine-tuning on this smaller, higher quality dataset results in a more preferred (by GPT-4) instruction following model. Liu et al. (2024b) propose a number of quality-diversity metrics (EvolQuality/EvolComplexity) which they use to filter instruction-tuning data. They find the Evol quality scores and RM score correlate well with downstream performance.

3.2 THE EFFECT OF DIVERSITY

Of equal importance to the quality of training data is its diversity during both pre-training and post-training. In contrast to quality, post-training approaches often do not see a huge improvement in in-distribution performance with more diverse training data. Instead, diversity most benefits OOD performance.

Data diversity in pre-training Data diversity plays a crucial role in pre-training (Raffel et al., 2020; Brown et al., 2020a; Li et al., 2023b). Many pre-training methods assume that a post-training process will occur, thereby making the goal of pre-training for the model to see as much data as possible. However, simply training on all available data can lead to highly duplicated content, and training inefficiency, motivating significant efforts to improve data deduplication. For example, Wenzek et al. (2020) and Ortiz Suárez et al. (2019) utilize hashing based deduplication methods when developing CCNet and OSCAR, respectively. Additionally, Lee et al. (2022) propose the EXACTSUBSTR algorithm, which detects duplicates based on shared substrings. More recently, model-based deduplication has become popular, with SEMDEDUP (Abbas et al., 2023) and D4 (Tirumala et al., 2023) being common examples. Each of these methods utilize a neural network to embed the data, followed by some clustering and a removal of data within each cluster according to some similarity metric. In particular for pre-training where the downstream goals are not known, data deduplication reduces the risk of models overfitting to anything in particular, and has been shown to sometimes improve accuracy on downstream tasks (Tirumala et al., 2023).

Additionally, a number of works have found that training on more diverse data can lead to improved performance (Lee et al., 2022; Tirumala et al., 2023). For example, Ye et al. (2024a) found that pretraining

on datasets with higher diversity generally improves performance on downstream domains. The Phi model series studies the impact of training data diversity from the perspective of training on synthetically generated data (Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024). They find it challenging to ensure that the model-generated data is truly diverse, and find methods of injecting randomness. For example, Javaheripi et al. (2023) seed the generation prompts for their synthetic data with 20K topics (e.g. science, daily activities) and include filtered web samples in the training data for additional diversity. Ye et al. (2024a) finds diverse mixture weights for data-sources from different topics is optimal pre-training performance. This aligns with Zhang et al. (2024b)'s finding that even imbalanced distributions of diverse data can still drive effective generalization, as long as there is sufficient semantic coverage across domains. Eldan & Li (2023) introduced TinyStories, a diverse synthetic dataset of simple stories. The authors demonstrate that small models trained on this diversity-controlled dataset can generate coherent text and show basic reasoning-capabilities that typically require much larger models on standard corpora. The impact of diversity appears to benefit both specialist and generalist models in distinct ways: For specialist models (e.g., code language models), Zhang et al. (2024b) demonstrate that extending data diversification beyond their core domain yields substantial performance improvements in instruction following (up to a limit) as compared to training solely on domain-specific data. For generalist models, using diverse data mixtures enhances their instruction-following capabilities across a broad range of domains more effectively than simply increasing the quantity of training data. Chen et al. (2024a) studies the impact of diversity in synthetic data on the training of LLMs by proposing a diversity measure pipeline using LLMs to perform clustering of text corpus while generating data. They show that higher synthetic data diversity correlates positively with pre-training and supervised fine-tuning performance.

Data diversity in fine-tuning In contrast to quality, diversity can improve the in-distribution generalization performance, but only up to a certain point (Albalak et al., 2023b). Instead, training dataset diversity appears to be essential for out-of-distribution (**OOD**) generalization (Wang et al., 2022; Yu et al., 2023b; Fan et al., 2023). The Supernatural instructions dataset (Wang et al., 2022) assembled a large instruction dataset comprising over 1600+ instruction types spanning a diverse collection of domains and languages. Their evaluation procedure proceeds by training models on subsets of instructions and testing generalization to held out OOD instruction sets. They found scaling both the number of instruction classes and the model size has a significantly improved test performance. However, increasing the number of instruction demonstrations per instruction type beyond 64 had little to no impact. A similar observation about scaling the number of instruction types versus the number of demonstrations per type was made in (Zhou et al., 2023a).Bukharin & Zhao (2024) design an algorithm that selects the highest quality samples, while making sure selected data is sufficiently diverse, by properly adjusting a parameter trading off data quality and diversity. They find training instruction-following models on more diverse datasets does not improve average-case performance on in-distribution benchmarks (AlpacaEval) but does improve performance on the hardest questions. In the reasoning domain, Yu et al. (2024) found training on more diverse data (as compared to higher quality data with the same level of diversity) improved the scaling laws of models on downstream tasks. Zhang et al. (2024a) experiment with varying levels of instruction diversity when training transformers to solve algorithmic string rewriting tasks. They find, when controlling for sample count, a more diverse task set with few samples per task generalizes to new tasks better than more samples concentrated on less-diverse tasks. (Wei et al., 2024) observed that instruction tuning increases sycophancy, a form of reward hacking where the model tends to repeat the user's opinion even if it is objectively incorrect. To mitigate and reduce this undesired behavior, they've used simple classification datasets and augmented them with diverse opinions to teach the model that a statement's truthfulness is independent of a user's opinion.

Notably, in some cases diversity inversely correlates with better performance (Ye et al., 2022b; Pourcel et al., 2024; Bukharin & Zhao, 2024; Zhang et al., 2024a).Zhang et al. (2024a) notes instruction imbalance can be detrimental but addressed via more diversity. For example, Yue et al. (2024) find both kNN and MTLD measures of instruction diversity are not good predictors of generalization when evaluated on in-distribution chat tasks. FLAN trains models on a diverse set of instructions and sees great OOD generalization to new

instructions (Wei et al., 2022). Dong et al. (2024) ablates fine-tuning on different domain ratios and finds there is some positive transfer at low-data amounts but limited transfer at higher amounts. They suggest diverse data is most helpful when high-quality in-distribution data is not available.

3.3 THE EFFECT OF COMPLEXITY

Finally, we consider the effect of data complexity on model generalization during both pre-training and post-training. Unlike quality and diversity, complexity appears to often benefit both in-distribution and OOD generalization. However, this is only true to a certain point, with excessively complex data instead harming generalization. For example, recent work by Kallini et al. (2024) demonstrate that while language models struggle with completely random patterns, they can learn unnatural but structured patterns, albeit less efficiently than natural ones. Hence an effective training data should maintain learnable structure while introducing sufficient challenge.

Data complexity in pre-training Complexity measures can be used to filter pre-training datasets. Albalak et al. (2023a) propose an approach that adapts data-mixing proportions in an online fashion by training a multi-armed bandit algorithm on domain perplexity. The higher the perplexity, the more complex the data is for the model and the larger the information gained by learning from it. This approach achieves lower text perplexity compared to DoReMi (Xie et al., 2024) and an unweighted baseline. However, it requires a relatively clean dataset to ensure that high perplexity is not correlated with low-quality data. For instance, Wenzek et al. (2019) train a language model on Wikipedia data to compute perplexity as a measure of distance from high-quality text, removing documents that deviate too much.

Alternative approaches focus on gradually increasing complexity during training. Dubey et al. (2024b) show that using an annealing phase — where the learning rate is lowered to zero while incorporating more complex domain data like code and mathematics — enhances performance on key benchmarks. Others (Pandey, 2024; Sharma & Kaplan, 2020; Havrilla & Liao, 2024) investigate how data complexity impacts pre-training, using measures like intrinsic dimension or gzip compression. They demonstrate that more complex data takes longer to learn.

Data complexity in fine-tuning Zhao et al. (2024) measures complexity of instruction following datasets by parsing instructions into a semantic tree. They find fine-tuning on more difficult instructions outperforms fine-tuning on more less-difficult instructions where the number of training tokens is equalized. INSTAG Lu et al. (2023b) measures complexity as the number of attribute tags assigned to a sample by a classifier LLM. They find models fine-tuned on more complex instruction samples perform better on MT-Bench. The WizardLM model series (Xu et al., 2023; Luo et al., 2023b;a; Zeng et al., 2024) employ LLMs to automatically evolve the complexity of seed data. They see significant improvement in models fine-tuned on the resulting synthetic data. Li et al. (2024c) measures the complexity of instruction following data using a novel Instruction Following Difficulty metric based on the ratio of the instruction’s perplexity given the response to instruction’s unconditional perplexity. They also find higher complexity datasets yield better instruction following results. Liu et al. (2024b) measures complexity of chat data by adopting similar evolutionary approach to WizardLM. They conduct a review of several baseline complexity measures, finding their best performing data split on MT-Bench also has high complexity as measured by their proposed metric. Similarly, Zhao et al. (2024) measure complexity of instruction data by decomposing samples into a tree structure and counting the size of the tree. Controlling for token count, they find training on fewer, complex instructions outperforms training on more, simple instructions. However, it is also true that training on excessively difficult data can harm model generalization. Recent work on weak-to-strong generalization (Sun et al., 2024) demonstrates training on lower-complexity math problems can generalize as well as, and sometimes even better than, training directly on more complex data.

Recap We'll now review the main takeaways from the last three sections covering the effects of quality, diversity, and complexity on model generalization:

Takeaways

- High-quality data improves in-distribution generalization more than OOD generalization.
- Highly diverse data improves OOD generalization more than in-distribution generalization.
- Complex data, at the right levels of difficulty, can improve both in-distribution and OOD generalization.

These takeaways summarize the qualitative impacts of QDC on model generalization. Further work in this direction could attempt to quantify these impacts in the form of *quality-diversity-complexity* scaling laws on downstream tasks. Existing scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022) can typically be used to predict pre-training loss $L(N, D, C)$ as function of model size N and data size D and compute size C . Typically lower pre-training loss also correlates with better benchmark performance. Recently accurate quantitative predictions of model benchmark capabilities as a function of size, data, and pre-training loss have been made (Ruan et al., 2024; OpenAI et al., 2024). These predictions of may be improved further by considering the quality, diversity, and complexity of training data. Already there exist several works demonstrating the complexity of pre-training data directly influences scaling behavior (Sharma & Kaplan, 2020; Havrilla & Liao, 2024; Pandey, 2024). Additionally, many questions remain surrounding maximal levels of data complexity useful for improving performance. If samples are too complex then they will likely instead harm generalization.

Open Questions

- Can we develop models predicting how down-stream metrics of interest will behave as a function of training data quality, diversity, and complexity?
- How can we determine the **right level** of complexity to most benefit model generalization.

3.4 TRADE-OFFS IN QDC AND IMPACTS ON PERFORMANCE

In order to reap the benefits of quality, diversity, and complexity, the ideal training dataset should simultaneously be high-quality, highly-diverse, and highly-complex. However, in practice this is not always possible due to data availability and natural trade-offs between quality and diversity. For example, this happens often during data filtering, as more aggressive filtering can improve quality but limit the amount and diversity of data (Longpre et al., 2023). As a result, higher-quality datasets tend to be less diverse and vice-versa. Fan et al. (2023) study this phenomena in the image domain by prompting text-to-image models to generate large sets of synthetic data to train image classifiers for the Imagenet classification task (Russakovsky et al., 2015). Quality is measured using a pre-trained image classifier and diversity is measured by extracting intermediate embeddings from the same pre-trained model and measuring standard deviation of the embeddings across all classes. They find synthetically generated Imagenet datasets with higher-quality tend to be less diverse. The models trained on these high-quality, low-diversity datasets achieve good in-distribution test performance. However, they struggle when generalizing out of distribution. Conversely, they find synthetically generated image datasets with high-diversity are often lower-quality. However, models trained on more diverse datasets generalize better to OOD benchmarks (MS-COCO). This demonstrates a trade-off between quality

and diversity, **with higher-quality data benefitting in-distribution generalization and higher-diversity data benefitting OOD generalization**. Hashimoto et al. (2019) and Holtzman et al. (2020) make similar observations about trade-offs between quality and diversity in text data.

Bukharin & Zhao (2024) conducts a similar study using quality and diversity metrics to filter publicly available instruction tuning datasets. To measure quality they use both a contrastively trained reward model and ChatGPT as a judge. To measure diversity of a training subset $S' \subseteq S$, they embed instruction-response pairs using Sentence Transformer (Reimers & Gurevych, 2019) and compute the *facility location function* of S' with respect to S . The quality and diversity measures are then used to construct an objective function $o(s) = \alpha q(s) + (1 - \alpha)d(s)$ used to greedily select instruction samples with $0 \leq \alpha \leq 1$. When the number of samples is fixed, for low values of α the selected datasets is highly diverse but lower quality. For high values of α the resulting dataset is high-quality but lacks diversity. Further, the instruction models trained on high-quality, low-diversity datasets tend to perform well over-all on both AlpacaEval (Zheng et al., 2023). Models trained on highly-diverse but lower-quality datasets perform worse overall. However, they generalize better than models trained only on high-quality samples to the most difficult benchmark instructions. Liu et al. (2024b) report similar findings for chat data, in the process comparing multiple measures of data quality, diversity, and complexity. Alpaganus (Chen et al., 2024b) also notes that filtering out lower quality samples simultaneously simultaneously limits diversity. As a result, coding quality of the fine-tuned model suffers.

Takeaways

- Levels of quality and diversity naturally trade-off in data, affecting model generalization.

Virtually no works have investigated the trade-offs between data **complexity** and quality and diversity. Understanding this relationship is essential for predicting the performance of models trained on data with varying mixtures of quality, diversity, and complexity.

Open Questions

- What is the relationship between **complexity** and quality and diversity in natural data?

4 QDC PROMOTING MECHANISMS IN SYNTHETIC DATA ALGORITHMS

In Section 2 we defined notions of quality, diversity, and complexity and surveyed numerous implementations of each measure in the literature. In Section 3 we discussed how the quality, diversity, and complexity of a training dataset \mathcal{D} impacts the generalization of models to downstream tasks. Now, in Section 4, we will discuss existing methods for synthetic data generation through the lens of quality, diversity, and complexity. We do this by identifying and categorizing mechanisms for promoting the quality, diversity, and complexity of synthetic data created by synthetic data generation algorithms. This allows us to start categorizing synthetic data generation algorithms themselves as being “quality-seeking”, “diversity-seeking”, “complexity-seeking”, or some mixture thereof. As in Section 3, we also discuss the interplay between quality, diversity, and complexity promoting mechanisms in synthetic data generation algorithms and the resulting effects on downstream model capability. Of particular interest to us is **the impact of quality, diversity, and complexity in synthetic data when used for recursive model self-improvement**: either through reinforcement learning or other means.

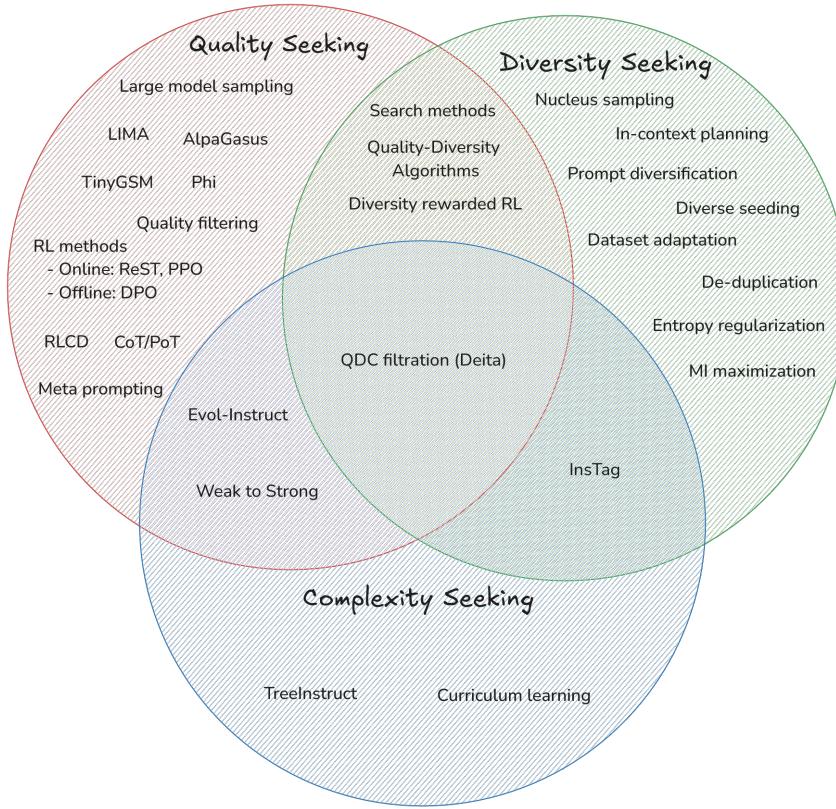


Figure 5: Venn diagram of quality, diversity, and complexity seeking algorithms.

Most existing methods for synthetic data pipelines can be broken into three iterable phases:

1. Data generation
2. Data filtration
3. Data distillation

The dataset generation phase proceeds by sampling many responses to input prompts in training data, yielding synthetic dataset \mathcal{D}_0 . During dataset filtration, the generated samples are filtered using a chosen objective function $O : \Omega \rightarrow \mathbb{R}$, yielding $\mathcal{D}'_0 \subseteq \mathcal{D}_0$. Finally, during dataset distillation the remaining desired synthetic data after filtration is distilled into a student model M , i.e., M is trained on \mathcal{D}'_0 to improve its performance. Note that M can be either the same model used for data generation or an entirely different model. We find the majority of approaches rely on a large, SOTA LLM to maximize answer quality during generation. The most common approach to maintaining diversity is through the use of a large seed-dataset of input prompts. Even this simple combination can be effective without any filtering provided the SOTA LLM consistently produces enough high-quality solutions. However, more sophisticated algorithms can be used to encourage better sample quality, diversity, and complexity during the entire process. Various sampling algorithms can be applied to improve the QDC of generated data. Complex filters can be applied to control and trade-off levels of QDC. Training procedures beyond the standard next token prediction loss can be used to improve model output quality, diversity, and complexity. Toward this end we highlight a small set of algorithms,

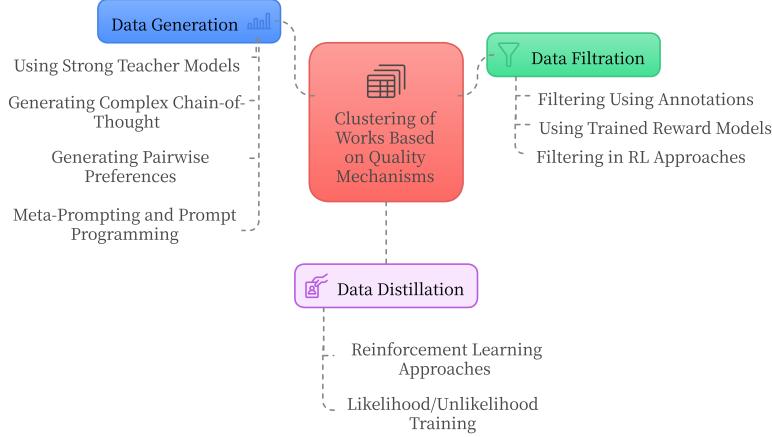


Figure 6: Mechanisms for promoting data quality in synthetic data generation.

largely inspired by techniques from QD literature (Mouret & Clune, 2015), which encourage sampling directly maximizing both quality and diversity. Finally, we discuss the impact of the quality, diversity, and complexity on synthetic data generation capability itself resulting effects on recursive self-improvement algorithms. We highlight an apparent trade-off between model output quality and model output diversity and the importance of balancing both for optimal improvement. Further, we note that most existing models are heavily optimized for answer quality (via maximizing single response pass@1 scores) and discuss how this comes at the expense of better output diversity in synthetic data. We finish with a discussion on how to sample and train models for simultaneously improving output quality, diversity and complexity. Note: all the mechanisms we discuss are collected in a table in Appendix Section B.

4.1 QUALITY PROMOTING MECHANISMS

1. Quality Mechanisms in Data Generation One simple yet effective approach generates data using large SOTA LLMs (e.g. GPT-4), which guarantees a certain level of quality relative to weaker, less capable models (Mukherjee et al., 2023; Gunasekar et al., 2023; Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024; Chiang et al., 2023; OpenAI et al., 2024; Wang et al., 2022; Honovich et al., 2022; Dubey et al., 2024b). For example, Unnatural Instructions (Honovich et al., 2022) use 3 examples from Supernatural instructions and ask the pretrained model to generate a fourth. they repeat this process with 5 different seeds, generating 15 instruction examples to automatically produce 64k diverse triplets of instructions, inputs and outputs. This is an example of *prompt generation* which generates both the an input prompt the LLM and the expected response. Distillation into the weaker student model can then be done via supervised fine-tuning, sometimes even with minimal data filtering. Alpaca, Vicuna (Chiang et al., 2023), and many derivatives take this approach by fine-tuning LLaMa (Touvron et al., 2023a) on continuations generated with InstructGPT (Ouyang et al., 2022), ChatGPT and GPT-4. These are examples of *response generation* where input prompts are sourced beforehand and only synthetic responses are generated. The Phi model series (Gunasekar et al., 2023; Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024) and Orca (Mukherjee et al., 2023) generate complex chain of thought reasoning traces across a variety of domains using GPT-4. For example, from Mukherjee et al. (2023), “Orca learns from rich signals from GPT-4 including explanation traces; step-by-step thought processes; and other complex instructions, guided by teacher assistance from ChatGPT.” Many papers have also applied the same recipe to generate synthetic data for math and reasoning problems

(Yu et al., 2024; Liu et al., 2023a; Yue et al., 2023; Toshniwal et al., 2024). TinyGSM (Liu et al., 2023a) generates millions of GSM8K (Cobbe et al., 2021) style questions using GPT-4 to fine-tune a small generator and verifier. Even with minimal filtering only done on questions which have bad code syntax, the resulting models achieve nearly 70% pass@1 accuracy on GSM8K. Using a verifier to rerank multiple solutions improves this by 12%. Kim et al. (2023) generate a dataset of pairwise preferences by prompting multiple models with varying parameter counts, quantities of in-context demonstrations, and qualities of in-context demonstrations. They make the assumption that larger models with higher quality and quantity of in-context examples *should* generally be preferred to smaller models with fewer and lower quality demonstrations.

Chain of thought prompting (Wei et al., 2023) and all its derivatives (Chen et al., 2023c) can be used to generate informative, higher-quality step by step responses to prompts. This type output is used in nearly all data generation schemes. Reynolds & McDonell (2021) employ meta-prompting to generate better LLM prompts, improving quality on downstream tasks. Reinforcement learning from contrastive distillation (Yang et al., 2024b) independently prompts an LLM to generate a “good” sample and a “bad” sample for a task and then trains a reward model using this synthetically generated data. Contrastive decoding (Li et al., 2023a) samples text that maximizes the difference between expert log-probabilities and amateur log-probabilities, subject to plausibility constraints which restrict the search space to tokens with sufficiently high probability under the expert LM. Another method for improving the quality of generated data is by training a critique model to generate natural language feedback on a model’s response, which can then be followed by a refinement stage to improve quality (Wang et al., 2023b). West-of-N sampling (Pace et al., 2024) draws inspiration from best-of-N sampling and defines the chosen and rejected continuations to be the best and worst N samples. One downside of this method is that it requires an initial reward model to make the initial ranking judgments.

2. Quality Mechanisms in Data Filtration Aside from using a strong teacher model, the primary way quality is maintained during the synthetic data pipeline is through aggressive filtering with various quality metrics discussed in Section 2.1. Many works (Zhou et al., 2023a; Chen et al., 2024b; Cao et al., 2024) demonstrate quality filtering synthetic data generated by strong base models can further improve performance of the student. Some filtering can be done at the *syntactic level* to ensure generated outputs follow an expected structure. For example, Honovich et al. (2022) filter out all samples not in prompt response format. More complex *semantic filtering* can also be done assessing the content of generate samples. Several popular and effective semantic measures of quality can be used including data filtered via human ratings, contrastive reward models, outcome based reward models, or even process based reward models (see Sec 2.1 for more quality measures). AlpagaSUS (Chen et al., 2024b) filters the synthetically generated Alpaca instruction tuning dataset from 52k samples to 9k samples via Likert-scale quality annotations from ChatGPT, with a threshold τ of 4.5, manually selected by looking at the histogram of the scores. Bukharin & Zhao (2024) filters common synthetically generated chat datasets for quality using a trained contrastive reward model. SPIN (Chen et al., 2024c) iteratively trains an instruction tuned model by generating synthetic data and prompting the model to distinguish between synthetic and human generated responses.

Quality in Reinforcement Learning for LLMs Additionally, nearly all RL-based approaches for LLM fine-tuning promote quality via their choice of reward function used. Approaches including ReST (Gulcehre et al., 2023), ReSTEM (Singh et al., 2024), Expert Iteration (Havrilla et al., 2024a;b), and Reward Ranked fine-tuning (Dong et al., 2023; Yuan et al., 2023) assume access to a fixed set of prompts and generate data by conditioning an initial LLM to generate responses on this prompt set. On training data with a ground truth reference often exact match on the final answer. When this is not feasible, again reward models can be trained to judge correctness as is commonly done during RLHF (Ziegler et al., 2020; Bai et al., 2022a). For example, ReST (Gulcehre et al., 2023), which trains an LLM to do translation, alternates between a “Grow” step, which samples many candidate translations on a set English paragraphs and filters them using a pre-

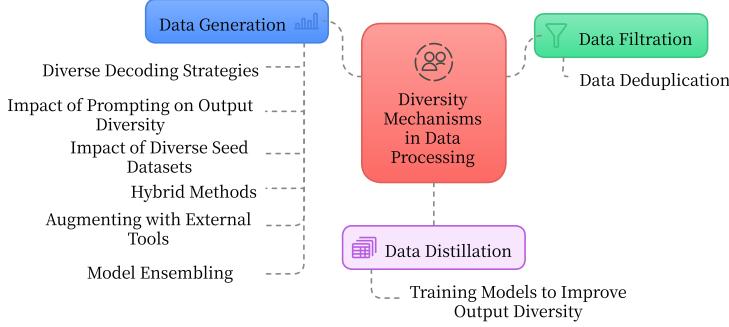


Figure 7: Mechanisms for promoting data diversity in synthetic data.

trained reward model, and an “Improve” step which fine-tunes the generator LLM on the filtered data. They use reward models to judge the quality of the translation and filter out samples below a certain threshold.

3. Quality Mechanisms in Data Distillation More sophisticated RL algorithms influence model output quality both by rewarding high-quality samples and penalizing low-quality samples (Rafailov et al., 2024; Pang et al., 2024; Schulman et al., 2017; Ahmadian et al., 2024; Setlur et al., 2024) thus further improving model output quality. DPO (Rafailov et al., 2024) formulates a contrastive objective between the logprobs of high-quality samples and low-quality samples. Iterative DPO (Pang et al., 2024; Setlur et al., 2024) iteratively samples synthetic data and trains a policy using the DPO objective. On-policy policy-gradient algorithms such as PPO (Schulman et al., 2017) and REINFORCE (Williams, 1992; Ahmadian et al., 2024) can be used to directly optimize for the given reward. In addition to RLHF training, both PPO and REINFORCE have been successfully applied to improving LLM reasoning capability (Wang et al., 2024d; Havrilla et al., 2024a; Sun et al., 2024; Le et al., 2022). Havrilla et al. (2024a) finds training with outcome based reward models trained using data from the SFT policy improves RL sample efficiency. (Wang et al., 2024d; Sun et al., 2024) both find RL training using dense process based rewards can lead to performance improvements over sparse terminal ground truth or model based outcome rewards. Also related is unlikelihood training (Welleck et al., 2020) which combines the standard log-likelihood objective with an “unlikelihood” objective penalizing undesirable tokens at each step.

4.2 DIVERSITY PROMOTING MECHANISMS

Of equal importance to promoting quality in synthetic data is the promotion of diversity. Here we survey the impact of components of synthetic data generation pipelines on data diversity.

1. Diversity in Data Generation In contrast to most quality promoting mechanisms, the majority of diversity promoting mechanisms take effect during the initial data generation phase. We group these into several categories of techniques for improving diversity.

Diverse Decoding Strategies Changing the sampling temperature is one of the simplest ways to change model output diversity (Viswanathan et al., 2023). Havrilla et al. (2024a) experimented with different sampling temperatures and rollout prompts when training using PPO. They found the best temperature depended on the initialization of the model being trained, with higher temperatures useful for SFT initialized models and lower-temperatures useful for prompted pre-trained models. Ye et al. (2022b) generate task specific data by prompting large pre-trained models. They find varying top-k, top-p and nucleus sampling (Holtzman et al., 2020) can also improve output diversity where diversity is measured by Self-Bleu score (Welleck

et al., 2020; Holtzman et al., 2020). Nucleus sampling (Holtzman et al., 2020) is designed to reduce token repetition and promote output diversity while maintaining fidelity and coherence of the generated text. Nucleus sampling proceeds at each step by truncating the token distribution at the first k tokens with cumulative probability above some chosen threshold $0 < p \leq 1$ and re-normalizing.

Impact of prompting on output diversity: Varying LLM prompts is another simple yet highly effective method of introducing more diversity into LLM outputs. Jiang et al. (2024a) prompt GPT to generate diverse plans in preparation to solve coding problems. Naik et al. (2024) try several prompting strategies to improve math problem solution output diversity including prompting with multiple personas (also explored in (Chan et al., 2024)), working backwards, and method of elimination. Toshniwal et al. (2024) prompt for solutions to problems based on the desired subject/skills. Yu et al. (2023b) tries varying prompt length and prompt style. Ye et al. (2022b) adopts task specific prompts depending on the domain. Bradley et al. (2023) and Fernando et al. (2023a) show that maintaining and selecting for a continually changing in-context examples pool in an evolutionary fashion is important for overcoming limited output diversity from (fine-tuned) LLMs, leading to improved diversity in responses from LLMs, while maintaining/improving output quality in code generation and task prompt generation domains respectively. Naik et al. (2024) devise a multi-stage process of creating diverse prompts by first identifying appropriate approaches for solving a problem, then identifying relevant personas to imitate and finally using the cross-product of approaches and personas to generate a set of diverse candidate prompts. Similarly, Yu et al. (2023b) generate diverse prompts by explicitly specifying the desired attributes (e.g. length and style). Yang et al. (2024b) create diversity across their chosen and rejected continuations by employing contrasting prompts. They use a positive prompt designed to encourage adhering to the given principles for the chosen continuation, and a negative prompt designed to encourage violations of the same principles. Liu et al. (2023b) utilizes an LLM to act as both a data generator and critic which assesses sample correctness and novelty. Sample generation is done by mutating an existing problem with respect to a set of mutation “principles”. The novelty of the new sample is then assessed relative to its parent.

Impact of diverse seed datasets All synthetic data generation algorithms start from a set of initial seed prompts from which data is generated. In principle, algorithms successfully bootstrapping from a limited set of prompts are more desirable, as they are in some sense generating truly synthetic data. In practice however, the size of the set of seed prompts, and the manner in which they are transformed, dramatically affects diversity. We call algorithms which do not rely on a large set of seed prompts *generation* algorithms. We call algorithms which do leverage large sets of seed prompts *adaptation* algorithms. These two categories are mainly differentiated by the scale of the seed dataset.

Self-instruct (Wang et al., 2023c) is an example of a dataset generation algorithm which generates an instruction following dataset from less than 500 seed prompts. These examples are used to prompt the pre-trained LLM to produce new instructions. Any low-quality or overly similar responses are then removed, and the remaining instructions are reintegrated into the task pool for further iterations. Toshniwal et al. (2024) presents a hybrid method which generates solutions to math problems in natural language and then converts them to a representation in code. Metamath (Yu et al., 2024) prompts GPT-4 to rephrase existing seed questions in addition to generating new solutions. In contrast to dataset generation algorithms, dataset adaptation algorithms typically generate data using much larger seed datasets. Because the seed-datasets are so large, often the focus is on annotating or transforming instead of generating entirely new samples. Toolformer (Schick et al., 2023) uses gpt-j (Wang & Komatsuzaki, 2021) to annotate common-crawl with tool calls lowering the perplexity of subsequent tokens. The annotated samples are then used to construct a synthetic fine-tuning dataset. prompt2model (Viswanathan et al., 2023) and DataTune (Gandhi et al., 2024b) are given a target task and retrieve relevant publicly available fine-tuning datasets, using an LLM to adapt the retrieved data to the format of the given task. Several papers (Viswanathan et al., 2023; Ding et al., 2023; Huang et al., 2022) use LLMs to generate labels for unlabeled datasets using self-consistency techniques. Lanchantin et al. (2023) annotates a pre-training dataset with self-notes improving the model’s downstream predictions. Yue et al. (2024) scrapes “educational” websites for STEM instruction style data, filtering out high-quality

samples using a FastText classifier and GPT-4. Rephrasing is then done using Qwen-72B with CoT reasoning added in when necessary. This broadly improves benchmark performance across the board. Gandhi et al. (2024b) proposes a recipe for task specific dataset transformation. The method generates new data from existing one with the aim of boosting performance for a specific task. The newly generated data can be used for pre-training. Agarwal et al. (2021) synthetically generate a knowledge-enhanced corpus for language model training by verbalizing a Wikipedia knowledge graph.

Many algorithms exist which exhibit characteristics of both dataset generation and dataset adaptation algorithms. Cosmopedia (Ben Allal et al., 2024) is a hybrid method which generates a new pre-training dataset by sampling continuations from strategically selected prompts in an existing corpus. Data Advisor (Wang et al., 2024b) proposes a systematic framework where one LLM acts as an advisor to monitor dataset coverage, identify underrepresented aspects, and guide another LLM’s generation of samples to fill these gaps. Backtranslation methods in machine translation (Sennrich et al., 2016) and Instruction backtranslation (Li et al., 2024d) both operate on large amounts of unstructured prompt data and generate large amounts of synthetic data by backtranslating to the target domain.

Augmenting with external tools: Integrating LLMs with tools can also be used to improve model output diversity. Ni et al. (2024) annotates program repair tasks with execution traces and generates zero-shot critiques and refinements (improves output quality and diversity). Gou et al. (2024); Lozhkov et al. (2024) leverages a python interpreter to engage in more diverse self-corrections.

Search Combining LLMs with exterior search algorithms can also be used to improve model output diversity. The most naive example of this is i.i.d sampling (Cobbe et al., 2021). Tree of thoughts (Yao et al., 2023) proposed combining generative LLMs with breadth first search and an LLM for state evaluation to improve state diversity in Game of 24, creative writing and Crosswords. MCTS (Tian et al., 2024) has also been used to in an attempt to find more diverse solutions. Stream of Search (Gandhi et al., 2024a) demonstrates the search process can also be performed fully in-context. Self-reflection techniques (Shinn et al., 2023; Madaan et al., 2023) can also be seen as a type of in-context search process. Havrilla et al. (2024b); Qu et al. (2024); Snell et al. (2024); Kumar et al. (2024) generates refinement data synthetically for reasoning tasks, demonstrating sampling more refinements can achieve better diversity than i.i.d generation. Miao et al. (2023) uses LLMs to self-check solutions as process based reasoner and improve multi-sample solution accuracy with weighted voting. Also related is the use of LLM generated plans (Jiang et al., 2024a; Wang et al., 2024a) in natural language which can improve the diversity of generated code.

Several recent papers (Li et al., 2024a; Brown et al., 2024; Snell et al., 2024; Wu et al., 2024b) compare the performance of various inference time strategies (parallel i.i.d sampling, Monte Carlo Tree Search, RE-BALANCE (Wu et al., 2024b), Self-reflection) when sampling large vs. small models with a fixed compute budget. Li et al. (2024a) shows that Llama-2-7B (Touvron et al., 2023b) can obtain 72% accuracy on MATH when i.i.d sampled 256 times despite achieving less than 20% accuracy when sampled once. Brown et al. (2024) finds that repeated i.i.d sampling scales log-linearly across many benchmarks (GSM8K, MATH, MiniF2f (Zheng et al., 2022)). Additionally they find sampling smaller, cheaper models can outperform sampling larger, expensive models at the same cost budget. However, the benefit of smaller models greatly depends on problem complexity, with harder problems being less amenable to efficient sampling with smaller models. Snell et al. (2024); Wu et al. (2024b) make similar observations, additionally comparing i.i.d sampling to more sophisticated search strategies leading to improvements in output diversity as measured by pass@n performance.

Model Ensembling Yuan et al. (2023) shows that sampling multiple models results in more diverse samples than a sampling from a single model. Using multiple training checkpoints of a single LLM can also improve diversity (Liu et al., 2023a). Kim et al. (2023) generate a dataset of pairwise preferences by prompting multiple models with varying parameter counts, quantities of in-context demonstrations, and qualities of in-context demonstrations. They make the assumption that larger models with higher quality and quan-

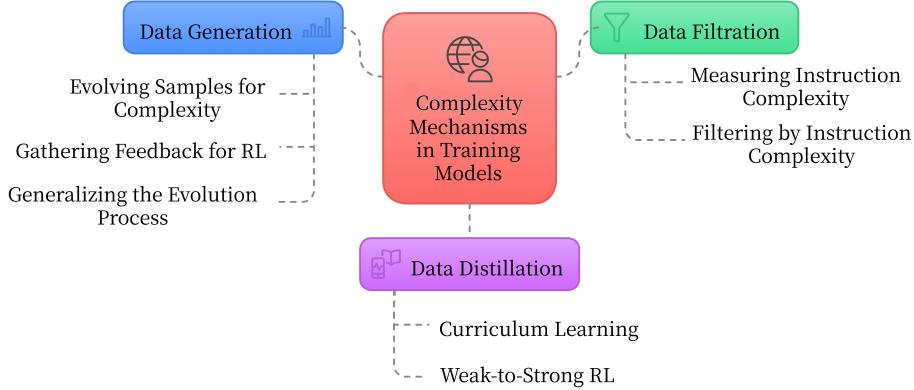


Figure 8: Mechanisms for promoting data complexity in synthetic data generation.

tity of in-context examples *should* generally be preferred to smaller models with fewer and lower quality demonstrations.

2. Diversity Mechanisms in Data Filtration As discussed in Section 3, there are many notions of data diversity that can be used to filter training data. Most prominent are various forms of deduplication (Wenzek et al., 2020; Abbas et al., 2023) which de-bias pre-training data away from frequently occurring strings of text. Synthetic solutions to math word problems can be deduplicated by their order of operations (Yu et al., 2023a; Havrilla et al., 2024a) resulting in improved downstream performance. Tian et al. (2024) measures semantic similarity of reasoning steps using heuristics and LLM-as-a-judge to combine semantic similar actions. Lu et al. (2023b) measures the diversity of synthetic chat data by tagging it with topics/skills using prompted LLM classifiers. Dataset diversity can then be measured as the number/overlap of unique tags in a dataset.

3. Diversity Mechanisms in Data Distillation While the vast majority of methods for improving model output diversity are done at inference time, some work exists attempting to train models for improved output diversity (Zhang et al., 2024e; Li et al., 2024f; 2016b; Havrilla et al., 2024a). Li et al. (2016b) introduces a mutual information based objective that improves dialogue diversity as measured with Bleu score. Zhang et al. (2024e) trains a language model to generate factual data by directly optimizing for the frequency of each fact in a given context. They use this to generate a dataset of synthetic biographies which is measured to be more diverse when counting the number of unique biographical combinations. Li et al. (2024f) combines the standard log-likelihood loss with a regularization term maximizing response entropy. Resulting models attain higher pass@n scores when fine-tuned on reasoning benchmarks. Havrilla et al. (2024a); Singh et al. (2024) finds that simply training on synthetic reasoning solutions with diverse solution paths outperforms the pass@n score of a SFT baseline.

4.3 COMPLEXITY PROMOTING MECHANISMS

Finally, we come to the impact of synthetic data generation components on data complexity.

1. Complexity Mechanisms in Data Generation Perhaps one of the best examples of generating increasingly complex data using LLMs is given by the WizardLM series (Xu et al., 2023; Luo et al., 2023b;a; Zeng et al., 2024). Xu et al. (2023) prompts GPT-4 (OpenAI et al., 2024) to “evolve” samples from seed

instruction datasets towards both increasingly complex and increasingly simple solutions. Follow up works (Luo et al., 2023b;a) apply similar ideas to generate large amounts of synthetic instruction following data in code and math domains respectively. In addition, Luo et al. (2023a) gathers feedback from GPT-4 to train correctness and instructiveness reward models used for RL on the generated data. Zeng et al. (2024) further generalizes the evolution process, evolving not just the samples but also the prompts used to guide the evolution of sample complexity.

2. Complexity Mechanisms in Data Filtration As covered in Section 3, there are many competing measures of complexity that can be applied to filter data. Tree instruct (Zhao et al., 2024) measures the complexity of instruction following samples by decomposing instructions into a semantic tree structure. Complexity can then be measured by counting the number of nodes in the tree and filtering out data below a certain number of nodes. InsTag (Lu et al., 2023b) can also be used to measure instruction complexity by counting the number of tags describing an instruction sample. Liu et al. (2024b) proposes a new measure of complexity using GPT-4 to judge multiple samples in-context at the same time. They demonstrate filtering instruction following data with this measure outperforms filtration with alternative complexity measures.

3. Complexity Mechanisms in Data Distillation Curriculum learning training approaches order training data/environments in increasing levels of complexity in the hope this will allow models to generalize to more complex tasks. For example, Graves et al. (2017), Mindermann et al. (2022), Albalak et al. (2023a), Fan & Jaggi (2023), and Jiang et al. (2024b) all apply a complexity-based curricula to language model pre-training. The underlying measure of complexity for all these methods is a variant of entropy, as measured by the language model itself. Curriculum learning has also been used during post-training. Unlike the pre-training curricula, Lee et al. (2024) use a curriculum that considers the difficulty of the subject matter (e.g. high school vs. college-level questions) for instruction tuning. Additionally, Chen et al. (2023b) extract an ordered set of “skills” from a dataset (either instruction tuning or continued pre-training), and develop a learning curriculum based on the ordering of skill difficulty.

Weak to strong (Burns et al., 2023) is a recent LLM alignment approach aiming to supervise strong models through the use of already aligned weaker models. Sun et al. (2024) apply weak to strong ideas to training process based reward models on MATH problems. They find training PRMs on easier questions sometimes allows models to generalize better than training models on all available data. This provides more evidence that the appropriate level of complexity in training depends heavily on the capability of the model being trained.

Recap The past three sections have explored the impact of various components in synthetic data generation pipelines on data quality, diversity, and complexity. Quality, diversity, or complexity promoting components can be found in all three stages of the pipeline ranging from data generation to filtration to distillation.

Takeaways

- Synthetic data quality, diversity, and complexity can be controlled during any of the three stages of synthetic data pipelines.

Given the existence of quality-diversity trade-offs discussed in Section 3.4 we can ask similar questions about the effect of quality promoting components on synthetic data diversity levels and vice versa. For example, how will the QD composition of an algorithm sampling only for quality then filtering via deduplication for diversity compare to the QD composition of an algorithm sampling data for diversity and then filtering for quality? How will the a distillation algorithm designed to improve model output quality affect model

output diversity? Answers to these questions will help us design more *efficient* synthetic data generation algorithms capable of generating a new dataset \mathcal{D} with a target number of samples and QDC composition. More efficient algorithms may require less compute or external resources (e.g. the size of a seed dataset).

Open Questions

- Quantitatively, what effect do various component choices have on QDC trade-offs in the resulting synthetic data?
- How *efficient* is one algorithm versus another in terms of the amount of compute expended to generate a target number of samples with desired QDC composition?

4.4 IMPACTS ON RECURSIVE SELF-IMPROVEMENT

QDC trade-offs in the model output distribution Up until now, we have primarily considered synthetic data generation as a static, single iteration process: we sample data from a single frozen model T_θ , filter it in some way, and then distill into another model S_θ . Many papers (including those employing online RL for LLM training) iterate this pipeline multiple times by using the same model M as both the generator T_θ and the student S_θ . This forms the basis of iterative self-improvement. It is a significant departure from the single iteration regime as now QDC composition of training data from the previous round will directly affect the QDC composition of synthetically generated data in the next round. From the model’s perspective, the model must attain good performance (i.e., output quality) while also exhibiting good exploration (i.e., output diversity) and complexity to drive gains in future rounds. In other words, the distilled model at each iteration must be capable of generating synthetic data sets with sufficient QDC to drive future self-improvement forward. Inspired by our discussion on the trade-off between quality and diversity in static datasets, the following two questions emerge:

- **Q1:** *How does the QDC composition of training data affect the quality, diversity, and complexity of the model output distribution?*
- **Q2:** *Are there trade-offs between model output quality, output diversity, and output complexity?*

In a partial answer to **Q2**, indeed it seems to be the case that some trade-off between model output quality and model output diversity exists. For example, standard RLHF practices (Bai et al., 2022a) heavily optimize models for answer quality with little to no components promoting answer diversity. It is often anecdotally observed RLHF models suffer from a lack of diversity when compared to their base model and even SFT counterparts. Kirk et al. (2024) carefully investigates these claims, finding RLHF models generalize better than SFT models but have worse sample diversity. Korbak et al. (2022) shows that RL for LLM fine-tuning with no KL penalty minimizes a reverse KL objective with the target distribution defined by the reward function. As a result, the trained model tends to collapse to a few high-reward samples. However, adding in the KL penalty regularizes this behavior, preventing collapse (and therefore potentially preserving output diversity if the reference policy is diverse). However, such a penalty can also prevent improvement to output diversity/exploration capability, as the student policy may not diverge too far from the reference (Yang et al., 2024a).

In the reasoning setting, Cobbe et al. (2021) plot the pass@1 versus pass@n performance of SFT models trained on GSM8K math word problems over several epochs. They show pass@1 and pass@100 scores both sharply increase early on in training. However, pass@n performance quickly peaks and starts decreasing while pass@1 performance continues to improve. This demonstrates another significant trade-off between model output quality and diversity. More recent works (Havrilla et al., 2024a; Singh et al., 2024) find similar

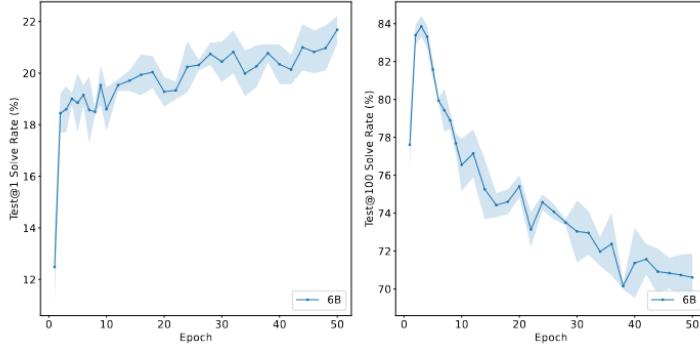


Figure 9: Trade-off between pass@1 (output quality) versus pass@100 (output diversity) throughout training.

behavior with RL fine-tuned models: pass@1 performance is greatly improved over the SFT baseline yet pass@n performance is not. However, (Havrilla et al., 2024a) also finds that simply training on more diverse data improves the pass@n and diversity of solutions of the resulting models. This suggests that the output quality and diversity of models can be controlled in part by the composition of quality and diversity in training data.

Better benchmarking of model QDC output distribution A better understanding of the relationship between model output quality and model output diversity is fundamental to the development efficient and effective iterative self-improvement algorithms. Models optimizing too heavily for diversity will fail to improve performance at the target task. Models optimizing too heavily for quality will struggle to explore diversely and quickly converge in the self-improvement process. However, we observe that the vast majority of model benchmarking work focuses almost exclusively on evaluating answer quality (Hendrycks et al., 2021b;a; Cobbe et al., 2021; Rein et al., 2023; Pang et al., 2024) by evaluating against a single expected answer. Comparatively few benchmarks exist which are equipped to evaluate solution diversity (Boratko et al., 2020). This disparity is likely due to the relative difficulty of checking for solution quality (checking for a single expected answer is easy) versus solution diversity. The majority of self-improvement/RL approaches also only reward for answer quality (Gulcehre et al., 2023; Bai et al., 2022a; Havrilla et al., 2024a; Sun et al., 2024; Singh et al., 2024). However, considering the importance of model output diversity for synthetic data generation and self-improvement capabilities, evaluation of model output diversity is also essential. Since the majority of benchmarks and self-improvement algorithms are designed only to optimize only for answer quality it is likely that answer diversity suffers. This emphasizes the need for better benchmarking of both model output diversity and the balance between model output quality and diversity.

Takeaways

- Model output quality and model output diversity trade-off.
- Many current models are benchmarked and optimized only for quality with output diversity likely suffering as a result.

Additionally, there are very few works investigating the role of complexity in iterative self-improvement algorithms for LLMs. As in Section 3.4, better understanding the relationship between model output complexity and quality and diversity will be essential for future algorithm development.

Open Questions

- How can we better benchmark model output diversity?
- How can we design self-improvement algorithms balancing output quality and diversity for optimal improvement?
- What is the relationship between model output complexity and quality and diversity?

4.5 QDC-AWARE SYNTHETIC DATA GENERATION ALGORITHMS

In the previous section we identified a trade-off between model output quality and model output diversity analogous to the trade-offs in training data from Section 3.4. As a result certain components of the synthetic data generation which promote quality may worsen dataset diversity and vice-versa. Most algorithms fail to take this trade-off into account. Further, very little research has been done investigating the synergistic and antagonistic effects of various pipeline components on dataset quality, diversity and complexity. However, a small set of synthetic data generation algorithms do attempt to directly optimize for both characteristics during the generation, filtration, or distillation phases. We call such algorithms *quality-diversity synthetic data generation algorithms*. Several of these algorithms are designed using tools from the quality-diversity and novelty search literature (Lehman et al., 2022; Ding et al., 2024; Bradley et al., 2023; Samvelyan et al., 2024b; Chan et al., 2024).

1. QDC Mechanisms in Data Generation One of the most widely known QD algorithms is MAP-Elites (Mouret & Clune, 2015). Given a sample space Ω , MAP-Elites describes each $\omega \in \Omega$ via a set of designed behavioral descriptors $T_1, \dots, T_k : \Omega \rightarrow \mathbb{N}$. These are used to discretize the sample space Ω into a hyper-grid. The algorithm’s goal is then to efficiently explore the grid (thereby maximizing diversity) while also selecting for high-quality solutions according to a pre-defined quality objective Q . New solutions are sampled from old-solutions via a pre-defined mutation operator. By mutating increasingly high-quality solutions across many different parts of the grid MAP-Elites simultaneously attempts to simultaneously maximize sample quality and sample diversity. Evolution through Language Models (**ELM**) (Lehman et al., 2022) was one of the first works to combine MAP-Elites with LLMs acting as a sophisticated mutation operator. They apply the LLM mutation driven algorithm to a synthetic robotic racing task where various *sodaracer* configurations are defined programmatically. The LLM is used to efficiently mutate programmatic configurations, improving over less sophisticated mutation baselines. Since the task is synthetic various notions of behavioral diversity are defined manually.

Quality-Diversity through Human Feedback (**QDHF**) (Ding et al., 2024) proposes to learn a sample level diversity function D by collecting human preference triples $(y_1, y_2, y_3, y_1 \sim y_i)$ where $y_1 \sim y_2$ indicates y_1 is more similar to y_2 than to y_3 . The diversity model is then trained with a contrastive objective on these preference triplets. MAP-Elites is then used to discover diverse and high-quality samples across several tasks. QDAIF (Bradley et al., 2023) extends the applicability of MAP-Elites to text generation domains by utilizing LLMs to both mutate and evaluate text through natural language AI feedback. Rainbow Teaming (Samvelyan et al., 2024b) applies MAP-Elites to generate adversarial red-teaming prompts for LLMs, using LLMs as a judge to evaluate responses. ACES (Pourcel et al., 2024) proposes a goal conditioned version of MAP-Elites for python program generation, utilizing LLMs to describe the relevant programming

techniques used in a solution. QDGS: Quality diversity generative sampling is used to repair biases in classifiers and improve diversity (Chang et al., 2024). Zhou et al. (2024) explores varying decoding strategies at inference time to balance quality and diversity.

2. QDC Mechanisms in Data Filtration The above QDC generation algorithms focus on controlling both quality and diversity during the data generation phase. A number of recent papers also investigate balancing synthetic dataset quality and diversity in the data filtration phase. InsTag measures both dataset diversity and complexity via LLM generated attributes (Lu et al., 2023b). ChatGPT is used to tag instructions by intention for each label, after which tags are combined if they are sufficiently similar. Dataset diversity is then measured by the diversity of tags occurring in the dataset. Complexity of a sample is measured by the number of tags it has. They then filter publicly available chat datasets using by selecting to produce complex, highly diverse fine-tuning data. Liu et al. (2024b) proposes new quality and complexity measures using in-context multi-sample comparison. They compare a number of quality, diversity, and complexity metrics on chat data, finding attribute based measurements coming from LLMs perform best.

3. QDC Mechanisms in Data Distillation Cideron et al. (2024) propose diversity-rewarded CFG distillation, which combines a distillation objective for maintaining generation quality, with a reinforcement learning objective that explicitly rewards diversity across generations. Their approach first distills classifier-free guidance (CFG) into model weights to preserve quality without the typical inference overhead, and then uses a diversity reward based on embedding similarity to encourage varied outputs. They further demonstrate that interpolating between models trained with different diversity coefficients creates a controllable quality-diversity front at deployment time. These results suggest that QDC-aware distillation strategies can effectively balance quality and diversity while reducing computational costs compared to traditional inference-time techniques. In a similar direction, (Omura et al., 2024) train an LLM to dynamically select its own sampling temperature using DPO.

Takeaways

- QDC synthetic data algorithms try to directly balance the quality and diversity of synthetic data.

While there exist several quality-diversity inspired algorithms for synthetic data generation, rarely is complexity simultaneously taken into account as a third important factor. We highlight this gap as an important open question.

Open Questions

- How can we design synthetic data algorithms jointly balancing quality, diversity, **and complexity?**

sk

5 QDC SYNTHETIC DATA ALGORITHMS OUTSIDE COMMON LLM TASKS

Most of the papers covered in the prior section focus on popular LLM benchmark tasks, e.g., question answering, instruction following, chat, math, and code. However, there are many of tasks outside this standard set to which LLM driven evolutionary/quality-diversity inspired algorithms have been successfully applied. We highlight several representative works in this section to illustrate the broad applicability and

effectiveness of these methods (Chen et al., 2023a; Romera-Paredes et al., 2024; Lu et al., 2023a; Fontaine et al., 2021b; Wang et al., 2023a; Team et al., 2021).

One particularly exciting application of generative models game environment design (Fontaine et al., 2021b; Zhang et al., 2023; Sudhakaran et al., 2024). Fontaine et al. (2021b) and later Sudhakaran et al. (2024) generate synthetic Mario levels to train a Mario RL policy. Omni-Epic (Zhang et al., 2023) generates a diverse, open-ended set of learning environments for RL agents using code-writing LLMs. These environments are used to teach agents increasingly complex skills by designing a curriculum of tasks tailored specifically to the agent’s abilities. Similar to Samvelyan et al. (2024b), quality-diversity search can be used to automatically red-team RL pre-trained agents (Bhatt et al., 2022; Samvelyan et al., 2024a). For example, Samvelyan et al. (2024a) show MAP-Elites can be used to discover exploitations in a strong soccer playing RL policy. Quality-diversity and synthetic data generation algorithms have also found many applications in robotics (Fontaine & Nikolaidis, 2021; Fontaine et al., 2021a; Yu et al., 2023a; Chen et al., 2023d; Mandlekar et al., 2023; Lu et al., 2023a). Several papers utilize quality-diversity approaches to generate human-robot interaction scenarios (Fontaine & Nikolaidis, 2021; Fontaine et al., 2021a). Many other works (Yu et al., 2023a; Chen et al., 2023d; Mandlekar et al., 2023; Lu et al., 2023a) generate synthetic data to train robotics policies in a variety of domains.

LLM driven evolutionary algorithms can also be used to solve complex, open-ended scientific optimization problems (Romera-Paredes et al., 2024; Chen et al., 2023a; Nasir et al., 2023). FunSearch Romera-Paredes et al. (2024) presents an evolutionary algorithm for producing programmatic solutions to combinatorics problems by sampling a small coding LLM. They show that, with enough sampling, novel solutions representing mathematically interesting counter-examples can be found for multiple open problems. Chen et al. (2023a) proposes to run an evolutionary neural architecture search using a coding LLM to produce network architectures in Pytorch. They are able to produce architectures competitive with SOTA methods while containing a fraction of the parameters in SOTA models. Nasir et al. (2023) shows the neural architecture search process can be improved using quality-diversity search. Zhang et al. (2024f) evolve Neural Cellular Automata (NCA) environment generators. PromptBreeder (Fernando et al., 2023b) and APE (Zhou et al., 2023b) evolve improved LLM prompts across a wide-range of tasks.

Finally, we highlight a set of papers motivating the development of and integration LLMs in increasingly open-ended problem solving environments. The ability of LLMs to explore and learn in such environments is (almost definitionally) necessary for the development of generally capable intelligence. Voyager (Wang et al., 2023a) proposes a complex system deploying GPT-4 in minecraft via API-based action interaction. Their system utilizes several key components including a continuously updated skill library, a planning module proposing candidate tasks, and a self-repair module attempting to fix malformed action code. They show the resulting agent significantly outperforms RL based agents (Hafner et al., 2024) in skill acquisition and open-ended world exploration. Team et al. (2021) discusses the essential role of open-ended learning in the development of more sophisticated, generally intelligent AI systems.

6 CONCLUSIONS AND OPEN QUESTIONS

Conclusions Over the course of this survey we examined the composition of synthetic data in terms of its quality, diversity, and complexity and the resulting effects on model generalization when used for training. We began in Section 2 by defining at a high-level what is meant by quality, diversity, and complexity in data. We then reviewed and categorized numerous practical measures of quality, diversity, and complexity in the literature. Overall, we found that domain specific, attribute measures utilizing LLMs-as-a-judge provide the best measures in complex tasks and domains in terms of correlation with downstream metrics. However, often these approaches require specific domain knowledge to implement effectively. In Section 3 we looked at the effects of different QDC compositions in training data (as measured by implementations in Section 2). We concluded that high-quality data primarily benefits in-distribution generalization, diverse data primarily

benefits OOD generalization, and appropriate levels of data complexity can benefit both. In addition, there is a natural trade-off between high-quality and highly-diverse data. As a result, there can be trade-offs for in-distribution and OOD generalization.

In Section 4 we taxonomized synthetic data generation algorithms in terms of the components each use to promote synthetic data quality, diversity, and complexity. Many algorithms promote quality by sampling from a large, SOTA LLM and maximize diversity by expanding the dataset of seed prompts. Very few algorithms explicitly consider trade-offs between quality, diversity, and complexity or attempt to maximize all three together. We then investigated the effect of training data QDC composition on the QDC of resulting synthetic data. We found a trade-off between model output quality and output diversity, resulting in higher-quality but less diverse synthetic data. Further, we observe that the majority of models today are evaluated almost exclusively for output quality, thereby limiting output diversity and the diversity of generated synthetic data. We argue that future algorithms for synthetic data generation and self-improvement should carefully trade-off data QDC composition and highlight a number of works going in this direction.

Open Questions Our discussions on QDC in synthetic data highlight many open questions and potential directions for future research. Most important is the development of new synthetic data generation algorithms for self-improvement that optimally trade-off QDC composition both in training data and model output distribution. Development of such algorithms will also require better benchmarking of model output *diversity* and *complexity* in addition to quality. Better benchmarking will allow researchers to identify the frontier of model output QDC trade-offs and find techniques moving the entire frontier forward. In particular, in comparison to quality and diversity, complexity in both data and model output has been understudied. Better understanding of what factors affect data/model complexity and how complexity is affected by quality and diversity will be crucial to future algorithm design. Finally, development of better measures of quality, diversity, and complexity will be necessary as tasks become increasingly complex and open-ended. Ideally these measures should be general purpose, inexpensive to compute, and correlate well with downstream metrics of interest.

Acknowledgements Many, many thanks to Minqi Jiang, Mikayel Samvelyan, Alex Bukharin, and Reshinh Adithyan for their helpful feedback on earlier versions of the survey. Additionally, thank you to Hailey Schoelkopf, Louis Castricato, and Yasin Abbasi Yadkori for enlightening conversations about the role of quality, diversity, and complexity in LLMs!

REFERENCES

- Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023. URL <https://arxiv.org/abs/2303.09540>.
- Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.
- Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3554–3565, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.278. URL <https://aclanthology.org/2021.naacl-main.278>.

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL <https://arxiv.org/abs/2402.14740>.
- Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023a. URL <https://openreview.net/forum?id=9Tze4oy4lw>.
- Alon Albalak, Colin Raffel, and William Yang Wang. Improving few-shot generalization by exploring and exploiting auxiliary data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL <https://openreview.net/forum?id=JDnLXc4NOn>.
- 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, Tatsumori Hashimoto, and William Yang Wang. A survey on data selection for language models, 2024. URL <https://arxiv.org/abs/2402.16827>.
- Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models, 2024. URL <https://arxiv.org/abs/2408.11791>.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a. URL <https://arxiv.org/abs/2204.05862>.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamara Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022b. URL <https://arxiv.org/abs/2212.08073>.
- André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. Comprehensive exploration of synthetic data generation: A survey, 2024. URL <https://arxiv.org/abs/2401.02524>.
- Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, February 2024. URL <https://huggingface.co/datasets/HuggingFaceTB/cosmopedia>.
- Charles H. Bennett. Logical depth and physical complexity. 1988. URL <https://api.semanticscholar.org/CorpusID:6365510>.
- Varun Bhatt, Bryon Tjanaka, Matthew Fontaine, and Stefanos Nikolaidis. Deep surrogate assisted generation of environments. *Advances in Neural Information Processing Systems*, 35:37762–37777, 2022.

Michael Boratko, Xiang Lorraine Li, Rajarshi Das, Tim O’Gorman, Dan Le, and Andrew McCallum. Protoqa: A question answering dataset for prototypical common-sense reasoning, 2020. URL <https://arxiv.org/abs/2005.00771>.

Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. Quality-diversity through ai feedback, 2023. URL <https://arxiv.org/abs/2310.13032>.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL <https://arxiv.org/abs/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, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020b. URL <https://arxiv.org/abs/2005.14165>.

Alexander Bukharin and Tuo Zhao. Data diversity matters for robust instruction tuning, 2024. URL <https://arxiv.org/abs/2311.14736>.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yingning Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL <https://arxiv.org/abs/2312.09390>.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models, 2024. URL <https://arxiv.org/abs/2307.06290>.

Navoneel Chakrabarty. A machine learning approach to comment toxicity classification, 2019. URL <https://arxiv.org/abs/1903.06765>.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas, 2024. URL <https://arxiv.org/abs/2406.20094>.

Allen Chang, Matthew C. Fontaine, Serena Booth, Maja J. Matarić, and Stefanos Nikolaidis. Quality-diversity generative sampling for learning with synthetic data, 2024. URL <https://arxiv.org/abs/2312.14369>.

Wang Chao, Jiaxuan Zhao, Licheng Jiao, Lingling Li, Fang Liu, and Shuyuan Yang. A match made in consistency heaven: when large language models meet evolutionary algorithms. *arXiv preprint arXiv:2401.10510*, 2024.

- Konstantinos Chatzilygeroudis, Antoine Cully, Vassilis Vassiliades, and Jean-Baptiste Mouret. Quality-diversity optimization: a novel branch of stochastic optimization. In *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*, pp. 109–135. Springer, 2021.
- Angelica Chen, David M. Dohan, and David R. So. Evoprompting: Language models for code-level neural architecture search, 2023a. URL <https://arxiv.org/abs/2302.14838>.
- Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. On the diversity of synthetic data and its impact on training large language models, 2024a. URL <https://arxiv.org/abs/2410.15226>.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagenus: Training a better alpaca with fewer data, 2024b. URL <https://arxiv.org/abs/2307.08701>.
- Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models, 2023b. URL <https://arxiv.org/abs/2307.14430>.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, 2023c. URL <https://arxiv.org/abs/2211.12588>.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models, 2024c. URL <https://arxiv.org/abs/2401.01335>.
- Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023d.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL <https://lmsys.org/blog/2023-03-30-vicuna/>.
- Geoffrey Cideron, Andrea Agostinelli, Johan Ferret, Sertan Girgin, Romuald Elie, Olivier Bachem, Sarah Perrin, and Alexandre Ramé. Diversity-rewarded cfg distillation, 2024. URL <https://arxiv.org/abs/2410.06084>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Antoine Cully and Yiannis Demiris. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259, 2017.
- Christine E. DeMars. Item response theory. *Assessing Measurement Invariance for Applied Research*, 2010. URL <https://api.semanticscholar.org/CorpusID:15155431>.
- Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Shafiq Joty, Boyang Li, and Lidong Bing. Is gpt-3 a good data annotator?, 2023. URL <https://arxiv.org/abs/2212.10450>.
- Li Ding, Jenny Zhang, Jeff Clune, Lee Spector, and Joel Lehman. Quality diversity through human feedback: Towards open-ended diversity-driven optimization, 2024. URL <https://arxiv.org/abs/2310.12103>.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition, 2024. URL <https://arxiv.org/abs/2310.05492>.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment, 2023. URL <https://arxiv.org/abs/2304.06767>.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. Glam: Efficient scaling of language models with mixture-of-experts, 2022. URL <https://arxiv.org/abs/2112.06905>.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay

Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhota, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vitor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaoqian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024a. URL <https://arxiv.org/abs/2407.21783>.

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

- Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023. URL <https://arxiv.org/abs/2305.07759>.
- Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with \mathcal{V} -usable information. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5988–6008. PMLR, 17–23 Jul 2022.
- Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training ... for now, 2023. URL <https://arxiv.org/abs/2312.04567>.
- Simin Fan and Martin Jaggi. Irreducible curriculum for language model pretraining, 2023. URL <https://arxiv.org/abs/2310.15389>.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Prompt-breeder: Self-referential self-improvement via prompt evolution, 2023a.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Prompt-breeder: Self-referential self-improvement via prompt evolution, 2023b. URL <https://arxiv.org/abs/2309.16797>.
- Matthew Fontaine and Stefanos Nikolaidis. A quality diversity approach to automatically generating human-robot interaction scenarios in shared autonomy. *Robotics: Science and Systems (RSS)*, 2021.
- Matthew Fontaine, Sophie Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the importance of environments in human-robot coordination. *Robotics: Science and Systems (RSS)*, 2021a.
- Matthew C. Fontaine, Ruilin Liu, Ahmed Khalifa, Jignesh Modi, Julian Togelius, Amy K. Hoover, and Stefanos Nikolaidis. Illuminating mario scenes in the latent space of a generative adversarial network, 2021b. URL <https://arxiv.org/abs/2007.05674>.
- Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. Stream of search (sos): Learning to search in language, 2024a. URL <https://arxiv.org/abs/2404.03683>.
- Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Better synthetic data by retrieving and transforming existing datasets, 2024b. URL <https://arxiv.org/abs/2404.14361>.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL <https://arxiv.org/abs/2101.00027>.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing, 2024. URL <https://arxiv.org/abs/2305.11738>.
- Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks, 2017. URL <https://arxiv.org/abs/1704.03003>.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling, 2023. URL <https://arxiv.org/abs/2308.08998>.

- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.
- Xu Guo and Yiqiang Chen. Generative ai for synthetic data generation: Methods, challenges and the future, 2024. URL <https://arxiv.org/abs/2403.04190>.
- Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL <https://arxiv.org/abs/2301.04104>.
- Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation, 2019. URL <https://arxiv.org/abs/1904.02792>.
- Alex Havrilla and Wenjing Liao. Understanding scaling laws with statistical and approximation theory for transformer neural networks on intrinsically low-dimensional data, 2024. URL <https://arxiv.org/abs/2411.06646>.
- Alex Havrilla, Yuqing Du, Sharath Chandra Raparth, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. Teaching large language models to reason with reinforcement learning, 2024a. URL <https://arxiv.org/abs/2403.04642>.
- Alex Havrilla, Sharath Raparth, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. Glore: When, where, and how to improve llm reasoning via global and local refinements, 2024b. URL <https://arxiv.org/abs/2402.10963>.
- Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. trlX: A framework for large scale reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8578–8595, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.530. URL <https://aclanthology.org/2023.emnlp-main.530>.
- Joy He-Yueya, Wanjing Anya Ma, Kanishk Gandhi, Benjamin W Domingue, Emma Brunskill, and Noah D Goodman. Psychometric alignment: Capturing human knowledge distributions via language models. *arXiv preprint arXiv:2407.15645*, 2024.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL <https://arxiv.org/abs/2009.03300>.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021b. URL <https://arxiv.org/abs/2103.03874>.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL <https://arxiv.org/abs/2203.15556>.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL <https://arxiv.org/abs/1904.09751>.
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor, 2022. URL <https://arxiv.org/abs/2212.09689>.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve, 2022. URL <https://arxiv.org/abs/2210.11610>.

Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktaschel. Open-endedness is essential for artificial superhuman intelligence, 2024.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sébastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 2023.

Minqi Jiang, Tim Rocktaschel, and Edward Grefenstette. General intelligence requires rethinking exploration. *Royal Society Open Science*, 10(6):230539, 2023.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. Self-planning code generation with large language models, 2024a. URL <https://arxiv.org/abs/2303.06689>.

Yiding Jiang, Allan Zhou, Zhili Feng, Sadhika Malladi, and J. Zico Kolter. Adaptive data optimization: Dynamic sample selection with scaling laws, 2024b. URL <https://arxiv.org/abs/2410.11820>.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. Mission: Impossible language models, 2024. URL <https://arxiv.org/abs/2401.06416>.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.

Sungdong Kim, Sanghwan Bae, Jamin Shin, Soyoung Kang, Donghyun Kwak, Kang Min Yoo, and Minjoon Seo. Aligning large language models through synthetic feedback, 2023. URL <https://arxiv.org/abs/2305.13735>.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlf on llm generalisation and diversity, 2024. URL <https://arxiv.org/abs/2310.06452>.

Tomasz Korbak, Ethan Perez, and Christopher L Buckley. Rl with kl penalties is better viewed as bayesian inference, 2022. URL <https://arxiv.org/abs/2205.11275>.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Srivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning, 2024. URL <https://arxiv.org/abs/2409.12917>.

Jack Lanchantin, Shubham Toshniwal, Jason Weston, Arthur Szlam, and Sainbayar Sukhbaatar. Learning to reason and memorize with self-notes, 2023. URL <https://arxiv.org/abs/2305.00833>.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning, 2022. URL <https://arxiv.org/abs/2207.01780>.

Alycia Lee, Brando Miranda, Sudharsan Sundar, and Sanmi Koyejo. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data, 2023. URL <https://arxiv.org/abs/2306.13840>.

Bruce W. Lee, Hyunsoo Cho, and Kang Min Yoo. Instruction tuning with human curriculum, 2024. URL <https://arxiv.org/abs/2310.09518>.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Mureşan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL <https://aclanthology.org/2022.acl-long.577>.

Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011a.

Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 211–218, 2011b.

Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. Evolution through large models, 2022. URL <https://arxiv.org/abs/2206.08896>.

Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities, 2024a. URL <https://arxiv.org/abs/2403.04706>.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2024b. URL <https://arxiv.org/abs/2406.11794>.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen Rambow (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL <https://aclanthology.org/N16-1014>.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2016b. URL <https://arxiv.org/abs/1510.03055>.

Ming Li and Paul M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, 1997.

Ming Li, Yong Zhang, Zhitao Li, Juhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning, 2024c. URL <https://arxiv.org/abs/2308.12032>.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation, 2024d. URL <https://arxiv.org/abs/2308.06259>.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization, 2023a. URL <https://arxiv.org/abs/2210.15097>.

Xiaomin Li, Mingye Gao, Zhiwei Zhang, Chang Yue, and Hong Hu. Rule-based data selection for large language models. *arXiv preprint arXiv:2410.04715*, 2024e.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023b.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. Entropic distribution matching in supervised fine-tuning of llms: Less overfitting and better diversity, 2024f. URL <https://arxiv.org/abs/2408.16673>.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL <https://arxiv.org/abs/2305.20050>.

Rensis Likert. *A technique for the measurement of attitudes / by Rensis Likert*. Archives of psychology ; no. 140. [s.n.], New York, 1932.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.

Bingbin Liu, Sébastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. Tinygsm: achieving ↗80URL <https://arxiv.org/abs/2312.09241>.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Exploration with principles for diverse ai supervision, 2023b. URL <https://arxiv.org/abs/2310.08899>.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. Best practices and lessons learned on synthetic data for language models, 2024a. URL <https://arxiv.org/abs/2404.07503>.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning, 2024b. URL <https://arxiv.org/abs/2312.15685>.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey, 2024. URL <https://arxiv.org/abs/2406.15126>.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, toxicity, 2023. URL <https://arxiv.org/abs/2305.13169>.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder 2 and the stack v2: The next generation, 2024. URL <https://arxiv.org/abs/2402.19173>.

Cong Lu, Philip J. Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay, 2023a. URL <https://arxiv.org/abs/2303.06614>.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Instag: Instruction tagging for analyzing supervised fine-tuning of large language models, 2023b. URL <https://arxiv.org/abs/2308.07074>.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct, 2023a. URL <https://arxiv.org/abs/2308.09583>.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct, 2023b. URL <https://arxiv.org/abs/2306.08568>.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL <https://arxiv.org/abs/2303.17651>.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albala. Generative reward models, 2024. URL <https://arxiv.org/abs/2410.12832>.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj S. Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the Conference on Robot Learning, CoRL*, 2023. URL <https://proceedings.mlr.press/v229/mandlekar23a.html>.

Philip M. McCarthy and Scott Jarvis. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment, 2009.

Elliot Meyerson, Mark J Nelson, Herbie Bradley, Arash Moradi, Amy K Hoover, and Joel Lehman. Language model crossover: Variation through few-shot prompting. *arXiv preprint arXiv:2302.12170*, 2023.

Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning, 2023. URL <https://arxiv.org/abs/2308.00436>.

John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: On the strong correlation between out-of-distribution and in-distribution generalization, 2021. URL <https://arxiv.org/abs/2107.04649>.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt, 2022. URL <https://arxiv.org/abs/2206.07137>.

Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, Inc., USA, 2009. ISBN 0195124413.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites, 2015. URL <https://arxiv.org/abs/1504.04909>.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023. URL <https://arxiv.org/abs/2306.02707>.

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. Diversity of thought improves reasoning abilities of llms, 2024. URL <https://arxiv.org/abs/2310.07088>.

Muhammad U Nasir, Sam Earle, Julian Togelius, Steven James, and Christopher Cleghorn. Llmatic: Neural architecture search via large language models and quality-diversity optimization. *arXiv preprint arXiv:2306.01102*, 2023.

Ansong Ni, Miltiadis Allamanis, Arman Cohan, Yinlin Deng, Kensen Shi, Charles Sutton, and Pengcheng Yin. Next: Teaching large language models to reason about code execution, 2024. URL <https://arxiv.org/abs/2404.14662>.

Motoki Omura, Yasuhiro Fujita, and Toshiki Kataoka. Entropy controllable direct preference optimization, 2024. URL <https://arxiv.org/abs/2411.07595>.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela

Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In Piotr Bański, Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder, Simon Clematide, Marc Kupietz, Harald Lünen, and Caroline Iliadi (eds.), *Proceedings of the Workshop on Challenges in the Management of Large Corpora*, Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pp. 9 – 16, Mannheim, 2019. Leibniz-Institut für Deutsche Sprache. doi: 10.14618/ids-pub-9021. URL <http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215>.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.

Laura O’Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Mathematical and Empirical Understanding of Foundation Models (ME-FoMo) workshop*, 2024.

Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. West-of-n: Synthetic preference generation for improved reward modeling, 2024. URL <https://arxiv.org/abs/2401.12086>.

Norman Packard, Mark A Bedau, Alastair Channon, Takashi Ikegami, Steen Rasmussen, Kenneth O Stanley, and Tim Taylor. An overview of open-ended evolution: Editorial introduction to the open-ended evolution ii special issue. *Artificial life*, 25(2):93–103, 2019.

Rohan Pandey. gzip predicts data-dependent scaling laws, 2024. URL <https://arxiv.org/abs/2405.16684>.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization, 2024. URL <https://arxiv.org/abs/2404.19733>.

- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002a. URL <https://api.semanticscholar.org/CorpusID:11080756>.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002b.
- Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL <https://arxiv.org/abs/2406.17557>.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828, 2021.
- Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tiny-benchmarks: evaluating llms with fewer examples. *arXiv preprint arXiv:2402.14992*, 2024.
- Julien Pourcel, Cédric Colas, Gaia Molinaro, Pierre-Yves Oudeyer, and Laetitia Teodorescu. Aces: Generating diverse programming puzzles with autotelic generative models, 2024. URL <https://arxiv.org/abs/2310.10692>.
- Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016.
- Yulei Qin, Yuncheng Yang, Pengcheng Guo, Gang Li, Hang Shao, Yuchen Shi, Zihan Xu, Yun Gu, Ke Li, and Xing Sun. Unleashing the power of data tsunami: A comprehensive survey on data assessment and selection for instruction tuning of language models, 2024. URL <https://arxiv.org/abs/2408.02085>.
- Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching LLM agents how to self-improve. In *ICML 2024 Workshop on LLMs and Cognition*, 2024. URL <https://openreview.net/forum?id=ze0nodITam>.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL <https://arxiv.org/abs/2305.18290>.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435. URL <https://jmlr.org/papers/volume21/20-074/20-074.pdf>.
- Georg Rasch. *Probabilistic models for some intelligence and attainment tests*. ERIC, 1993.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL <https://arxiv.org/abs/1908.10084>.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof qa benchmark, 2023. URL <https://arxiv.org/abs/2311.12022>.
- Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm, 2021. URL <https://arxiv.org/abs/2102.07350>.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, I Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6. URL <https://doi.org/10.1038/s41586-023-06924-6>.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance, 2024. URL <https://arxiv.org/abs/2405.10938>.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. URL <https://arxiv.org/abs/1409.0575>.

Mikayel Samvelyan, Davide Paglieri, Minqi Jiang, Jack Parker-Holder, and Tim Rocktäschel. Multi-agent diagnostics for robustness via illuminated diversity. *arXiv preprint arXiv:2401.13460*, 2024a.

Mikayel Samvelyan, Sharath Chandra Raparthi, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow teaming: Open-ended generation of diverse adversarial prompts, 2024b. URL <https://arxiv.org/abs/2402.16822>.

Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander Kranias, John J. Nay, Kshitij Gupta, and Aran Komatsuzaki. Arb: Advanced reasoning benchmark for large language models, 2023. URL <https://arxiv.org/abs/2307.13692>.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL <https://arxiv.org/abs/2302.04761>.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL <https://aclanthology.org/P16-1009>.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RI on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold, 2024. URL <https://arxiv.org/abs/2406.14532>.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Huawei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.

Utkarsh Sharma and Jared Kaplan. A neural scaling law from the dimension of the data manifold, 2020. URL <https://arxiv.org/abs/2004.10802>.

Vasu Sharma, Karthik Padthe, Newsha Ardalani, Kushal Tirumala, Russell Howes, Hu Xu, Po-Yao Huang, Shang-Wen Li, Armen Aghajanyan, Gargi Ghosh, and Luke Zettlemoyer. Text quality-based pruning for efficient training of language models, 2024. URL <https://arxiv.org/abs/2405.01582>.

- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL <https://arxiv.org/abs/2303.11366>.
- Olivier Sigaud, Gianluca Baldassarre, Cedric Colas, Stephane Doncieux, Richard Duro, Nicolas Perrin-Gilbert, and Vieri-Giuliano Santucci. A definition of open-ended learning problems for goal-conditioned agents. *arXiv preprint arXiv:2311.00344*, 2023.
- Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models, 2024. URL <https://arxiv.org/abs/2312.06585>.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- Asilia Song. A little taxonomy of open-endedness. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022.
- Lisa B Soros, Joel Lehman, and Kenneth O Stanley. Open-endedness: The last grand challenge you've never heard of, 2017.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL <https://arxiv.org/abs/2009.01325>.
- Shyam Sudhakaran, Miguel González-Duque, Matthias Freiberger, Claire Glanois, Elias Najarro, and Sebastian Risi. Mariogpt: Open-ended text2level generation through large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision, 2024. URL <https://arxiv.org/abs/2403.09472>.
- Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents, 2021. URL <https://arxiv.org/abs/2107.12808>.
- David Thissen. Marginal maximum likelihood estimation for the one-parameter logistic model. *Psychometrika*, 47:175–186, 1982.
- Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. Toward self-improvement of llms via imagination, searching, and criticizing, 2024. URL <https://arxiv.org/abs/2404.12253>.
- Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. D4: Improving llm pretraining via document de-duplication and diversification, 2023. URL <https://arxiv.org/abs/2308.12284>.

Bryon Tjanaka. Quantifying efficiency in quality diversity optimization. In *Workshop on Benchmarks for Quality-Diversity Algorithms at GECCO*, 2022.

Shubham Toshniwal, Ivan Moshkov, Sean Narendhiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset, 2024. URL <https://arxiv.org/abs/2402.10176>.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023a. URL <https://arxiv.org/abs/2302.13971>.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenjin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madiam Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yunling Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Bin Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL <https://arxiv.org/abs/2307.09288>.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. URL <https://arxiv.org/abs/2211.14275>.

Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Wu, and Graham Neubig. Prompt2model: Generating deployable models from natural language instructions, 2023. URL <https://arxiv.org/abs/2308.12261>.

Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

Evan Wang, Federico Cassano, Catherine Wu, Yunfeng Bai, Will Song, Vaskar Nath, Ziwen Han, Sean Hendryx, Summer Yue, and Hugh Zhang. Planning in natural language improves llm search for code generation, 2024a. URL <https://arxiv.org/abs/2409.03733>.

Fei Wang, Ninareh Mehrabi, Palash Goyal, Rahul Gupta, Kai-Wei Chang, and Aram Galstyan. Data advisor: Dynamic data curation for safety alignment of large language models, 2024b. URL <https://arxiv.org/abs/2410.05269>.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023a. URL <https://arxiv.org/abs/2305.16291>.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. A survey on data selection for llm instruction tuning, 2024c. URL <https://arxiv.org/abs/2402.05123>.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024d. URL <https://arxiv.org/abs/2312.08935>.

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O’Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Shepherd: A critic for language model generation, 2023b. URL <https://arxiv.org/abs/2308.04592>.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kunal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks, 2022. URL <https://arxiv.org/abs/2204.07705>.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023c. URL <https://arxiv.org/abs/2212.10560>.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. URL <https://arxiv.org/abs/2109.01652>.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.

Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V. Le. Simple synthetic data reduces sycophancy in large language models, 2024. URL <https://arxiv.org/abs/2308.03958>.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=SJeYe0NtvH>.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4003–4012, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.494>.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. Qurating: Selecting high-quality data for training language models, 2024. URL <https://arxiv.org/abs/2402.09739>.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, may 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL <https://doi.org/10.1007/BF00992696>.

Xingyu Wu, Sheng hao Wu, Jibin Wu, Liang Feng, and Kay Chen Tan. Evolutionary computation in the era of large language model: Survey and roadmap, 2024a. URL <https://arxiv.org/abs/2401.10034>.

- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2024b. URL <https://arxiv.org/abs/2408.00724>.
- Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling, 2023. URL <https://arxiv.org/abs/2302.03169>.
- Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36, 2024.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Dixin Jiang. Wizardlm: Empowering large language models to follow complex instructions, 2023. URL <https://arxiv.org/abs/2304.12244>.
- Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment, 2024a. URL <https://arxiv.org/abs/2404.01730>.
- Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrastive distillation for language model alignment, 2024b. URL <https://arxiv.org/abs/2307.12950>.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL <https://arxiv.org/abs/2305.10601>.
- Jiacheng Ye, Jiahui Gao, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Progen: Progressive zero-shot dataset generation via in-context feedback, 2022a. URL <https://arxiv.org/abs/2210.12329>.
- Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation, 2022b. URL <https://arxiv.org/abs/2202.07922>.
- Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance, 2024a. URL <https://arxiv.org/abs/2403.16952>.
- Zihuiwen Ye, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. Improving reward models with synthetic critiques, 2024b. URL <https://arxiv.org/abs/2405.20850>.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2024. URL <https://arxiv.org/abs/2309.12284>.
- Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023a.
- Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: A tale of diversity and bias, 2023b. URL <https://arxiv.org/abs/2306.15895>.

- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL <https://arxiv.org/abs/2308.01825>.
- Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhua Chen. Mammoth: Building math generalist models through hybrid instruction tuning, 2023. URL <https://arxiv.org/abs/2309.05653>.
- Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhua Chen. Mammoth2: Scaling instructions from the web, 2024. URL <https://arxiv.org/abs/2405.03548>.
- Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. Automatic instruction evolving for large language models, 2024. URL <https://arxiv.org/abs/2406.00770>.
- Dylan Zhang, Justin Wang, and Francois Charton. Instruction diversity drives generalization to unseen tasks, 2024a. URL <https://arxiv.org/abs/2402.10891>.
- Dylan Zhang, Justin Wang, and Francois Charton. **Only-IF**:revealing the decisive effect of instruction diversity on generalization, 2024b. URL <https://arxiv.org/abs/2410.04717>.
- Jenny Zhang, Joel Lehman, Kenneth Stanley, and Jeff Clune. Omni: Open-endedness via models of human notions of interestingness. *arXiv preprint arXiv:2306.01711*, 2023.
- Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2024c. URL <https://arxiv.org/abs/2408.15240>.
- Yifan Zhang, Yifan Luo, Yang Yuan, and Andrew C Yao. Autonomous data selection with language models for mathematical texts. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024d. URL <https://openreview.net/forum?id=bBF077z8LF>.
- Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, Zico Kolter, and Daphne Ippolito. Forcing diffuse distributions out of language models, 2024e. URL <https://arxiv.org/abs/2404.10859>.
- Yulun Zhang, Matthew Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. Arbitrarily scalable environment generators via neural cellular automata. *Advances in Neural Information Processing Systems*, 36, 2024f.
- Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1053. URL <https://aclanthology.org/D19-1053>.
- Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Fei Huang, Yongbin Li, and Nevin L. Zhang. A preliminary study of the intrinsic relationship between complexity and alignment, 2024. URL <https://arxiv.org/abs/2308.05696>.
- Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: a cross-system benchmark for formal olympiad-level mathematics, 2022. URL <https://arxiv.org/abs/2109.00110>.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL <https://arxiv.org/abs/2306.05685>.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*, 2022.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivas Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment, 2023a. URL <https://arxiv.org/abs/2305.11206>.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers, 2023b. URL <https://arxiv.org/abs/2211.01910>.

Yuxuan Zhou, Margret Keuper, and Mario Fritz. Balancing diversity and risk in llm sampling: How to select your method and parameter for open-ended text generation, 2024. URL <https://arxiv.org/abs/2408.13586>.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texxygen: A benchmarking platform for text generation models, 2018. URL <https://arxiv.org/abs/1802.01886>.

Yan Zhuang, Qi Liu, Yuting Ning, Weizhe Huang, Rui Lv, Zhenya Huang, Guanhao Zhao, Zheng Zhang, Qingyang Mao, Shijin Wang, et al. Efficiently measuring the cognitive ability of llms: An adaptive testing perspective. *arXiv preprint arXiv:2306.10512*, 2023.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL <https://arxiv.org/abs/1909.08593>.

A TABLE OF QDC METRICS

Table 1: Clustering of Works Based on Quality Metrics

Category	Subcategory	Works
Ground Truth	Computational Correctness	(Yu et al., 2024), (Toshniwal et al., 2024), (Singh et al., 2024)
	Code Correctness	(Pourcel et al., 2024)
	N-gram overlap metrics such as BLEU	(Papineni et al., 2002b), (Samvelyan et al., 2024b)
	MAUVE	(Ye et al., 2022a)
Reward Modeling	Bradley–Terry	(Ziegler et al., 2020), (Ouyang et al., 2022)
	Outcome-Based	(Uesato et al., 2022)
	Process-Based	(Lightman et al., 2023)
Attribute-Based	HHH (Helpfulness, Honesty, and Harmlessness) Comparison Evaluations	(Bai et al., 2022a)
	Comparison Evaluations for Summarization	(Stiennon et al., 2022)
	Constitutional AI Critiques	(Bai et al., 2022b)
	AI-Generated Comparisons for Categories of Human Intuitions about Data Quality	(Wettig et al., 2024)
	Quality Increase through Evolution	(Wettig et al., 2024)
	UniEval	(Zhong et al., 2022)
Other	Student Solve Rate	(Havrilla et al., 2024a)
	Downstream Performance Increase	(Zhou et al., 2023a), (Viswanathan et al., 2023), (Gandhi et al., 2024a)
	Binary Classification (High-Quality vs Unfiltered Data)	(Brown et al., 2020b), (Gao et al., 2020), (Xie et al., 2023)
	Perplexity	(Wenzek et al., 2020), (Sharma et al., 2024)

Table 2: Clustering of Works Based on Diversity Metrics

Category	Subcategory	Works
Lexical	Inter-Sample N-gram Frequency	(Yu et al., 2023b)
Attribute-Based	Number of Unique Task Types	(Wang et al., 2022)
	Number of Unique Languages	(Wang et al., 2022)
	Number of Unique Domains or Topics	(Wang et al., 2022)
	Number of Unique Root Verbs and Nouns	(Li et al., 2024d)
	Skills Used to Arrive at a Solution	(Pourcel et al., 2024)
	Order of Mathematical Operations	(Tian et al., 2024), (Havrilla et al., 2024a)
	Attribute Labels	(Zhou et al., 2023a), (Wang et al., 2022)
	MAP-Elites Coverage	(Samvelyan et al., 2024b), (Bradley et al., 2023)
	Automatically Discovered Attributes	(Yu et al., 2023b)
	Average Pairwise Cosine Similarity	(Yu et al., 2023b), (Kirk et al., 2024), (Yu et al., 2024)
Embedding	i-th Nearest Neighbor	(Cao et al., 2024)
	SemDeDup	(Abbas et al., 2023)
	Facility Location Function	(Reimers & Gurevych, 2019), (Bukharin & Zhao, 2024)
	Diversity Coefficient for Natural Language	(Lee et al., 2023)
Other	NLI Diversity	(Kirk et al., 2024)
	MTLD	(McCarthy & Jarvis, 2009)
	N-gram Frequency	(Li et al., 2016a)

Table 3: Clustering of Works Based on Complexity Metrics

Category	Subcategory	Works
Attribute-Based	Semantic Node Count	(Zhao et al., 2024)
	Number of Intention and Semantics Tags	(Lu et al., 2023b)
	Direct scoring	(Chen et al., 2024b)
	Difficulty Level	(Hendrycks et al., 2021b)
	Evolution Complexity	(Liu et al., 2024b)
Other	Instruction Following Difficulty	(Li et al., 2024c)
	Parse Tree Complexity	(Sharma et al., 2024)
	Perplexity	(Albalak et al., 2023a)
	Instruction Length	(Liu et al., 2024b)

B TABLE OF QDC MECHANISMS IN SYNTHETIC DATA

Table 4: Clustering of Works Based on Quality Mechanisms

Category	Subcategory	Works
Data Generation	Using Strong Teacher Models	(Mukherjee et al., 2023), (Gunasekar et al., 2023; Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024), (Chiang et al., 2023), (OpenAI et al., 2024), (Wang et al., 2022), (Honovich et al., 2022)
	Generating Complex Chain-of-Thought	(Gunasekar et al., 2023; Li et al., 2023b; Javaheripi et al., 2023; Abdin et al., 2024), (Mukherjee et al., 2023)(Yu et al., 2024), (Liu et al., 2023a), (Yue et al., 2023), (Toshniwal et al., 2024)
	Generating Pairwise Preferences	(Kim et al., 2023), (Yang et al., 2024b), (Li et al., 2023a), (Pace et al., 2024)
	Meta-Prompting and Prompt Programming	(Reynolds & McDonell, 2021)
Data Filtration	Filtering Using Annotations	(Zhou et al., 2023a), (Chen et al., 2024b), (Cao et al., 2024)
	Using Trained Reward Models	(Bukharin & Zhao, 2024), (Gulcehre et al., 2023), (Singh et al., 2024)
	Filtering in RL Approaches	(Havrilla et al., 2024a), (Dong et al., 2023), (Yuan et al., 2023)
Data Distillation	RL Algorithms Influencing Quality	(Rafailov et al., 2024), (Pang et al., 2024), (Setlur et al., 2024), (Schulman et al., 2017), (Williams, 1992), (Ahmadian et al., 2024)
	Applying RL to Improve Reasoning	(Havrilla et al., 2024a), (Wang et al., 2024d), (Sun et al., 2024)
	Unlikelihood Training	(Welleck et al., 2020)

Table 5: Clustering of Works Based on Diversity Mechanisms

Category	Subcategory	Works
Data Generation	Diverse Decoding Strategies	(Viswanathan et al., 2023), (Havrilla et al., 2024a), (Ye et al., 2022b), (Holtzman et al., 2020), (Welleck et al., 2020)
	Impact of Prompting on Output Diversity	(Jiang et al., 2024a), (Naik et al., 2024), (Chan et al., 2024), (Toshniwal et al., 2024), (Yu et al., 2023b), (Bradley et al., 2023), (Fernando et al., 2023a), (Yang et al., 2024b)
	Impact of Diverse Seed Datasets	<i>Generation algorithms:</i> (Wang et al., 2023c), (Toshniwal et al., 2024), (Yu et al., 2024); <i>Adaptation algorithms:</i> (Schick et al., 2023), (Wang & Komatsuzaki, 2021), (Viswanathan et al., 2023), (Gandhi et al., 2024b), (Ding et al., 2023), (Huang et al., 2022), (Lanchantin et al., 2023), (Yue et al., 2024), (Agarwal et al., 2021)
	Hybrid Methods	(Ben Allal et al., 2024), (Sennrich et al., 2016), (Li et al., 2024d)
	Augmenting with External Tools	(Ni et al., 2024), (Gou et al., 2024), (Lozhkov et al., 2024)
	Search	(Cobbe et al., 2021), (Yao et al., 2023), (Tian et al., 2024), (Gandhi et al., 2024a), (Shinn et al., 2023), (Madaan et al., 2023), (Havrilla et al., 2024b), (Qu et al., 2024), (Miao et al., 2023), (Wang et al., 2024a)
	Model Ensembling	(Yuan et al., 2023), (Liu et al., 2023a), (Kim et al., 2023)
Data Filtration	Data Deduplication	(Wenzek et al., 2020), (Abbas et al., 2023), (Yu et al., 2023a), (Havrilla et al., 2024a), (Tian et al., 2024), (Lu et al., 2023b)
Data Distillation	Training Models for Improved Output Diversity	(Zhang et al., 2024e), (Li et al., 2024f), (Li et al., 2016b), (Havrilla et al., 2024a)

Table 6: Clustering of Works Based on Complexity Mechanisms

Category	Subcategory	Works
Data Generation	Evolving Samples for Complexity	(Xu et al., 2023), (Luo et al., 2023b), (Luo et al., 2023a), (Zeng et al., 2024)
	Gathering Feedback for RL	(Luo et al., 2023a)
	Generalizing the Evolution Process	(Zeng et al., 2024)
Data Filtration	Measuring and Filtering by Instruction Complexity	(Zhao et al., 2024), (Lu et al., 2023b)
Data Distillation	Curriculum Learning and Weak-to-Strong RL	(Sun et al., 2024)