Reasoning-Driven Synthetic Data Generation and Evaluation

Anonymous authors
Paper under double-blind review

Abstract

Although many AI applications of interest require specialized multi-modal models, relevant data to train such models is inherently scarce or inaccessible. Filling these gaps with human annotators is prohibitively expensive, error-prone, and time-consuming, leading model builders to increasingly consider synthetic data as a scalable alternative. However, existing synthetic data generation methods often rely on manual prompts, evolutionary algorithms, or extensive seed data from the target distribution – limiting their scalability, explainability, and control. In this paper, we introduce Simula: a novel reasoning-driven framework for data generation and evaluation. It employs a seedless, agentic approach to generate synthetic datasets at scale, allowing users to define desired dataset characteristics through an explainable and controllable process, enabling fine-grained resource allocation. We show the efficacy of our approach on a variety of datasets, rigorously testing both intrinsic and downstream properties. Our work (1) offers guidelines for synthetic data mechanism design, (2) provides insights into generating and evaluating synthetic data at scale, and (3) unlocks new opportunities for developing and deploying AI in domains where data scarcity or privacy concerns are paramount.

1 Introduction

Data availability and access have been central to advances in artificial intelligence research. In recent years, the abundance of highly-diverse internet data enabled the development of increasingly capable generalist models (Gemini et al., 2023; OpenAI et al., 2023; Anthropic, 2024; Touvron et al., 2023). Despite these models' impressive versatility, widespread integration will require them to specialize on novel, uncommon, and privacy-sensitive applications. Unfortunately, specialized data in these areas is often intrinsically scarce or inaccessible, motivating enormous investments by frontier research labs (Paul & Tong, 2024; Wiggers, 2024; Cottier et al., 2025) and the rapid rise of dedicated "data foundries" (Liu, 2025; Vinn & Hu, 2025). However, creating specialized datasets manually is expensive, time-consuming, and error-prone (Chen et al., 2023; Gilardi et al., 2023; Hosking et al., 2024), leading many to consider synthetic data as a promising, scalable alternative (Singh et al., 2024a; Abdin et al., 2024; Guo et al., 2025). Nevertheless, how to best balance the various desiderata of synthetic data generation at scale remains an open question.

Beyond limiting AI's progress to data-rich domains, e.g., coding and creative writing, the field's reliance on real-world data imposes significant structural, safety, and operational costs. The high cost of manual data collection creates an economic threshold that marginalizes contributions from the broader research community (Chen et al., 2023). This dependency also enforces a reactive approach to safety; models can only be hardened against rare edge cases after failures are observed in deployment, a practice untenable for safety-critical systems. Operationally, the static nature of real-world data slows down development cycles, in stark contrast to the programmable workflows that a synthetic-first approach enables. Furthermore, the "black box" nature of real data makes mitigating societal biases and untangling ownership rights intractable issues that are addressable by design when the data generation process is fully controlled (Bolukbasi et al., 2016; Buolamwini & Gebru, 2018).

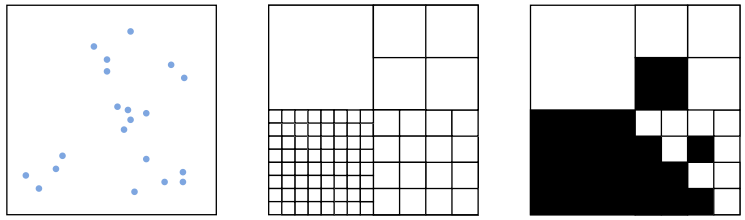
Using synthetic data to solve the specialized data bottleneck faces three distinct challenges concerning (i) the definition of "good" synthetic data, (ii) designing a mechanism that meets realistic requirements, and (iii) conducting generalizable evaluations. Generally, "good" synthetic data is discussed along the axes of

"*quality, diversity, and complexity*" (Havrilla et al., 2024). As a running example, suppose our goal is to generate a dataset of cat images to train a specialized classifier. Instead of indicating the usefulness of data (Swayamdipta et al., 2020; Marion et al., 2023; Tirumala et al., 2023), "quality" commonly refers to how well data points fit specific requirements. For instance, if the intention is to generate an image of a "red cat", does the resulting image have a cat in it, and, is that cat indeed red? Meanwhile "complexity" can refer to how confusing, uncommon, or elaborate a specific data point is (Ethayarajh et al., 2022; Shao et al., 2023; Fu et al., 2023) but is often equated with the relative concept of "difficulty". This, in turn, begs the question: difficult for whom and under what circumstances? In the case of our red-cat image, a complex example might be a partially obscured cat or one lying in the shadows. Finally, "diversity" offers both a global and local interpretation: does the generated data globally cover the main factors of interest, and does it locally exhibit sufficient variation of specific factors? In our case, global factors could be represented by "cat breed", "color", and "environment" and the various members of these factors. Locally, we would be interested in the number of different interpretations of a combination of factors, e.g., different takes on a red cat lying on a couch. Most work on synthetic data focuses on how to best interpret "good" data, often only optimizing for a subset of the above desiderata (Havrilla et al., 2024).

The requirements of the second synthetic data challenge are less commonly discussed, namely, what are the *mechanism* criteria for generating good data? We posit that realistic deployment conditions require a mechanism that works at scale while maintaining explainability and control. Scalability is a non-negotiable: scaling-law research has repeatedly vindicated the bitter lesson that *more* good data improves outcomes (Kaplan et al., 2020; Hoffmann et al., 2022). Consequently, this quickly disqualifies methods that rely on elaborate custom prompts (Gupta et al., 2024; Xu et al., 2024b; Yu et al., 2023). Legal and operational requirements further dictate the need for transparent "audit traces", explaining the ingredients used to generate specific data points. This is problematic for methods based on stochastic, evolutionary algorithms (Mehrotra et al., 2024; Fernando et al., 2024). As will be shown in this work, not all data problems are created equal; some benefit from more quality control and others from higher complexity or local diversity. Optimal resource allocation thus requires fine-grained control over the data generation process.

The third challenge concerns evaluation: how well does the generated data intrinsically approximate a target distribution and how effective is it to train a model for a specialized downstream use case? For the latter, the most common approach is to generate data that simulates an existing benchmark, train a model on the synthetic data, and measure the trained model's performance on the benchmark's hold-out test set. However, benchmark datasets are incomplete samples of a target domain, often contain incorrect labels (Northcutt et al., 2021), are unbalanced (van Breugel et al., 2024), "under annotated", e.g., they lack multi-level annotations and details of interest, and risk being leaked into pre-training datasets. Hence, naively optimizing the data generative process toward a benchmark dataset can be deceiving. It can lead to overfitting, where synthetic data is optimized to fit the specific quirks of a benchmark dataset, which may not be representative of the target domain. Care should thus be taken to evaluate synthetic data approaches on diverse, *recent* datasets to improve generalization and minimize potential upstream contagion.

In this work, we propose Simula, a framework for synthetic data generation and evaluation that prioritizes transparency, fine-grained control, and scalability. Unlike many existing efforts that rely on opaque, seed-dependent processes, Simula employs a "reasoning-first" methodology, constructing each data point from first principles. This approach ensures that Simula's generation capabilities improve naturally with the reasoning capabilities of its underlying models, making the system inherently future-proof. Our three-stage pipeline first maps the conceptual space of the target domain to ensure coverage, then uses iterative agentic refinements to populate it with diverse and complex examples, and finally uses a dual-critic filtering stage to enhance quality. We rigorously test the core reasoning assumptions underlying our approach and demonstrate its efficacy on a series of carefully designed experiments using a variety of datasets. Our contributions are as follows: (1) a new reasoning-first framework for synthetic data generation capable of creating explainable and controllable data at scale; (2) a comprehensive empirical analysis connecting intrinsic properties of generated data (e.g., complexity, diversity) to extrinsic downstream model performance, offering novel, actionable evaluation metrics and insights into scaling synthetic data; (3) a set of pragmatic principles that establish mechanism design – how data is generated – as a distinct research axis, orthogonal to the traditional focus on what constitutes "good data."



(a) (b) (c)

Figure 1: **Synthetic Coverage Examples.** (a) Characterizes "random" sampling behavior, with no notion of a global coverage space. This often results in samples clustered around semantic modes; (b) Represents perfect global planning at increasing granularity; and (c) shows global planning with progressive coverage loss.

2 Simula: A Reasoning-First Framework for Data Generation and Evaluation

Suppose we want to create a dataset with the description $y :=$ *"A dataset of stories about cats."* Due to the under-specification of y, it is infeasible to exhaustively describe the space of all datasets $\mathcal{Y}$, that fit this description. This is problematic as it prevents us from developing an actionable notion of coverage, i.e., given a dataset $\mathcal{D}_y \sim \mathcal{Y}$, what area of $\mathcal{Y}$ does it represent?

2.1 Using Taxonomies to Capture Dataset Coverage

To formulate a first-order approximation of $\mathcal{Y}$, we start by disentangling our target dataset into its prime factors of variation.[1] For example, a dataset that fits description y might consist of data points covering "*cat type*," "*story format*," and "*intended audience*." In Simula, a multi-modal model (M3) is used to propose factors f_i based on a set of human-provided instructions, e.g., a description like y, and/or a sample of existing data $\mathcal{S}$. These factors can be accepted or rejected by an M3 (or a Human). Next, an M3 is used to expand each f_i breadth-first into taxonomies, $\mathcal{T}_i$, of a (user-) specified depth d_i:

$$\text{M3}(y, \mathcal{S}, (d_0, f_0), \cdots, (d_K, f_K)) = \{\mathcal{T}_i\}_{i=0}^{K} = \mathcal{T}^y \tag{1}$$

Using taxonomies provides granular explainability and control of $\mathcal{Y}$ compared to random sampling (Figure 1.a). Intuitively, as we increase the number of factors and taxonomy depths, we sharpen our coverage control (Figure 1.b). However, this granularity comes at a potential cost: with every taxonomy expansion we risk "missing" nodes of interest, resulting in the progressive coverage loss depicted in Figure 1.c.

To mitigate potential coverage loss resulting from missing nodes, we generate $\mathcal{T}_i$ by alternating between three steps (full algorithm in App. B.4): (1) Given a node, its ancestors and its siblings, an M3 is prompted N times to propose an initial set of children nodes. This sampling strategy is inspired by the "Best-of-N" literature to increase the proposal distribution and cover edge cases (Brown et al., 2024). (2) In a separate call, an M3 is prompted to act as a critic, refining the initial proposals by adding, removing, merging, or editing nodes to improve their completeness, soundness, and specificity, leveraging the generator-critic gap (Huang et al., 2024). Optionally, (3) after generating all nodes of a specific level, an M3 is prompted to generate a "plan" for the next level. This enables consistent and fast parallel generation by ensuring a similar degree of granularity at different node expansions on the same level across independent predictions. At each step, the M3 also has access to the user-provided input y, and/or a sample $\mathcal{S}$ from the target distribution.

2.2 Controlled, Explainable, and Scalable Data Synthesis with Agentic Steps

Having obtained the taxonomies $\mathcal{T}_i$, we can generate synthetic datasets that fit our requirements through two phases: taxonomic sampling (Figure 2.c) and agentic refinement (Figure 2.d-e). Full algorithm is in App. C.

Sampling Strategies to Control Global Diversity. An M3 first formulates a plan composed of sampling strategies. A strategy defines which taxonomies are sampled together and with what weight, preventing illogical combinations by grouping compatible subsets. For instance, a "children's content" strategy might

[1] Note that perfect disentanglement is of course not always possible (Locatello et al., 2019).

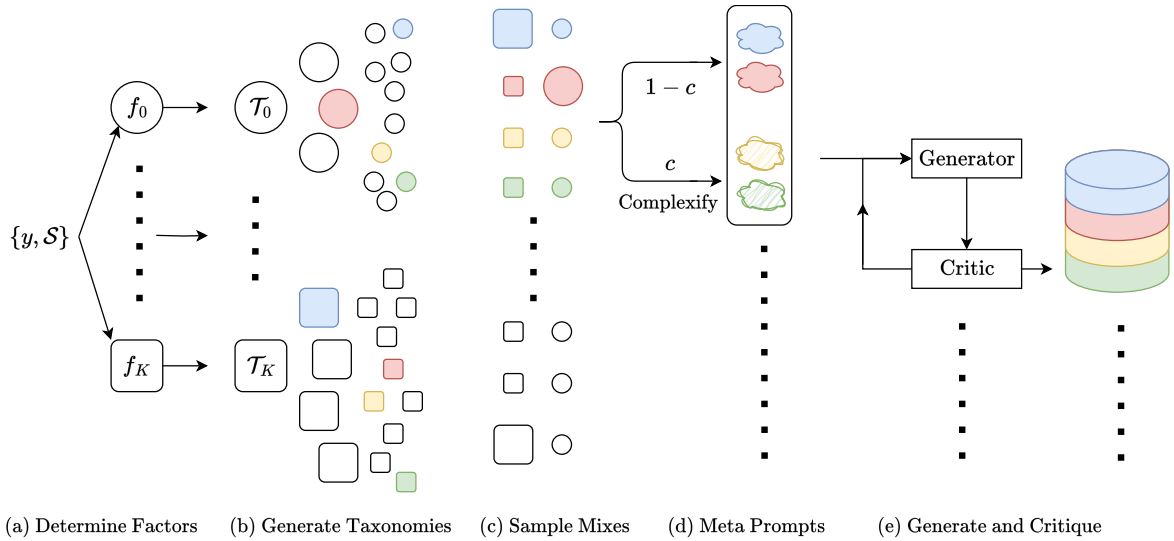


Figure 2: **Schematic of Simula Framework.** Given user instructions y and/or a data sample $\mathcal{S}$, we first (a) determine factors of interest f_i, which (b) are expanded into taxonomies $\mathcal{T}_i$. Next, (c) nodes of $\mathcal{T}_i$ are sampled to obtain mixes, and (d) turned into "meta prompts". A user-defined fraction, c, of meta prompts is "complexified." (e) Finally meta prompts are used to generate data proposals by a `Generator`, and iteratively refined using a `Critic` step.

combine taxonomies for simple topics and age-appropriate formats while excluding those for mature themes, thus preventing the generation of a horror novel for toddlers. By weighting different strategies, the framework can control the proportion of generated data for each target group. Given a strategy, the framework samples nodes from the corresponding taxonomies $\mathcal{T}_i$. These sampled nodes can be thought of as data point "requirements," that, along with the original dataset instructions y, then guide an M3 to construct one or more "meta prompts". For instance, M3(y, {*house cat, poem, travel enthusiast*}), becomes "*Compose an exciting haiku about a house cat who goes on an adventure.*" Finally, these meta prompts are passed to an M3 to generate data output proposals.

Optimizing Local Diversity and Complexity. Suppose we want to construct a dataset of size $N = 100$, and our factor and strategy selection has yielded $V = 200$ unique node-sets. Since $N < V$, our sampling budget allows for at most 100 unique node-sets with a single meta prompt each, resulting in a global coverage ratio of $100/200 = 0.5$. Conversely, for $N > V$, e.g., $N = 800$, we can sample up to four meta prompts for each node-set and cover the entire space. As the number of meta prompts per node-set grows, we gradually increase *local* diversity. However, for larger values of N/V, independently generating meta prompts from fixed requirements can lead to mode collapse, i.e., increasingly similar meta prompts. This is mitigated by generating multiple meta prompts simultaneously, then sub-sampling the required fraction. Next, we perform *complexification* on a fraction of the samples, by prompting the M3 to increase the complexity of the generated meta prompts and outputs while maintaining our other requirements. Optimizing local diversity and complexity this way works well for smaller sample sizes, but degrades as N/V grows very large. Instead, for larger ratios, Simula can be configured to sequentially generate more diverse or complex meta prompts with previous attempts in context, allowing an M3 to reflect on previous generations.

Enhancing Sample Quality with Critics. The system performs a series of agentic refinement steps to optimize sample output quality. It starts with point-wise checks to ensure the generated samples pass the specified semantic and syntactic requirements coming from the taxonomy nodes. This involves letting the M3 "critique" the generated samples by providing the meta prompts used for generation and requesting an explanation and a binary verdict. Given the generated sample for the adventurous house-cat haiku above, the M3 would check if the cat in the story is indeed a house cat, if the output is a haiku, and if adventures were had. For tasks requiring outputs with a defined notion of correctness (e.g., classification or multiple-choice questions), the system employs an additional "*double-critic*" step, which independently assesses correctness and incorrectness to mitigate sycophancy bias (Sharma et al., 2024) (More on this in Section 3.1). Following

these "*critic refinement*" steps, if the M3 responds with a negative verdict, the system either rejects the sample or applies automated modifications based on the explanation and repeats the critique step.

2.3 Evaluating and Comparing Dataset Properties

The previous section described how taxonomies and agentic steps can be combined to generate synthetic datasets optimized for diversity, complexity, and quality. Beyond generation, we are interested in evaluating the properties of datasets – both synthetic or real – and comparing them. We therefore describe two approaches that use agentic reasoning and taxonomies to evaluate and compare complexity and diversity.

Reasoning about Complexity using Calibrated Attribute Scoring. Different levels of data complexity are desired depending on the target task. Ideally, one would like to assign "complexity scores" to individual data points, enabling control over sample complexity as well as comparisons between synthetic and real data. This presents several challenges: (i) synthetic data generation is unsupervised, (ii) real data is generally not annotated for complexity, and (iii) "complexity" is a relative concept that depends on the domain and task at hand. To overcome these issues we propose a "batch-wise" evaluation approach based on reasoning: Firstly, batches of data points are sampled such that each data point appears K times. Secondly, an M3 assigns scores to each sample in each batch reflecting their *complexity* using the dataset description y (Algorithm in App. E.3). By providing a batch to score instead of individual points, the model can calibrate its outputs among the different points to reduce noise resulting from per-sample overconfidence (Zheng et al., 2023; Xiong et al., 2024). We further improve our scoring signal by breaking batch-specific score assignments into pairwise comparisons and computing Elo scores (Elo & Sloan, 1978). The resulting Elo scores enable complexity comparisons of data points across different datasets and can be used to control a sample's complexity.

Using Taxonomies to Curate Dataset Coverage. The importance of data selection during both M3 training and inference time is emphasized by a growing body of research (Swayamdipta et al., 2020; Marion et al., 2023; Ye et al., 2024; Xia et al., 2024; Hu et al., 2024; Hübotter et al., 2024), *inter alia*. Access to training data that accurately reflects test-time conditions and minimizes biases further is essential to assess model performance, fairness, and preparedness. Despite their vital role, most datasets are sparsely labeled using high-level semantic descriptions, e.g., a "*math*" question, a "*harmful*" comment, a "*cat*" image, etc., or not labeled at all. This complicates efforts to curate optimal corpora, allocate resources (Qian et al., 2024), and catch potential misalignment between train and test sets (van Breugel et al., 2024). Using taxonomies offers a way forward not only for generating synthetic data, but also for better understanding existing data. Given a dataset, we can generate or use existing taxonomies that define the conceptual space of interest. We then query an M3 to perform taxonomy assignments for each data point by prompting it with the full taxonomy in context, linking each item to the most relevant node. This process yields a detailed map of the dataset's composition, from which various coverage metrics can be derived. In our evaluation, for instance, we use a "Level Ratio Coverage" metric, which calculates the proportion of unique nodes covered by a dataset at each taxonomy level. This provides a fine-grained, actionable view into a dataset's coverage, highlighting both well-represented areas and potential gaps to fill.

3 Experimental Setup

3.1 Verifying Core Reasoning Components

Simula primarily relies on three core assumptions about M3 reasoning capabilities: M3s can (i) generate high-quality taxonomies; (ii) function as effective "critics" of their own outputs; and (iii) distinguish between more and less complex examples. Below, we show how we test each of these assumptions using Gemini 2.5 Flash (non-thinking) (Gemini et al., 2024).

Can M3s Generate High-quality Taxonomies? Evaluating the quality of taxonomies is inherently challenging due to the lack of standardized criteria and methods (Szopinski et al., 2020; Kaplan et al., 2022; Kundisch et al., 2021). We differentiate between *grounded* taxonomies, e.g., phylogenetic trees, and *conceptual* ones, e.g., types of harmfulness. Given an expert taxonomy, $\mathcal{T}_E$, we compare to an M3-generated taxonomy

for the same topic, $\mathcal{T}_{\text{M3}}$. Structurally, we care about completeness (does $\mathcal{T}_{\text{M3}}$ cover $\mathcal{T}_E$?), soundness (does $\mathcal{T}_{\text{M3}}$ contain irrelevant or unnecessary nodes?), and novelty (does $\mathcal{T}_{\text{M3}}$ contain relevant nodes *not* in $\mathcal{T}_E$?). We evaluate both the Simula generator-critic approach and 0-shot expansion on six real-world taxonomies. Additional experimental details are available in Appendix B.

Are M3s Effective Critics? We test the ability of our "double critic"—which independently queries if an answer is *correct* and *incorrect*—to verify answer correctness as follows: In the *Controlled Setting*, we take reference tasks with correct answers (and if applicable, explanations), $\mathcal{D}_{\text{true}} = \{(x_i, y_i)\}_{i=0}^{N}$, and create a corrupted copy $\mathcal{D}_{\text{corrupt}}$ by prompting an M3 to subtly change y_i while keeping x_i fixed. After performing this causal intervention on answer correctness (Pearl, 1994), we evaluate the double critic on both datasets. Note that, for effective critic-rejection sampling, we need the likelihood of accepting a correct answer to be high and accepting a corrupted answer to be low. In the *Empirical Setting*, we test if critic capabilities *transport* (Pearl & Bareinboim, 2011) to the model's own outputs. We prompt an M3 to generate reasoning traces and answers for the benchmark tasks covering free-form math problems (MATH, Hendrycks et al. (2021)) and multiple-choice questions in selected languages (Global MMLU, Singh et al. (2024b)). We apply double-critic rejection sampling, measuring the rejection rate and the empirical change in accuracy. We note that this differs from the full Simula pipeline where the M3 generates *both* questions and answers. The net impact of the critic in this fully synthetic scenario is tested through our downstream evaluations (Sec. 4.3).

Can M3s Distinguish Data Complexity? We first investigate whether complexity rankings computed using our calibrated scoring method align with human-provided complexity labels. Second, we examine whether the model's complexity scores align with its own generation and verification capabilities, i.e., we study whether its performance declines on problems it rates as more complex. For this analysis, we again use the MATH dataset, which contains explicit human-annotated complexity ratings (1-5), and selected Global MMLU subjects, where education levels (elementary, high school, and college) serve as a proxy for complexity.

3.2 System Versions and Datasets

System Versions. Table 1 shows an overview of the different Simula components evaluated. To ensure a consistent basis for comparison, we first generate a set of fine-grained taxonomies for each task. We then evaluate system configurations by ablating different components: For the **Baseline** configuration, top-level nodes are sampled from the relevant taxonomies; each mix of nodes is then turned into a data point by the M3. The **Local** component also uses top-level nodes but incorporates agentic refinement by converting them to "meta prompts" and applying "complexification" to a fraction $c = 0.5$ of these. The **Global** component leverages the full taxonomic depth by sampling from nodes at all levels, but omits meta-prompting. Finally, we evaluate a version combining both **Local + Global** diversification, and another that adds a final **Critiquing** step to enhance quality.

Table 1: **System Versions.** We evaluate the efficacy and additive properties of different system components.

	taxonomy depth	meta prompting	critic steps
Baseline	1	×	×
Local	1	✓	×
Global	>1	×	×
Local + Global	>1	✓	×
Local + Global + Critique (i.e., Full System)	>1	✓	✓

Datasets. We chose datasets that cover a spectrum of niche, recent datasets that are not commonly optimized for, as well as common datasets whose domains are optimized for in popular benchmarks. For the former, focused on specialized tasks, we chose two datasets from the Cyber Threat Intelligence Benchmark (CTIBench) (Alam et al., 2024): **CTI-MCQ**, a multiple-choice question dataset with four choices and a single answer to assess M3s' understanding of CTI standards, threats, and mitigation (2.5k test questions); and **CTI-RCM**, a task requiring the open-ended generation of a Common Weakness Enumeration (CWE) category based on a Common Vulnerabilities and Exposures (CVE) description (1k test CVEs). We also

selected a dataset from the legal domain, **LEXam** (Fan et al., 2025), which consists of 1.66k multiple-choice questions from Swiss, EU, and international law examinations in both English and German.

To cover popular domains, we selected two well-known datasets: **GSM8k** (Cobbe et al., 2021), high-quality, linguistically diverse grade-school math problems requiring multi-step reasoning (1.32k test problems); and a subset of **Global MMLU** (Singh et al., 2024b), a multilingual, multiple-choice question dataset with four choices and a single answer. To keep the data generation manageable, we selected a subset across Math, Computer Science, and Physics in English (High-resource), Korean (Mid-resource), and Nepali (Low-resource), for a total of 1.44k test questions per language. We applied a post-filtering process for the generated datasets, including de-duplication and decontamination against the real test sets (based on 13-gram overlap with a Jaccard threshold of 0.8). We ended up with 512k unique data points for each of the above datasets.

3.3 Intrinsic Metrics

We compute intrinsic metrics on a sample of 1k data points for each dataset and each system version.

Embedding Diversity and Taxonomic Coverage. Following Yu et al. (2023), we evaluate diversity by transforming datapoints into an embedding space and measuring cosine distances. For global diversity we report the average dataset-wide, pairwise cosine distance. To evaluate local diversity, we first group data points by taking the $k = 10$ nearest neighbors to each data point in embedding space to ensure semantically meaningful clusters. We then take the average pairwise cosine distance across these clusters. For both, a higher cosine distance indicates higher diversity. We compute the embeddings using the model from Lee et al. (2024b). In practice, cosine distance scores are coarse metrics that do not provide actionable insights to improve diversity. We therefore also evaluate each dataset's taxonomic coverage using the assignment-based methodology described in 2.3, which provides a more structured and actionable measure of how comprehensively data covers a conceptual domain.

Complexity. Our goals for intrinsic complexity evaluation are twofold: First, we want to test if the different Simula components can effectively control the complexity distribution of a synthetic dataset. Secondly, we want to compare the complexity of synthetic datasets to the "real" reference datasets. After generating synthetic datasets using the different systems, we follow the calibrated complexity scoring method outlined in Section 2.3, mixing together the various synthetic datasets with the real data.

3.4 Downstream Metrics

For all downstream experiments we use Gemma 3 4B as our student model (Gemma et al., 2024) and Gemini 2.5-Flash (non-thinking) as our teacher model. For datasets covering domains that are commonly optimized for such as GSM8k and Global MMLU, we use the pre-trained (PT) version as the starting point. For the more niche, recent datasets, CTI-MCQ, CTI-RCM, and LEXam, we use the instruction-tuned (IT) version.

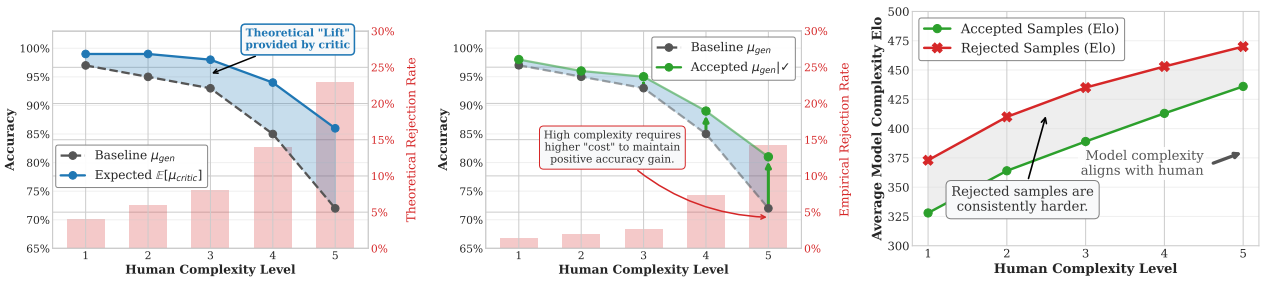
We perform 10 iterations of LoRA finetuning (Hu et al., 2022) with different seeds per configuration and dataset size, reporting mean accuracy for each data size and 95% confidence intervals computed as the standard error of the mean across seeds (see App. F for our hyperparameter optimization process).

To isolate the impact of complexity on downstream metrics, we split datasets produced by the full system into three subsets. To avoid confounding missing complex concepts with complexification per concept, we sample by complexity score per taxonomy node. This gives a *Low Complexity* split (bottom 40% of the data), a *High Complexity* split (top 40%), and an *All Complexity* split (a random subsample of the previous two).

4 Results

4.1 Results for Core Reasoning Components

M3s Can Generate Good Taxonomies. Table 2 shows Simula taxonomies approximately cover those created by human experts. For conceptual taxonomies, almost all generated nodes are sound, with many



(a) **Controlled Setting** (b) **Empirical Setting** (c) **Complexity Analysis**

Figure 3: **Double-Critic Rejection Sampling on MATH.** (a) We establish the theoretical lift in the controlled setting, noting that rejection costs increase with task complexity. (b) Critic capabilities transport to the empirical setting but lose some effectiveness. (c) Calibrated Elo scores show model-human alignment. Stratified by complexity level, rejected samples are consistently assigned higher model complexity.

novel expansions resulting in increased total coverage. Note that we are less concerned with "over" coverage, as this ensures that most edge cases are present. We can always reduce the depth of a taxonomy if we believe it to be too granular. These results support our approach of approximating a global coverage space using taxonomies. Appendix B contains further details and analyses on evaluated taxonomies, as well as limitations and examples of synthetic taxonomies.

Table 2: **Taxonomy evaluation.** Note that novelty and total coverage are only reported for Conceptual taxonomies, since we assume that grounded taxonomies should have no room for further extensions.

	Grounded		Conceptual			
	Completeness	Soundness	Completeness	Soundness	Novelty	Coverage
Simula	**0.74**	**0.75**	**0.78**	**0.97**	**0.94**	**1.72**
0-Shot	0.52	0.70	0.50	**0.97**	0.32	0.83

Double-Critic Rejection Sampling Enhances Quality. Figure 3 summarizes our double-critic evaluation on the MATH dataset, where μ_{gen} is the model's generative baseline accuracy. In the *Controlled Setting* (Figure 3a), we measure the probabilities of the critic accepting correct, $p(y)$, and incorrect, $p(y^{\text{corrupt}})$, answers. The accepted proportion is defined as $|\mathcal{D}_{\text{accept}}| = \mu_{\text{gen}} \cdot p(y) + (1 - \mu_{\text{gen}}) \cdot p(y^{\text{corrupt}})$. The expected accuracy of these items is $\mathbb{E}[\mu_{\text{critic}}] = \mu_{\text{gen}} \cdot p(y)/|\mathcal{D}_{\text{accept}}|$. We observe a consistent theoretical "lift" ($\mathbb{E}[\mu_{\text{critic}}] > \mu_{\text{gen}}$). As human-assigned complexity increases, maintaining this lift requires a higher "cost" in rejection rate ($|\mathcal{D}_{\text{reject}}| = 1 - |\mathcal{D}_{\text{accept}}|$). In the *Empirical Setting* (Figure 3b), the critic filters the model's own outputs into accepted and rejected subsets. While still realizing a lift over the baseline, $\mu_{\text{gen}}|\checkmark > \mu_{\text{gen}}$, it is less effective. Again, higher complexity necessitates a higher empirical rejection rate to sustain accuracy gains.

Complexity Scores Align; Rejected Samples Perceived Harder. Figure 3c validates our calibrated scoring method: model-assigned Elo scores align with the humans ones. We also see that stratified by human-assigned complexity, rejected samples have higher Elo scores than accepted ones, indicating the critic systematically filters samples with higher perceived model complexity. Additional results are in App. D, E.

4.2 Intrinsic Metric Results

Intrinsic Diversity. Figure 4 shows results for the different diversity metrics. Starting with the embeddings-based diversity metrics (top and middle), we first note the Global diversification component is crucial to increase the dataset-wide embedding-based diversity (top). Second, the Local component improves over the baseline; hence meta-prompting effectively increases the diversity of the k nearest points (middle). Third, the Local and Global diversification components generally have an additive effect for embedding-based metrics.

While embedding-based diversity of real data can be both higher or lower than the synthetic variants, when it comes to taxonomic coverage, we observe a clear pattern; the full Simula system exhibits higher coverage

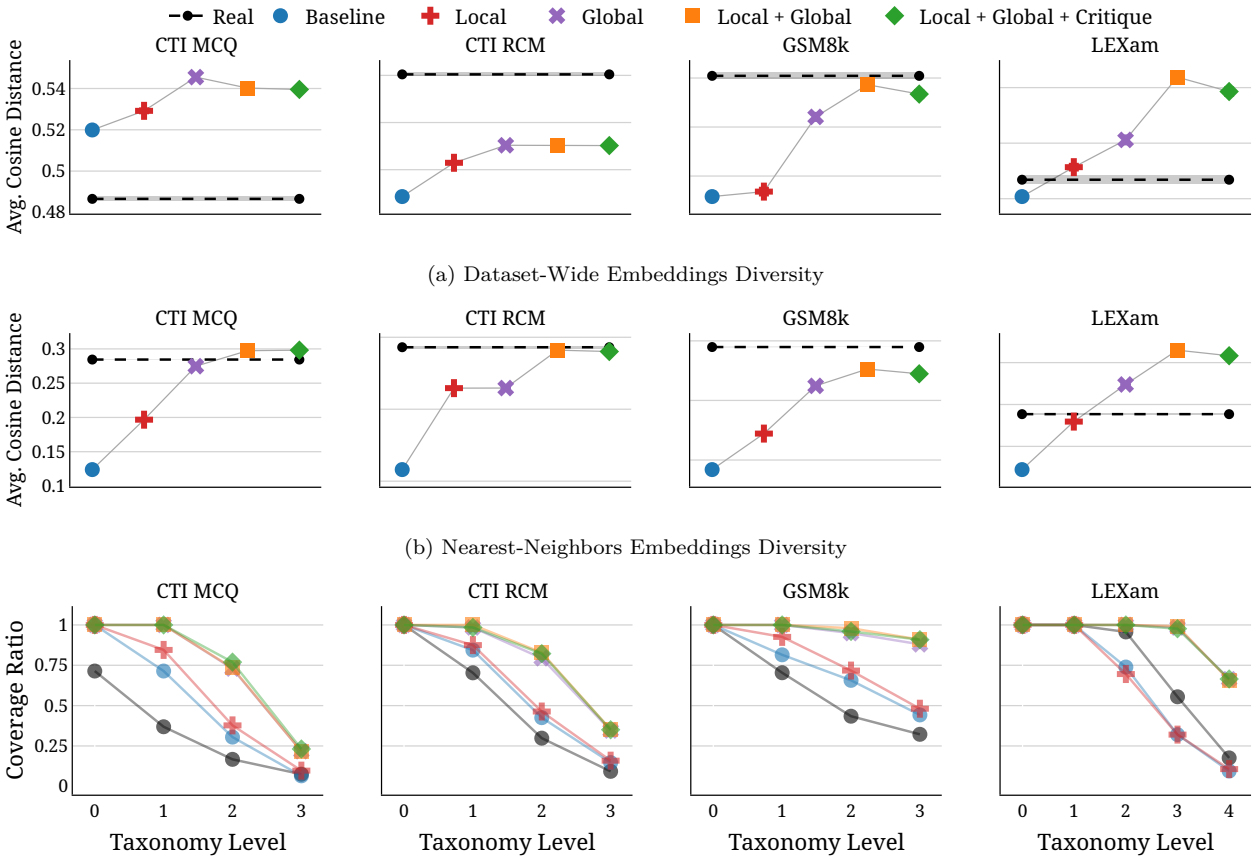


Figure 4: **Intrinsic Diversity Metrics.** We display dataset-wide (top) and nearest-neighbors (middle) embedding-based diversity, and taxonomic coverage (bottom). We note that Global diversification is crucial for increasing dataset-wide diversity, and that Local and Global diversification generally have an additive effect. We further note that while real data can be more or less diverse according to embedding-based metrics, it almost always covers less of the target domain than Simula variants on a taxonomy basis.

as the taxonomy levels increase. Assuming our taxonomies describe comprehensive coverage of the target domain, this indicates that the real data misses large subsets of interest. Finally, critiquing does not appear to affect diversity in a significant way, almost always overlapping with the variant without critic steps.

Intrinsic Complexity. We display density plots of our calibrated complexity evaluation in Figure 5. It can be observed that the full Simula system is capable of covering the entire complexity range of real data, often even allowing for more complex data points. We also note that Local and Global components generally have an additive relation, providing different types of complexity. The different datasets have distinctly different complexity profiles, with the first three panels covering a wide range while LEXam is much more concentrated.[2] Furthermore, while increased diversity and critiquing can reduce complexity (GSM8k), as we will see in the next section, this does not necessarily have an adverse effect on downstream performance.

4.3 Downstream Results

In Figure 6, we show how downstream performance evolves per dataset across data sizes and system versions. Results for additional Global MMLU subsets are available in App. F.2.

[2]This aligns with the teacher model's poor performance on LEXam, as shown in Section 4.3.

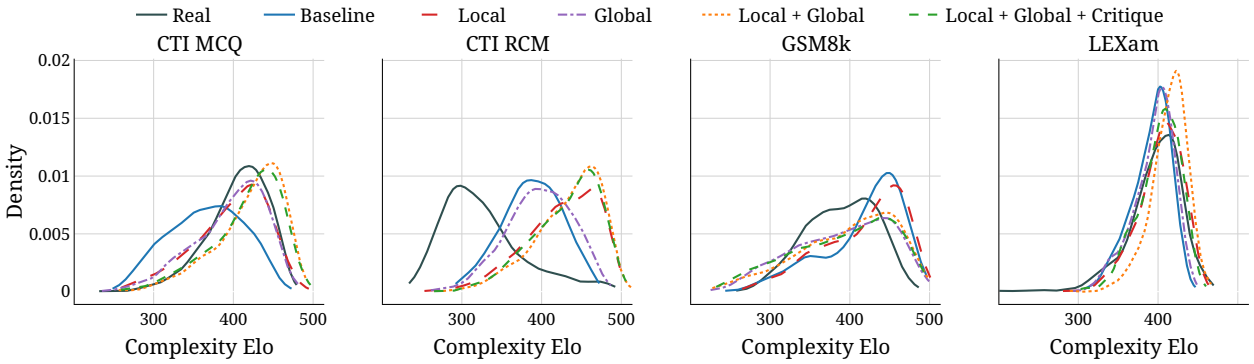


Figure 5: **Complexity Elo Distribution of Synthetic and Real Data.** We display density plots of the complexity Elo rankings for the various system versions on four datasets. We note that synthetic data can cover the entire complexity range of all datasets and that Local and Global components are generally additive, i.e., they account for different types of complexity.

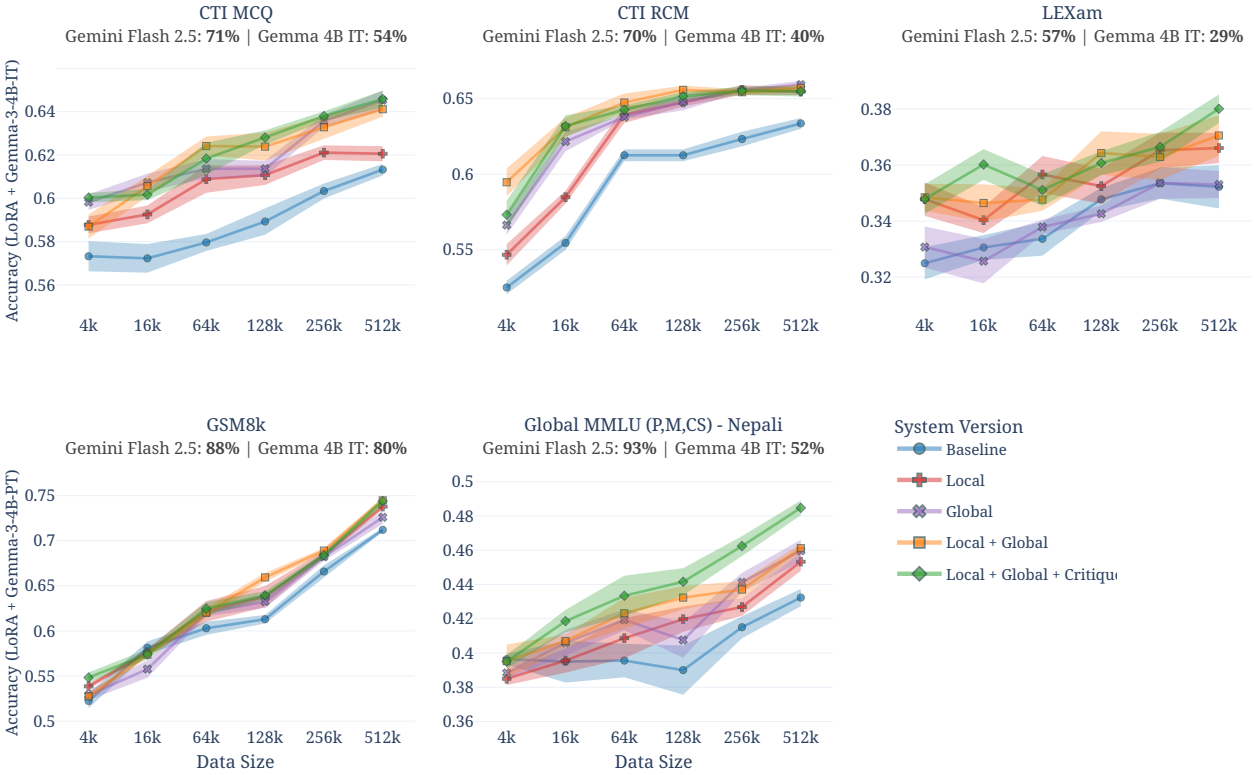


Figure 6: **Downstream Performance on Different Datasets.** We report mean accuracy with 95% CI for different system versions. We note that while the contribution of different components to performance depends on the dataset and data size, the full Simula system always reaches the best result.

Simultaneously Optimizing all Data Axes Never Hurts. We first observe that using the full Simula system is almost always the dominant strategy across all datasets and data sizes. This in contrast to the Baseline, which scales the worst, indicating that the data-scaling law is a function of the data properties, not size alone. This also means that, without strong priors, using the full system is a good strategy.

Student-Teacher Gap Impacts Scaling Laws. We see that downstream performance of the student model at different data scales is impacted by the relative difference against the teacher's performance. In the case of CTI RCM, we see that performance saturates at around 128k, after bridging $\frac{65-40}{70-40} \simeq 83\%$ of

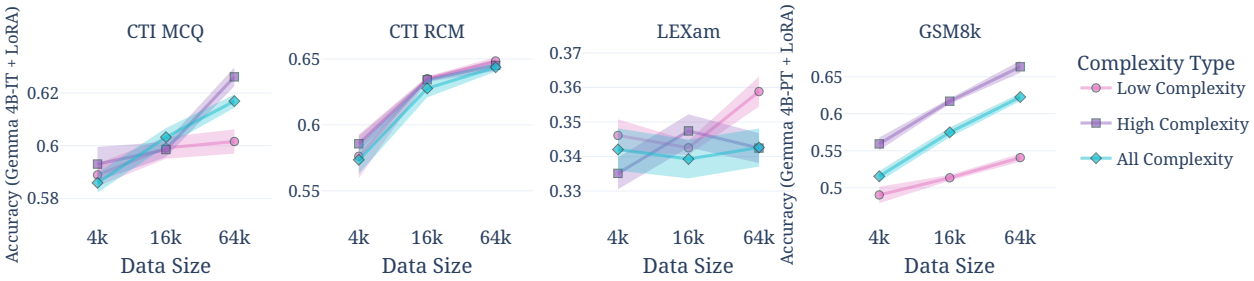


Figure 7: **Downstream Impact of Data Complexity on Mean Accuracy.** For GSM8k and CTI MCQ, higher complexity data improves performance and scaling. Conversely, for LEXam, only the Low Complexity split shows improvement, suggesting that complex data can be detrimental when the teacher model is weak.

the performance gap between the starting point of the student (40%) and the teacher (70%) model. This is not the case with GSM8k, where all versions continue to grow, as the performance of the student model (maximum of 75%) remains sufficiently far from the teacher's performance (88%) on this task.

Combining Global and Local Diversity is Critical. We see how Local and Global diversification in isolation provide suboptimal returns depending on the dataset and data size. In the case of CTI MCQ, Local diversification plateaus before variants containing a Global component. Local diversification also leads to significantly weaker performance in the small data regime for CTI RCM (4k and 16k). On the other hand, Global diversification does not surpass the Baseline in LEXam and scales slightly worse in the case of GSM8k. Combining the two improved performance on all datasets and dataset sizes.

Critiquing Impact is Dataset Dependent. Adding the critiquing step does not hurt performance on any of the studied datasets. In the case of Global MMLU it made a significant improvement, despite the rejected data being limited to only 3%. The critic rejection rate was 2% for CTI MCQ, 9% for CTI RCM, 9% for GSM8k, and 61% for LEXam. For LEXam, the high rejection rate can likely be attributed to the weaker performance of the teacher model (57% accuracy). Even when critic-based rejection sampling does not significantly improve downstream performance, we argue that adding a critic step is still desired. First, ensuring data points follow diversity requirements maintains coverage control. Second, increasing "correctness" of labels promotes causal learning, thus improving model robustness.

Complexity is Crucial in Most Domains but Can Hurt with a Weak Teacher. Figure 7 illustrates the impact of data complexity on mean accuracy across four datasets. The importance of complexity is most evident in the GSM8k dataset. Here, we observe a direct correlation between data complexity and accuracy, with the High Complexity split yielding a 10% accuracy gain over the Low Complexity split at 64k data items. A similar trend is present in the CTI MCQ dataset, where the Low Complexity split shows negligible accuracy scaling with increasing data size. Conversely, for the LEXam dataset, only the Low Complexity split demonstrates improved performance with scale. We hypothesize that this is a consequence of weaker teacher performance, which leads to diminished accuracy on more complex data.

5 Discussion

Synthetic Data Generation Has No Single Optimal Solution. "Data" is a frozen reflection of reality as it was or could be. No wonder then, that there is no single "optimal" way to generate an infinite number of possibilities. Our extensive experiments show that the impact of different data properties, e.g., complexity and diversity, depends on the target domain, the model, use case, scale, and likely many other factors. Instead of looking for silver bullets, we should thus design our synthetic data systems to be as flexible as the worlds we intend to capture. With Simula we offer a system than maintains explainability and control at scale, giving practitioners the tools to customize synthetic data to fit their unique requirements.

Synthetic Data Evaluation is a Multi-faceted Challenge. The evaluation of synthetic data is fundamentally challenging due to the ambiguity of its core objectives, the coarse level of existing metrics, and

its disconnect from practical context. Key properties describing "good data" are ambiguously defined and inherently entangled. For instance, one could argue that covering rare domain instances falls under "diversity." However, one could just as easily attribute these to "complexity." Similarly, the challenge of generating samples faithful to many – possibly conflicting – requirements falls as much under "quality" as the other two axes. One could thus argue that an ideal evaluation target does not exist; only context-specific trade-offs. While established intrinsic evaluation metrics like embedding-based cosine distance provide a high-level signal, they fail to provide actionable insights or comparisons. By introducing alternatives like node-level taxonomic coverage and reasoning-based complexity scores, we strive to make intrinsic evaluation more pragmatic. Ironically, the very mechanisms that create a need for synthetic data – novelty, rareness, privacy – make it challenging to robustly assess any synthetic data system against canonical references. Although we evaluated on a mix of niche and popular benchmarks, availability of more high-quality, specialized references would benefit this field.

The Life Cycle Costs of Synthetic Data. In our experiments we compared various Simula variants to a baseline that requires up to 5x fewer inference calls per data point. Ignoring the one-off costs of creating a deeper taxonomy, this implies one could generate five times the amount of data for a fixed generation budget using the baseline system. Nevertheless, this "cheaper" larger dataset is likely more expensive when taking into account the full data life cycle. As we saw in Figure 6, data generated using Simula can be both more efficient and reach higher total performance. Since the costs of training are significantly higher than those of doing inference, smaller datasets are preferred even if they are more expensive to generate. When using synthetic data to evaluate model preparedness, larger datasets naturally require more evaluation steps.

Limitations and Future Directions. This work presented a comprehensive analysis of different factors relevant to synthetic data generation and evaluation. Nevertheless, our study has several limitations worth discussing. We conducted our experiments using a single model family to generate data (Gemini 2.5 Flash). As shown in Section 4.3, the teacher model used in our ablations has a significant performance gap with the student model even after finetuning, making it unlikely to be a bottleneck on downstream performance. Additionally, as each data point in Simula is generated from first principles using reasoning-based components, we expect outputs to naturally improve with the underlying models powering the system.

To isolate the effects of different system components, we fixed the student model size and finetuning method. This approach is grounded in established scaling laws showing that performance improves with better data, regardless of model scale (Kaplan et al., 2020; Hoffmann et al., 2022). While the improvements from using more data can diminish, these reflect diminishing marginal returns due to data redundancy (Muennighoff et al., 2023) or quality (Chen et al., 2025), not a plateau in which larger models cease to benefit from more or better data. We therefore expect our findings to hold across various model scales. Following Schulman & Lab (2025), it is further reasonable to assume that LoRA finetuning approaches full finetuning performance for the evaluated data regimes. Comparing reinforcement learning to supervised finetuning on our target tasks remains an interesting experiment for future work.

Although the experiments presented here focus exclusively on text-based synthetic data, we see no reason why the same system could not be used for other modalities. We leave the exploration of these for future work.

6 Conclusion

Traditionally powered by data abundance, AI progress is at a junction; just as its potential is becoming evident, the specialized data needed to realize it is unlikely to be generated by humans (Villalobos et al., 2024). Synthetic data is primed to play a central role in new AI leaps; enabling the creation of specialist systems, equitable availability, fairness, and preparedness evaluation. Beyond focusing on what it means to generate "good" data, this work emphasizes the importance of aligning *mechanism design* to practical requirements. With Simula, we provide a reasoning-first framework capable of generating explainable and controllable synthetic data at scale. Aided by this controllable framework, our in-depth experiments highlighted just how idiosyncratic the relation between "good" data and downstream performance is. We therefore urge the community to resist searching for "silver bullets," and instead embrace synthetic data's deep context-dependency through controllable, reasoning-driven approaches.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Md Tanvirul Alam, Dipkamal Bhusal, Le Nguyen, and Nidhi Rastogi. Ctibench: A benchmark for evaluating llms in cyber threat intelligence. *Advances in Neural Information Processing Systems*, 37:50805–50825, 2024.

Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

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

Michele Banko, Brendon MacKeen, and Laurie Ray. A unified taxonomy of harmful content. In Seyi Akiwowo, Bertie Vidgen, Vinodkumar Prabhakaran, and Zeerak Waseem (eds.), *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pp. 125–137, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.alw-1.16. URL https://aclanthology.org/2020.alw-1.16/.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

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. *arXiv preprint arXiv:2407.21787*, 2024.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pp. 77–91. PMLR, 2018.

Olaf Bánki, Yury Roskov, Markus Döring, Geoff Ower, Diana Raquel Hernández Robles, Camila Andrea Plata Corredor, Thomas Stjernegaard Jeppesen, Ari Örn, Thomas Pape, Donald Hobern, Stephen Garnett, Holly Little, R. Edward DeWalt, Keping Ma, Joe Miller, Thomas Orrell, Rolf Aalbu, John Abbott, Carlos Aedo, and Others. Catalogue of life, 1 2025. ISSN 2405-8858. URL https://www.checklistbank.org/dataset/307664.

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. *arXiv preprint arXiv:2410.15226*, 2024.

Zhengyu Chen, Siqi Wang, Teng Xiao, Yudong Wang, Shiqi Chen, Xunliang Cai, Junxian He, and Jingang Wang. Revisiting scaling laws for language models: The role of data quality and training strategies. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23881–23899, 2025.

Zhihong Chen, Feng Jiang, Junying Chen, Tiannan Wang, Fei Yu, Guiming Chen, Hongbo Zhang, Juhao Liang, Chen Zhang, Zhiyi Zhang, et al. Phoenix: Democratizing chatgpt across languages. *arXiv preprint arXiv:2304.10453*, 2023.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier AI models, 2025. URL https://arxiv.org/abs/2405.21015.

Gary N. Curtis. Taxonomy of logical fallacies. https://www.fallacyfiles.org/taxonnew.html, 2023. Accessed 2025-01-23.

Tim R. Davidson, Viacheslav Surkov, Veniamin Veselovsky, Giuseppe Russo, Robert West, and Caglar Gulcehre. Self-recognition in language models. In *EMNLP*, 2024a.

Tim Ruben Davidson, Veniamin Veselovsky, Michal Kosinski, and Robert West. Evaluating language model agency through negotiations. In *ICLR*, 2024b.

Arpad E Elo and Sam Sloan. The rating of chessplayers: Past and present, 1978.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *EMNLP-IJCNLP*, 2019.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pp. 5988–6008. PMLR, 2022.

Yu Fan, Jingwei Ni, Jakob Merane, Etienne Salimbeni, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Florian Geering, Oliver Dreyer, et al. Lexam: Benchmarking legal reasoning on 340 law exams. *arXiv preprint arXiv:2505.12864*, 2025.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *ICML*, 2024.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *ICLR*, 2023.

Gemini, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Gemma, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Himanshu Gupta, Kevin Scaria, Ujjwala Anantheswaran, Shreyas Verma, Mihir Parmar, Saurabh Arjun Sawant, Chitta Baral, and Swaroop Mishra. Targen: Targeted data generation with large language models. *COLM*, 2024.

Alex Havrilla, Andrew Dai, Laura O'Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fabrizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, et al. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models. *arXiv preprint arXiv:2412.02980*, 2024.

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. In *NeurIPS*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017.

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, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

Tom Hosking, Phil Blunsom, and Max Bartolo. Human feedback is not gold standard. In *ICLR*, 2024.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W Ma. Most influential subset selection: Challenges, promises, and beyond. *NeurIPS*, 2024.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *NeurIPS Workshop on Mathematics of Modern Machine Learning*, 2024.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of llms. In *NeurIPS Workshop on Mathematics of Modern Machine Learning*, 2024.

International Union of Pure and Applied Chemistry (IUPAC). Periodic table of the elements. https://iupac.org/what-we-do/periodic-table-of-elements/, 2022. Accessed: 2025-01-23; Standard atomic weights current as of 2022-05-04.

Angelika Kaplan, Thomas Kühn, Sebastian Hahner, Niko Benkler, Jan Keim, Dominik Fuchß, Sophie Corallo, and Robert Heinrich. Introducing an evaluation method for taxonomies. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*, pp. 311–316, 2022.

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. *arXiv preprint arXiv:2001.08361*, 2020.

Abhinav Ramesh Kashyap, Thanh-Tung Nguyen, Viktor Schlegel, Stefan Winkler, See-Kiong Ng, and Soujanya Poria. A comprehensive survey of sentence representations: From the bert epoch to the chatgpt era and beyond. *arXiv preprint arXiv:2305.12641*, 2023.

Dennis Kundisch, Jan Muntermann, Anna Maria Oberländer, Daniel Rau, Maximilian Röglinger, Thorsten Schoormann, and Daniel Szopinski. An update for taxonomy designers: methodological guidance from information systems research. *Business & Information Systems Engineering*, pp. 1–19, 2021.

Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In Chris Callison-Burch, Philipp Koehn, Cameron Shaw Fordyce, and Christof Monz (eds.), *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/W07-0734/.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *ICML*, 2024a.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024b.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. LLM2LLM: boosting llms with novel iterative data enhancement. In *ACL*, 2024c.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *CoRR*, 2024a.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*, 2024b.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Phoebe Liu. The ai billionair you've never heard of, September 2025. URL https://www.forbes.com/sites/phoebeliu/2025/09/17/the-ai-billionaire-youve-never-heard-of/.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

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. In *ICLR*, 2024.

Sheng Lu, Shan Chen, Yingya Li, Danielle Bitterman, Guergana Savova, and Iryna Gurevych. Measuring pointwise $\mathcal{V}$-usable information in-context-ly. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15739–15756, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *NeurIPS*, 36, 2024.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. In *NeurIPS Workshop on Attributing Model Behavior at Scale*, 2023.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum S Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. In *NeurIPS*, 2024.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=XccDXrDNLek.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *NeurIPS*, 2024.

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, 2002.

Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146, 2023.

Katie Paul and Anna Tong. Inside big techs underground race to buy AI training data, April 2024. URL https://www.reuters.com/technology/inside-big-techs-underground-race-buy-ai-training-data-2024-04-05/.

Judea Pearl. A probabilistic calculus of actions. In *UAI*, pp. 454–462, 1994.

Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 247–254, 2011.

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. *NeurIPS*, 2021.

Crystal Qian, Michael Xieyang Liu, Emily Reif, Grady Simon, Nada Hussein, Nathan Clement, James Wexler, Carrie J Cai, Michael Terry, and Minsuk Kahng. The evolution of llm adoption in industry data curation practices. *arXiv preprint arXiv:2412.16089*, 2024.

Emily Reif, Crystal Qian, James Wexler, and Minsuk Kahng. Automatic histograms: Leveraging language models for text dataset exploration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024. ISBN 9798400703317. doi: 10.1145/3613905.3650798.

Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Nicolaus Foerster, et al. Rainbow teaming: Open-ended generation of diverse adversarial prompts. In *NerIPS*, 2024.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. In *ICML*, volume 202, pp. 30706–30775. PMLR, 2023.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. In *ICLR*, 2024.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4603–4611. PMLR, 2018.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy 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. *TMLR*, 2024a.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, et al. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. *arXiv preprint arXiv:2412.03304*, 2024b.

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

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*, 2024.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *EMNLP*, 2020.

Daniel Szopinski, Thorsten Schoormann, and Dennis Kundisch. Criteria as a prelude for guiding taxonomy evaluation. In *HICSS*, pp. 1–10, 2020.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*, 2023.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. D4: Improving LLM pretraining via document de-duplication and diversification. In *NeurIPS*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Muhammad Usman, Ricardo Britto, Jürgen Börstler, and Emilia Mendes. Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method. *Information and Software Technology*, 85:43–59, 2017.

Boris van Breugel, Nabeel Seedat, Fergus Imrie, and Mihaela van der Schaar. Can you rely on your model evaluation? improving model evaluation with synthetic test data. *NeurIPS*, 2024.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: will we run out of data? limits of llm scaling based on human-generated data. In *ICML*, 2024.

Milana Vinn and Krystal Hu. Exclusive: Scale AI's bigger rival Surge AI seeks up to $1 billion capital raise, sources say, July 2025. URL https://www.reuters.com/business/scale-ais-bigger-rival-surge-ai-seeks-up-1-billion-capital-raise-sources-say-2025-07-01.

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. Large language models enable few-shot clustering. *TACL*, 12:321–333, 2024.

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. In *ACL*, 2023.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 214–229, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533088. URL https://doi.org/10.1145/3531146.3533088.

Kyle Wiggers. AI training data has a price tag that only big tech can afford, June 2024. URL https://techcrunch.com/2024/06/01/ai-training-data-has-a-price-tag-that-only-big-tech-can-afford.

John & Sons Wiley, Richard V Gaines, Catherine Skinner H., Eugene E. Foord, Brian Mason, Abraham Rosenzweig, and Vandall T. King. Minerals arranged by the new dana classification. https://webmineral.com/danaclass.shtml, 1997-2024. Accessed 2025-01-23.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: selecting influential data for targeted instruction tuning. In *ICLR*, 2024.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *ICLR*, 2024.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *ICLR*, 2024a.

Zerui Xu, Fang Wu, Tianfan Fu, and Yue Zhao. Retrieval-reasoning large language model-based synthetic clinical trial generation. *arXiv preprint arXiv:2410.12476*, 2024b.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *CoRR*, 2024.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. Large language model as attributed training data generator: A tale of diversity and bias. *NeurIPS*, 2023.

Yunhao Zhang and Renée Gosline. Human favoritism, not ai aversion: People's perceptions (and bias) toward generative ai, human experts, and human–gai collaboration in persuasive content generation. *Judgment and Decision Making*, 18:e41, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *NeurIPS*, 2023.

Appendices

A Related Work **21**

A.1 Synthetic dataset evaluation. 21

A.2 Synthetic dataset generation. 22

B Reasoning-driven Taxonomy Generation and Evaluation **23**

B.1 Defining Taxonomy Evaluation Metrics 23

B.2 Taxonomy Evaluation Results 24

B.3 Limitations 25

B.4 Taxonomy Generation Algorithm 26

B.5 Qualitative Examples 27

C Synthetic Data Generation Algorithm **28**

D Evaluating Double-Critic Steps **29**

D.1 Multi-Lingual MMLU 29

E Calibrated Complexity Scoring **29**

E.1 Open Generation: MATH 29

E.2 Multiple-Choice Generations: MMLU Global 29

E.3 Calibrated Complexity Scoring Algorithm 31

F Downstream Experiments **32**

F.1 Training Configuration and Hyperparameters 32

F.2 Additional Evaluation Results 32

A Related Work

A.1 Synthetic dataset evaluation.

Early Evaluation Methods. When evaluating a synthetic dataset, we can differentiate between *comparative* and *intrinsic* evaluation. The former expects the availability of a reference dataset. In the case of "closed" tasks that allow for narrow comparisons, we can directly compare model-generated outputs to reference solutions, e.g., single-word answers, translations, or summaries. Originally, these were done directly on the observed outputs, e.g., by comparing overlapping words or n-grams (Papineni et al., 2002; Lin, 2004; Lavie & Agarwal, 2007), *inter alia*.

Rise of Semantic Metrics. As tasks might have multiple different, but semantically equivalent solutions, evaluations shifted to embedding-based approaches (Patil et al., 2023). Such approaches can even be used in the absence of one-to-one reference mappings, by comparing output distributions on a dataset level (Heusel et al., 2017; Pillutla et al., 2021). Although embedding-based approaches allow for automated evaluation, they can struggle in specialized domains and miss semantic subtleties (Ethayarajh, 2019; Kashyap et al., 2023).

Auto-Verifiable Tasks. Then, we have a class of problems that allow for many trajectories ending in a final solution whose correctness can be automatically verified quickly, e.g., program synthesis and mathematical reasoning (Song et al., 2024), or negotiation games (Davidson et al., 2024b).

The Challenge of Open Tasks. What remains is a large class of "open" tasks, for which there are no well-defined criteria of correctness, e.g., creative writing, advice, and most visual media. For these, we largely rely on the reasoning capabilities of human annotators to provide quality references and to obtain pairwise preference labels (Christiano et al., 2017; Bai et al., 2022). However, creating such datasets manually is expensive, time-consuming, and error-prone (Chen et al., 2023; Gilardi et al., 2023; Hosking et al., 2024).

Emergence of LLM Judges. Recently, the growing capabilities of frontier models have opened up the possibility of model-based reasoning (Lee et al., 2024a; Zheng et al., 2023; Li et al., 2024a; Saunders et al., 2022; Wang et al., 2023; Madaan et al., 2024), *inter alia*. As humans and AI start to prefer AI-generated text (Zhang & Gosline, 2023; Panickssery et al., 2024), and AI prefers text by the strongest AI models (Davidson et al., 2024a), model-based evaluation is set to become increasingly prevalent.

Intrinsic Evaluation Metrics. In the absence of reference data, one must rely on *intrinsic* evaluation. As described in the introduction, we typically focus on the axes of quality, diversity, and complexity. When we treat quality as how well a set of data points meets stated requirements, we are quickly forced to rely on reasoning-based evaluation. Instead, if understood as the utility of a dataset for a specific downstream task, we can directly measure the lift in downstream performance, e.g., through classification accuracy. Diversity is often approximated using pairwise cosine similarity after embedding outputs into a higher dimensional space (Yu et al., 2023; Gupta et al., 2024). Without grouping the data using an appropriate clustering step, e.g., semantic clusters, average statistics are sensitive to outliers and fail to differentiate between global and local diversity. Crucially, such diversity statistics provide few semantic, actionable insights. Attempts to automate complexity scoring range from measuring the length of outputs (Shao et al., 2023), or the relative entropy over output alternatives (Ethayarajh et al., 2022; Lu et al., 2023), to generating a large solution set and using a reward model to estimate correctness (Snell et al., 2024). Reasoning-based approaches instead directly query a model to provide a "difficulty" score (Li et al., 2024a). Because models are generally poorly calibrated (Zheng et al., 2023; Tian et al., 2023; Xiong et al., 2024), and difficulty is a relative concept, such absolute scores can be noisy.

Our Reasoning-Based Approach. Our work continues the trend of incorporating model-based reasoning to evaluate data. Instead of using approximate statistics based on output embeddings, reasoning-based approaches provide explainable traces that can be audited and controlled. Mapping out a global coverage space using taxonomies allows end-users to quickly evaluate if their generated data meets the appropriate global diversity requirements (Sections 2.1, 2.3). We further carefully tested popular assumptions about the use of model-based critics through a series of controlled experiments. On the evaluated datasets, we find that critic-rejection sampling of synthetic outputs consistently succeeds in increasing average sample quality

(Sections 2.2, 3.1, 4.1) across different datasets and complexity levels. We also found that model-assigned complexity scores are promising proxies for human notions of difficulty, and correlate well with models' critic and generation capabilities (Sections 2.3, 4.1, and Appendix E).

A.2 Synthetic dataset generation.

Seed-Based Expansion Methods. Popular synthetic data methods generally only account for a subset of the quality, diversity, and complexity axes (Havrilla et al., 2024). For example, Wang et al. (2023) start with a set of seed examples and iteratively expand them using hand-crafted semantic diversity prompts. Xu et al. (2024a) similarly expand seed examples, but focus on both diversity and complexity. The main quality check performed is to ensure that these expanded examples do not become degenerative. An attempt at increasing global diversity is done by maximizing pairwise cosine similarity through rejection sampling. The authors note that expanding semantic diversity and complexity are both positively correlated with downstream performance after fine-tuning.

Factor Identification Approaches. Reif et al. (2024); Chen et al. (2024); Viswanathan et al. (2024); Lu et al. (2024) use seed examples to automatically detect relevant factors through iterative sampling of the target dataset, after which factors are extracted using reasoning modules. Other approaches side-step the need for seed examples by manually inspecting or reasoning about a target dataset to find globally relevant factors of variation (Yu et al., 2023; Gupta et al., 2024; Samvelyan et al., 2024). They then sample from these factors for conditional generation. Relevant to our framework is work done by Li et al. (2024b), who attempt to generate a single, large taxonomy to cover a variety of topics. Inspired by curriculum-based learning in human education systems, the authors generate a variety of learning modules to implicitly vary complexity.

Leveraging the Critic Gap. Many have by now pointed out the apparent gap between current models' generative and verification capabilities (Huang et al., 2024). This gap allows models to act as critics of their own outputs (Saunders et al., 2022; Madaan et al., 2024) and has been successfully used by many of the above methods (Lee et al., 2024c; Gupta et al., 2024; Samvelyan et al., 2024; Chen et al., 2024), *inter alia*.

Scaling Test-Time Compute. Recent efforts in scaling test-time computation show that models are capable of generating correct outputs even for complex questions, given enough attempts (Song et al., 2024; Brown et al., 2024). Yet, how to best scale such a test-time computation budget depends on the complexity of the particular problem (Snell et al., 2024). The authors suggest that "easier" tasks most benefit from exploiting an existing output attempt through iterative refinement, whereas more "difficult" tasks benefit more from exploring a larger proposal distribution.

Our Orchestration Approach. With Simula, we build on many of the existing insights into the merits of optimizing quality, diversity, and complexity for downstream performance. In contrast to existing methods, we explicitly orchestrate the generative process on a dataset level to increase global control and explainability. We carefully split the data generation process into separate steps, i.e., global diversity, local diversity, complexity, and correctness, allowing end users to tailor datasets to their specific requirements (Section 2). In doing so, it becomes possible to allocate computational resources where they are most desired, e.g., by generating more meta prompts to increase the proposal distribution of complex samples or adding additional critic steps to refine the outputs.

B Reasoning-driven Taxonomy Generation and Evaluation

To address the inherent challenges of assessing taxonomy quality and completeness, we use a critic-model based framework for evaluation. Traditional taxonomy evaluation often relies on manual expert review, which is time-consuming, expensive, and often difficult to attain. Our proposed framework leverages the capabilities of large language models as "critic models" to provide a more automated, scalable, and reproducible evaluation.

B.1 Defining Taxonomy Evaluation Metrics

First, a hierarchical representation for each expert taxonomy ($\mathcal{T}_E$) or model-generated taxonomy ($\mathcal{T}_{\text{M3}}$) is provided, and the critic model classifies each node into the following categories:

- Good and Overlapping: The node is good (well-defined, relevant to its parent node, and fits appropriately within the overall taxonomy) AND overlapping (there is a semantically equivalent node in the other taxonomy which represents the same concept).
- Good and Exclusive: The node is good (well-defined, relevant to its parent node, and fits appropriately within the overall taxonomy) AND NOT overlapping (there is a no semantically equivalent node in the other taxonomy which represents the same concept; this concept appears uniquely in this taxonomy).
- Redundant: The node is a duplicate within its own taxonomy, there is another node in the same taxonomy representing the same concept.
- Bad: The node is irrelevant, poorly defined, misclassified, or otherwise inappropriate for its position in the taxonomy.

Based on the critic model's classifications, we compute several quantitative metrics to evaluate each taxonomy:

- Completeness: This metric measures the extent to which $\mathcal{T}_{\text{M3}}$ covers the concepts present in $\mathcal{T}_E$. The LM critic assesses, for each node (concept) in $\mathcal{T}_E$, whether a semantically equivalent node exists in $\mathcal{T}_{\text{M3}}$. This serves as a measure of coverage and recall, quantified by the ratio (Good and Overlapping) / (Total Good) in $\mathcal{T}_E$. Here, Total Good = Good and Overlapping + Good and Exclusive.
- Soundness: This metric assesses the proportion of relevant and correct nodes within $\mathcal{T}_{\text{M3}}$. The LM critic examines each node in $\mathcal{T}_{\text{M3}}$ to judge its relevance to the topic and whether it constitutes a non-redundant entry. Fewer irrelevant or incorrect nodes result in greater soundness. This serves as a measure of precision, quantified by the ratio (Total Good) / (Total Nodes) in $\mathcal{T}_{\text{M3}}$. Here, Total Good = Good and Overlapping + Good and Exclusive.

It is worth noting here, that there are different taxonomy types in practice. *Grounded* taxonomies are typically revised over time as new empirical evidence is gathered through the scientific method. A recently accepted version in the literature can be considered closer to a ground truth than a conceptual taxonomy, in that it offers less scope for a language model to generate new terms absent new evidence. We use the following grounded taxonomies for evaluation: "*Animal Phylogenetic Classes*" (Bánki et al., 2025), "*Periodic Chemical Elements*" (International Union of Pure and Applied Chemistry (IUPAC), 2022), and "*Mineral Classification*" (Wiley et al., 1997-2024).

Conceptual taxonomies are more subjective than grounded ones; even the definitions and usage of terminology can vary across the literature (Usman et al., 2017; Szopinski et al., 2020; Kundisch et al., 2021; Kaplan et al., 2022). We use the following conceptual taxonomies for evaluation: "*Risk of Language Models*" (Weidinger et al., 2022), "*Online Harmful Content*" (Banko et al., 2020), and "*Logical Fallacies*" (Curtis, 2023)).

A more elaborate description of the grounded and conceptual taxonomies used can be found in B.2.

Because of this subjectivity, we additionally compute the following metrics for conceptual taxonomies:

- Novelty: This metric assesses whether $\mathcal{T}_{\text{M3}}$ contains relevant nodes that are not present in $\mathcal{T}_E$. The LM critic identifies nodes in $\mathcal{T}_{\text{M3}}$ that are not semantically equivalent to any node in $\mathcal{T}_E$ and then judges the relevance of these novel nodes. A higher number of relevant novel nodes indicates greater novelty. We define this as the ratio (Good and Exclusive in $\mathcal{T}_{\text{M3}}$) / (Total Good in $\mathcal{T}_E$).
- Coverage: This represents the total number of "good" items in $\mathcal{T}_{\text{M3}}$ relative to the total number of "good" items in $\mathcal{T}_E$. Coverage is equivalent to Completeness + Novelty and provides a comparative metric of the number of sound items. It follows that a coverage value greater than 1.0 indicates that $\mathcal{T}_{\text{M3}}$ covers more relevant and correct items than $\mathcal{T}_E$ within the global space for the given taxonomy.

B.2 Taxonomy Evaluation Results

Table 3: **Performance metrics for different taxonomies.**

Type	Topic	Method	Completeness	Soundness	Novelty	Coverage
Conceptual	Online Harmful Content	Simula	**0.749**	**0.980**	**0.865**	**1.614**
		0-shot	0.588	0.957	0.412	1.000
	Logical Fallacies	Simula	**0.726**	0.919	**1.679**	**2.405**
		0-shot	0.458	**0.966**	0.193	0.651
	Risks of Language Models	Simula	**0.867**	**1.000**	0.267	**1.134**
		0-shot	0.467	**1.000**	**0.367**	0.834
Grounded	Animal Phylogenetic Classes	Simula	**0.458**	**0.926**	—	
		0-shot	0.349	0.918		
	Periodic Chemical Elements	Simula	**0.993**	**0.987**	—	
		0-shot	0.775	0.864		
	Mineral Classification	Simula	**0.762**	**0.340**	—	
		0-shot	0.442	0.329		

Description of the taxonomies from Table 3:

- **[Conceptual] Online Harmful Content:** This typology aims to provide a unified classification of harmful content found online. It synthesizes common abuse types described by industry content policies, policy recommendations, community standards, and health expert guidelines. The goal is to create readily usable categories for content moderation, encourage the development of accurate datasets for model training, and raise awareness of less-studied abuse types to improve online safety. This taxonomy categorizes different types of harmful content found online into four main groups: Hate and Harassment; Self-Inflicted Harm; Ideological Harm; and Exploitation — and further branches each into a set of more specific types. (Banko et al., 2020).
- **[Conceptual] Logical Fallacies:** This taxonomy classifies types of logical fallacies into a hierarchical structure. It divides fallacies into two main branches: Formal Fallacies (errors in the structure of the argument) and Informal Fallacies (errors in the content or context of the argument). These main categories are further subdivided into numerous specific types of fallacies, such as Propositional Fallacies, Quantificational Fallacies, and various informal fallacies like Appeal to Ignorance, and Red Herring, and then branching down to increasingly specific types (Curtis, 2023).
- **[Conceptual] Risks of Language Models:** This taxonomy identifies ethical and social risks associated with large-scale language models (LMs). It categorizes these risks into six areas: Discrimination, Hate speech and Exclusion; Information Hazards; Misinformation Harms; Malicious Uses; Human-Computer Interaction Harms; and Environmental and Socioeconomic harms. The taxonomy distinguishes between "observed" risks (already evidenced in LMs) and "anticipated" risks

(considered likely but not yet observed). The goal is to provide a comprehensive framework for understanding and mitigating the potential negative consequences of LMs. (Weidinger et al., 2022).

- **[Grounded] Animal Phylogenetic Classes:** This taxonomy represents the hierarchical classification of animals based on their evolutionary relationships. It is truncated to two levels deeper into the animal kingdom, encompassing its phyla and classes. (Bánki et al., 2025).
- **[Grounded] Periodic Chemical Elements:** This taxonomy organizes chemical elements into a hierarchical structure based on their atomic number, electron configuration, and recurring chemical properties, primarily reflecting their placement in the periodic table. It positions elements into groups from Group 1 - Alkali Metals through Group 18 - Noble Gases, as well as the Lanthanides and Actinides. Each group is further lists individual elements like Sodium (Na) or Gold (Au). The structure represents the periodic trends and shared characteristics within groups, enabling chemists to understand relationships and predict elemental behavior. (International Union of Pure and Applied Chemistry (IUPAC), 2022).
- **[Grounded] Mineral Classification:** Presented in Dana's New Mineralogy, Eighth Edition, this taxonomy is a hierarchical classification system for minerals, employing a four-part numerical code to categorize each species. This system, analogous to the Linnaean taxonomy for biology, organizes minerals based on both their chemical composition and crystal structure. The first number denotes the mineral's class (e.g., anhydrous carbonates), reflecting broad compositional categories or dominant structural features (especially in silicates). The second number signifies the mineral's type, sometimes based on atomic properties, or formula. The third number groups minerals with similar structural arrangements. Finally, the fourth number uniquely identifies the individual mineral species, such as Calcite or Magnesite within the Calcite Group. This numerical system offers a structured and expandable framework, allowing new minerals to be easily integrated while highlighting the chemical and structural relationships between different mineral species. (Wiley et al., 1997-2024).

B.3 Limitations

- **Preference Bias.** As discussed in the Related Work in Appendix A, there is evidence that M3s prefer model-generated text over text generated by humans. In our case, we do not prompt the M3 to express a preference. Rather, we ask if certain nodes semantically approximate other nodes, or if certain nodes are appropriate given the context. However, we did use models from the same model family for both the generation and the evaluation. Thus, future work might want to repeat this experiment with separate generator and critic models.
- **Stochastic Sensitivity.** We did not optimize prompts for each separate taxonomy. As such, the reported metrics likely represent lower bounds.
- **Downstream Application.** While we performed a comparative evaluation of synthetic and real taxonomies, it is not directly clear which are better suited for certain downstream applications.

B.4 Taxonomy Generation Algorithm

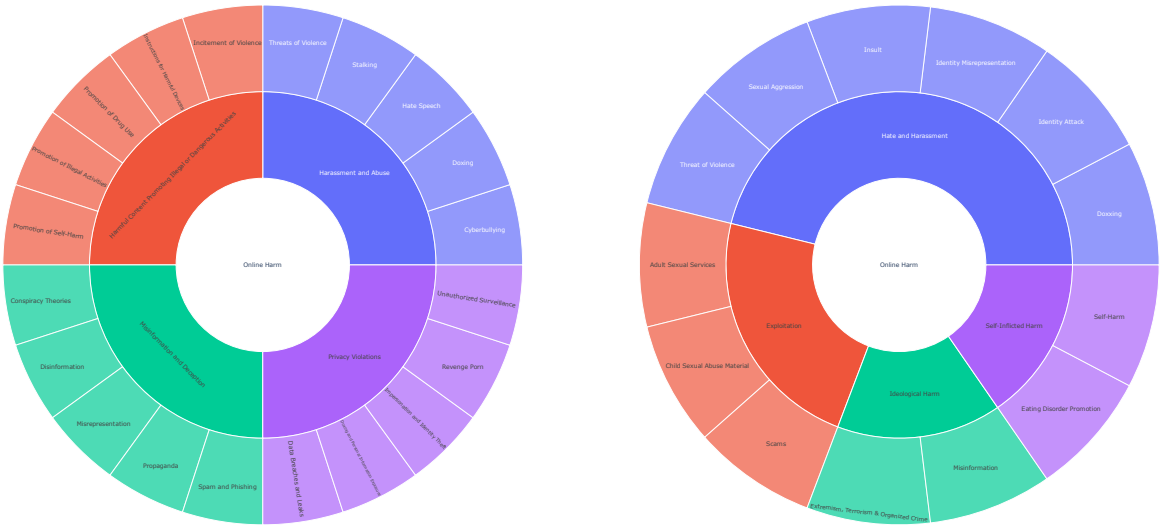
Algorithm 1 Reasoning-Driven Taxonomy Generation

Require: User instructions y, Target depth D, Multi-modal Model M3
Ensure: $\mathbb{T} = \{\mathcal{T}_i\}_{i=0}^{K}$, set representing conceptual space $\mathcal{Y}$

Phase 1: Factor Disentanglement
▷ *Extract prime factors of variation from instructions*
$F \leftarrow$ M3.ProposeFactors(y) ▷ $F = \{f_0, \ldots, f_K\}$
▷ *Initialize set to store generated taxonomies*
$\mathbb{T} \leftarrow \emptyset$

Phase 2: Breadth-First Taxonomic Expansion
for each factor $f_i \in F$ **do**
  ▷ *Initialize taxonomy $\mathcal{T}_i$ with root node*
  $n_{root} \leftarrow$ Node(y, f_i)
  $\mathcal{T}_i \leftarrow \{n_{root}\}$
  ▷ *Queue for nodes at current depth level*
  $Q_{curr} \leftarrow \{n_{root}\}$
  ▷ *Strategic plan guiding expansion consistency*
  $P \leftarrow$ "Expand based on y and f_i"
  for depth $d = 1$ **to** D **do**
    ▷ *Queue to collect nodes for the next depth*
    $Q_{next} \leftarrow \emptyset$
    for each node $n \in Q_{curr}$ **do**
      ▷ *Gather global and local reasoning context*
      $\mathcal{C} \leftarrow \{y, \text{Ancestors}(n), \text{Siblings}(n)\}$
      ▷ ***Step 1: Proposal (Best-of-N)***
      Query M3 N times using context $\mathcal{C}$ and guidance Plan P
      $\mathcal{N}_{raw} \leftarrow \bigcup_{j=1}^{N} \text{M3.Output}_j$ ▷ *Set of raw child nodes*
      ▷ ***Step 2: Critic Refinement***
      ▷ *Prompt M3 to improve completeness, soundness, specificity*
      $\mathcal{N}_{refined} \leftarrow$ M3.Critique($\mathcal{C}, \mathcal{N}_{raw}$) ▷ *Add, remove, merge, edit*
      Add $\mathcal{N}_{refined}$ as children of n in $\mathcal{T}_i$
      $Q_{next} \leftarrow Q_{next} \cup \mathcal{N}_{refined}$
    ▷ ***Step 3: Planning (Per Level)***
    if $d < D$ **then**
      ▷ *Generate guidance for next level based on all new nodes*
      $P \leftarrow$ M3.GeneratePlan(y, Q_{next})
    ▷ *Advance to next depth*
    $Q_{curr} \leftarrow Q_{next}$
  $\mathbb{T} \leftarrow \mathbb{T} \cup \{\mathcal{T}_i\}$
return $\mathbb{T}$

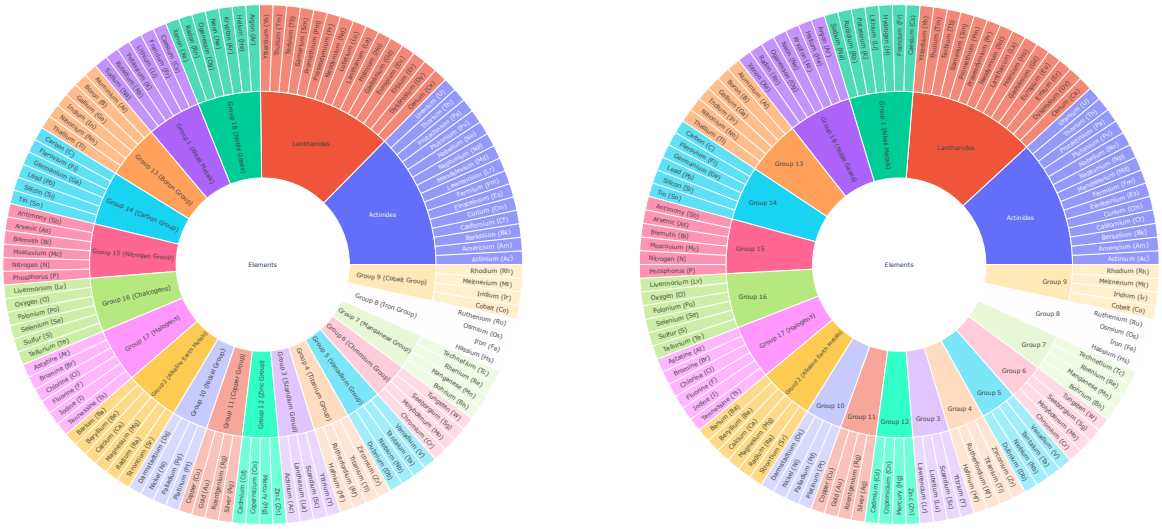
B.5 Qualitative Examples



(a) Harmful Content Taxonomy — Simula

(b) Harmful Content Taxonomy — Expert

Figure 8: Comparison of Online Harmful Content Taxonomy (Simula vs. Expert).



(a) Chemical Elements — Simula

(b) Chemical Elements — Expert

Figure 9: Comparison of Chemical Element Taxonomy (Simula vs. Expert).

C Synthetic Data Generation Algorithm

Algorithm 2 Agentic Data Generation

Require: Taxonomies $\mathbb{T}$, User instructions y, Target size N, Model M3, Complexity ratio $c \in [0, 1]$, Sampling Strategies $\mathcal{S}$, Scenarios per mix K
Ensure: Synthetic Dataset $\mathcal{D}$ containing exactly N high-quality data points

Initialization
▷ *Initialize set to store validated data points*
$\mathcal{D} \leftarrow \emptyset$

Generation Loop
▷ *Continue until target size is reached*
while $|\mathcal{D}| < N$ **do**
  ▷ ***Step 1: Global Diversity Sampling***
  ▷ *Sample a single "Mix" (combination of nodes) based on strategies*
  $\mathcal{M} \leftarrow \text{SampleSingleMix}(\mathbb{T}, \mathcal{S})$
  ▷ ***Step 2: Meta-Prompt Generation (Local Diversity)***
  ▷ *Generate K diverse scenarios (meta-prompts) based on the Mix*
  $\mathcal{P}_{scenarios} \leftarrow \text{M3.GenerateScenarios}(y, \mathcal{M}, K)$
  ▷ *Choose one scenario at random*
  $p_{meta} \leftarrow \text{RandomSample}(\mathcal{P}_{scenarios})$
  ▷ ***Step 3: Complexification***
  if $\text{Random}(0, 1) < c$ **then**
    ▷ *Inject constraints or edge-cases to increase difficulty*
    $p_{meta} \leftarrow \text{M3.Complexify}(p_{meta})$
  ▷ ***Step 4: Generation and Critic-Refinement Loop***
  ▷ *Generate initial proposal based on meta-prompt*
  $d_{point} \leftarrow \text{M3.Generate}(p_{meta})$
  $is_satisfying \leftarrow$ **false**
  repeat
    ▷ *Critic evaluates if data point fulfills meta-prompt requirements*
    $(verdict, explanation) \leftarrow \text{M3.Critique}(p_{meta}, d_{point})$
    if $verdict$ == "satisfying" **then**
      $is_satisfying \leftarrow$ **true**
    else
      ▷ *Agentic refinement based on critic's explanation*
      $d_{point} \leftarrow \text{M3.Refine}(p_{meta}, d_{point}, explanation)$
  until $is_satisfying$ **or** max retries reached
  if $is_satisfying$ **then**
    ▷ *Only add to dataset if it passed the critic*
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{d_{point}\}$
return $\mathcal{D}$

D Evaluating Double-Critic Steps

D.1 Multi-Lingual MMLU

In Table 4 we show empirical critic-rejection sampling results for a subset of MMLU questions on Mathematics, Computer Science, and Physics. We use the subjects' education levels (elementary, high-school, and college) as ground truth complexity categories. We evaluate performance on languages with different resource categories, e.g., "Low", "Mid", and "High", according to their recorded, written, and cataloged NLP resources (Singh et al., 2024b). We observe that our critic-rejection sampling strategy is effective for each language under each complexity condition.

Table 4: **Critic Rejection Sampling on Multi-Lingual MMLU Questions.** We evaluate our critic-rejection sampling method for MMLU questions on Mathematics, Physics, and Computer Science. We use the subject education level (elementary, high-school, and college) as the ground-truth Complexity. We display the realized change in accuracy, μ_{gen} of following critic rejections. Also shown are the average Elo complexity score and the size of the rejected and accepted subsets.

		English (High)			Korean (Mid)			Nepali (Low)		
Complexity	Critic	μ_{gen}	Elo	$\lvert\mathcal{D}\rvert$	μ_{gen}	Elo	$\lvert\mathcal{D}\rvert$	μ_{gen}	Elo	$\lvert\mathcal{D}\rvert$
Level 1	×	0.93 ±0.07	290 ±8	14	0.86 ±0.07	306 ±7	29	0.76 ±0.07	308 ±7	41
	✓	0.98 ±0.01	303 ±2	364	0.97 ±0.01	301 ±2	349	0.97 ±0.01	304 ±2	337
Level 2	×	0.52 ±0.07	427 ±7	46	0.64 ±0.06	428 ±5	53	0.58 ±0.06	431 ±6	60
	✓	0.94 ±0.01	427 ±2	578	0.92 ±0.01	426 ±2	571	0.93 ±0.01	425 ±2	564
Level 3	×	0.67 ±0.10	473 ±5	24	0.54 ±0.10	471 ±5	26	0.41 ±0.09	473 ±4	34
	✓	0.89 ±0.02	467 ±2	278	0.88 ±0.02	468 ±2	276	0.86 ±0.02	465 ±2	268

E Calibrated Complexity Scoring

We compare model-assigned complexity scores against the ground-truth human annotations. We ablate model-assigned complexity scores varying the number of times each sample is scored (**N**) and the batch size (**BS**) of questions being scored simultaneously. Importantly, for fixed N, the number of samples being scored simultaneously (BS) increases the context length but reduces the number of inference passes. For example, for $|\mathcal{D}| = 1000$, N=10 and BS=1, we require $10,000$ inference passes. Setting BS to 5 instead reduces this to $10,000/5 = 2,000$. All things equal, in practice we would thus like to see a higher BS to have similar or better performance than a lower BS.

E.1 Open Generation: MATH

In Figure 10, we show results of comparing model-assigned complexity scores to the human-annotated ground truth (Levels 1-5). To enable side-by-side comparison, we scale both the raw average scores (Score) and the computed ELO rankings (ELO) to lie between 0 and 100 for each {BS, N} grouping. As we increase the number of samples, N, clusters become better separated. Increasing BS > 1 enables the use of latent skill methods like ELO to increase consistency. We found BS = N = 5 to strike an appropriate balance between separation and inference cost.

E.2 Multiple-Choice Generations: MMLU Global

In Table 5 we compare model-assigned complexity scores for a subset of MMLU questions on Mathematics, Computer Science, and Physics. We use the subjects' education levels (elementary, high-school, and college) as ground truth complexity categories. We evaluate performance on languages with different resource categories, e.g., "Low", "Mid", and "High", according to their recorded, written, and catalogued NLP resources per Singh et al. (2024b). After running KMeans on the model-assigned complexity scores, we compute the Normalized Mutual Information (**NMI**) and the Adjusted Rand Index (**ARI**) to evaluate cluster approximation of the ground truth complexity. Finally, we train a logistic regression on model-assigned complexity scores to

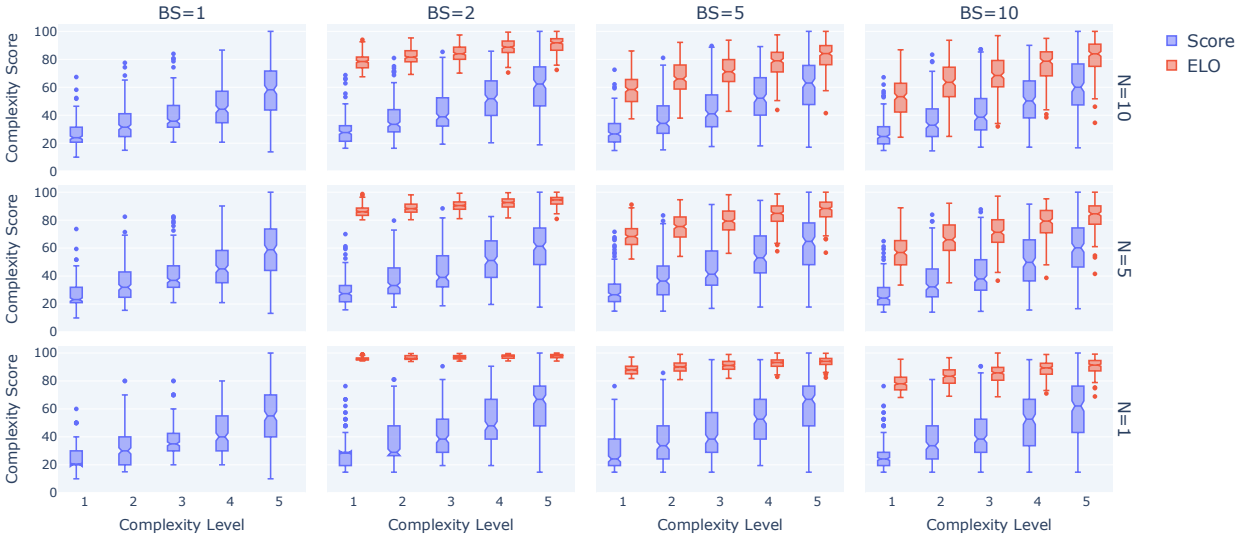


Figure 10: **Complexity score ablation MATH test set.** The MATH dataset comes with ground truth complexity levels ranging from 1-5. We group by the ground truth complexity level and plot the model-assigned complexity scores for each group. We compare using average raw scores (Score) against using ELO rankings (ELO) and ablate the batch size (**BS**) of question scored simultaneously and the number of times each question is scored (**N**). All results are scaled per (BS, N)-grouping to lie between 0 and 100. For $BS = 1$, no pairwise rankings are done, so ELO rankings do not apply.

evaluate them as an estimator for the model's generative performance, reporting the Area Under the Curve (**AUC**). Similar to our findings in Appendix D.1, we find model-assigned complexity scoring robust across several languages. Choosing BS = N = 5 again emerge as reasonable hyper-parameters.

Table 5: **Multi-lingual MMLU Complexity Scoring.** We use exam questions for the topics Mathematics, Physics, and Computer Science, on education levels elementary (Mathematics only), high-school, and college. Taking the education level as our ground-truth complexity level (1-3), we run KMeans on the complexity scores generated by the model and compute the Normalized Mutual Information (**NMI**) and the Adjusted Rand Index (**ARI**). Finally, we compute the Area Under the Curve (**AUC**) of using the complexity scores as an estimator for the model's generative performance. We ablate the batch size (**BS**) of exam questions scored simultaneously and the number of times each exam question is scored (**N**).

BS	**N**	Enlish (High)			Arabic (High)			Dutch (High)			Korean (Mid)			Nepali (Low)		
		NMI	ARI	AUC	NMI	ARI	AUC	NMI	ARI	AUC	NMI	ARI	AUC	NMI	ARI	AUC
1	1	0.27	0.23	0.61	0.32	0.27	0.60	0.32	0.28	0.59	0.32	0.27	0.59	0.32	0.27	0.56
	5	0.27	0.24	0.62	0.33	0.28	0.61	0.34	0.30	0.59	0.34	0.29	0.59	0.33	0.29	0.56
5	1	0.36	0.31	0.63	0.38	0.32	0.60	0.37	0.31	0.59	0.38	0.33	0.59	0.37	0.32	0.57
	5	0.42	0.36	0.64	0.42	0.35	0.62	0.41	0.35	0.59	0.40	0.34	0.60	0.41	0.35	0.57
10	1	0.38	0.33	0.64	0.39	0.34	0.61	0.37	0.32	0.60	0.40	0.35	0.61	0.39	0.33	0.57
	5	0.42	0.37	0.63	0.41	0.36	0.62	0.40	0.34	0.60	0.43	0.38	0.61	0.43	0.37	0.57

E.3 Calibrated Complexity Scoring Algorithm

Algorithm 3 Calibrated Attribute Scoring

Require: Dataset $\mathcal{D}$, Attribute guidance $\mathcal{A}$, Model M3, Samples per item K, Batch size BS
Ensure: Scoring mapping $\mathcal{S}_{final} : d \rightarrow (\text{score}_{raw}, \text{rank}_{elo})$ for all $d \in \mathcal{D}$

Phase 1: Batch Preparation
▷ *Create batch schedule ensuring every item appears in $\approx K$ varied batches*
$\mathbf{B} \leftarrow \text{PrepareBatches}(\mathcal{D}, BS, K)$
▷ *Initialize map to store lists of raw scores for each item*
$RawScores \leftarrow \{d : [] \text{ for } d \in \mathcal{D}\}$

Phase 2: Batch-wise Relative Scoring
for each batch $B \in \mathbf{B}$ **do**
    ▷ *Prompt M3 to score items relative to others within this batch*
    $Scores_B \leftarrow \text{M3.ScoreSingleBatch}(B, \mathcal{A})$
    ▷ *Collect raw scores for aggregation*
    for each $(d, score) \in Scores_B$ **do**
        Append $score$ to $RawScores[d]$

Phase 3: Score Calibration (Elo Reranking)
▷ *Derive pairwise comparisons from batches to compute global Elo ratings*
$EloRatings \leftarrow \text{CalibrateScores}(RawScores, \mathbf{B})$

Phase 4: Aggregation and Finalization
$\mathcal{S}_{final} \leftarrow \emptyset$
for each $d \in \mathcal{D}$ **do**
    ▷ *Compute average raw score across K appearances*
    $\mu_{raw} \leftarrow \text{Mean}(RawScores[d])$
    ▷ *Retrieve globally calibrated rank*
    $r_{elo} \leftarrow EloRatings[d]$
    $\mathcal{S}_{final}[d] \leftarrow (\mu_{raw}, r_{elo})$
return $\mathcal{S}_{final}$

F Downstream Experiments

F.1 Training Configuration and Hyperparameters

We first conducted a hyperparameter sweep to identify the optimal LoRA rank and batch size for each dataset. Checkpoint selection was based on the validation accuracy of the final answer, where we chose the checkpoint corresponding to the peak of a 3-point moving average of scores over the final evaluation steps. This process yielded a fixed set of hyperparameters for each dataset. However, since the optimal learning rate proved sensitive to the training data size, we retained a sweep over two values (0.005, 0.01) for our final experiments. These final runs were performed with 10 different seeds for each configuration. For each data size, we then selected the runs from the best-performing learning rate on average for that specific system version.

We use the Adafactor optimizer for training Shazeer & Stern (2018). To ensure stability, we apply global gradient clipping with a maximum norm of 1.0. The learning rate follows a linear warm-up, followed by cosine decay schedule. The number of warm-up steps is calculated dynamically based on the total training steps (still bounded between a minimum of 500 and a maximum of 5,000 steps)

Table 6: Final hyperparameters used for each dataset. The final learning rate was chosen from the specified set based on validation performance for each training data size.

Dataset	Learning Rates	LoRA Rank	Batch Size
CTI RCM	{0.005, 0.01}	8	16
CTI MCQ	{0.005, 0.01}	8	16
GSM8K	{0.005, 0.01}	16	16
LExAm	{0.005, 0.01}	16	16
MMLU	{0.005, 0.01}	16	16

F.2 Additional Evaluation Results

In Figure 11, we show the results on two additional subsets of Global MMLU (Physics, Math, and Computer Science). The trend shown in Section 4.3 for the Nepali subset continues to be the case here, where the full Simula system shows a better accuracy growth trend with larger datasets.

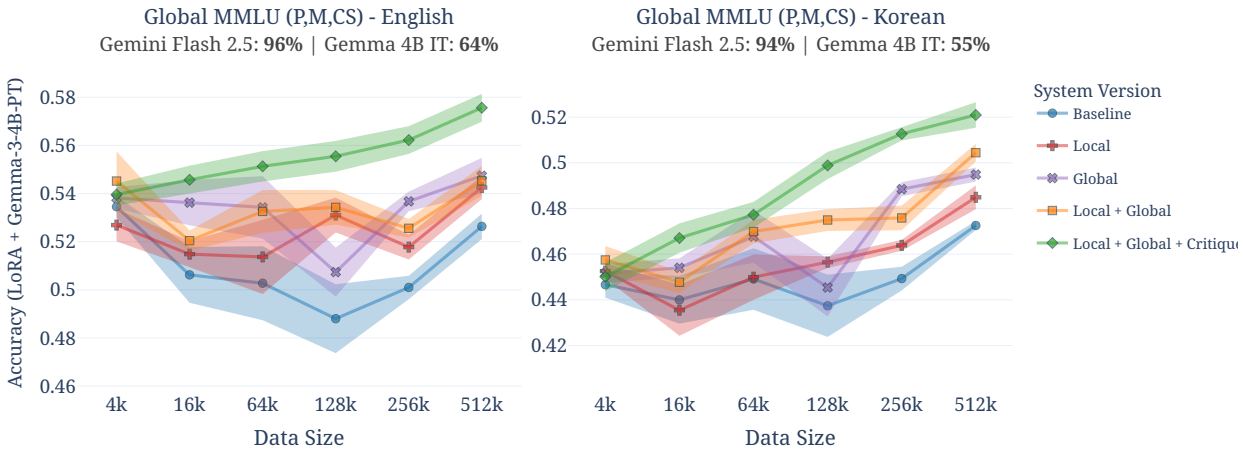


Figure 11: **Downstream Performance on Global MMLU English and Korean.** We report mean accuracy with 95% CI for these additional datasets. We continue to see that the full Simula system version with the critic, provides the best performance compared to the other versions.