Unveiling the Flaws: Exploring Imperfections in Synthetic Data and Mitigation Strategies for Large Language Models

Anonymous ACL submission

Abstract

Synthetic data has been proposed as a solution to address the issue of high-quality data scarcity in the training of large language models (LLMs). Studies have shown that synthetic data can effectively improve the performance of LLMs on downstream benchmarks. However, despite its potential benefits, our analysis suggests that there may be inherent flaws in synthetic data. The uniform format of synthetic data can lead to pattern overfitting and cause significant shifts in the output distribution, thereby reducing the model's instruction-following capabilities. Our work delves into these specific flaws associated with question-answer (Q-A) pairs, a prevalent type of synthetic data, and presents a method based on unlearning techniques to mitigate these flaws. The empirical results demonstrate the effectiveness of our approach, which can reverse the instruction-following issues caused by pattern overfitting without compromising performance on benchmarks at relatively low cost. Our work has yielded key insights into the effective use of synthetic data, aiming to promote more robust and efficient LLM training.

1 Introduction

The remarkable success of large language models (LLMs) (Zhao et al., 2023) largely depends on the quality and diversity of the datasets used for training. However, acquiring large amounts of high-quality data can be challenging due to data scarcity, privacy concerns, and high costs (Liu et al., 2024a). Synthetic data has emerged as a promising solution to address these challenges (Nikolenko, 2019).

Synthetic data, generated through algorithms or generative models rather than collected from real-world events, can be produced at scale and supplement areas where real-world data is scarce or difficult to obtain, such as in mathematical or reasoning tasks. Numerous studies have demonstrated the efficacy of synthetic data in improving model performance (Microsoft, 2024; Mukherjee et al., 2023). Among the various methods of generating synthetic data, a common approach is the creation of synthetic question-answer (Q-A) pairs (NVIDIA, 2024; Maini et al., 2024b; Wei et al., 2023), as Q-A pairs exhibit diversity and richness, encompassing a range of question types from simple factual queries to complex reasoning problems. Another prevalent method is to generate data closely mimicking downstream tasks (Luo et al., 2023; Yu et al., 2023a). These methods have achieved excellent performance on both general-purpose and specialized benchmarks for LLMs.

Despite numerous experiments demonstrating that synthetic data significantly enhances the capabilities of pre-trained models on downstream benchmarks, in this work, we observe a notable decline in the instruction-following capabilities of models after being pre-trained on synthetic data, specifically on synthetic Q-A pairs generated by GPT-4, and subsequent supervised fine-tuning (SFT). This observation prompts a deeper investigation into the underlying causes. While existing studies have extensively covered the applications of synthetic data, there is a notable lack of studies examining its impact on the instruction-following capabilities of LLMs. Furthermore, studies addressing the flaws in synthetic data have primarily focused on historical models or those with capabilities similar to currently trained models (Shumailov et al., 2024; Seddik et al., 2024; Alemohammad et al., 2023), leaving a gap in exploring the deficiencies of synthetic data generated by advanced models like GPT-4.

Our work focuses on exploring the inherent flaws of synthetic data and its impact on LLMs. We find that the token distribution of synthetic data significantly differs from that of the real pre-training data, with synthetic data patterns being relatively uniform. Consequently, models trained on such synthetic data are likely to experience pattern over-

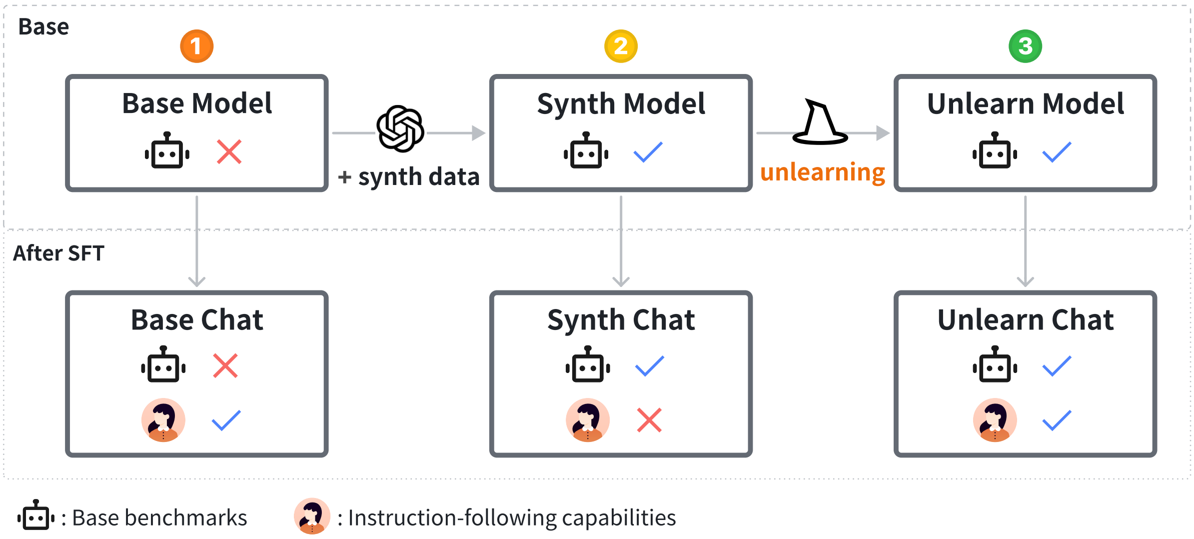

Figure 1: The overall pipeline of our study.

fitting, leading to substantial shifts in their output distributions and resulting in inferior performance.

Based on these observations, we propose a novel strategy that leverages unlearning techniques to reduce the impact of misleading synthetic data patterns while preserving the LLM's foundational abilities on benchmarks and restoring its instruction-following capabilities. This strategy employs a lower-bounded forgetting loss, which is controllable and superior to traditional unlearning approaches. Our experimental results demonstrate that this strategy effectively mitigates the adverse impacts of synthetic data, balancing the LLM's performance on benchmarks with its ability to follow instructions at significantly low training costs. Our contributions are summarized as follows:

- **Identification of Synthetic Data Limitations:** We provide a comprehensive analysis of the inherent limitations in synthetic data, specifically synthetic Q-A pairs, focusing on data distribution differences and pattern overfitting observed in models.
- **Unlearn Method to Address Synthetic Data Issues:** We propose a novel unlearning strategy that effectively mitigates the adverse effects of synthetic data, thereby preserving the LLM's foundational abilities on benchmarks while reversing its instruction-following capabilities at significantly low training costs.

2 Related Work

Applications and Limitations of Synthetic Data. Studies have shown that synthetic data has achieved remarkable results on downstream benchmarks (Luo et al., 2023; Microsoft, 2024; Mukherjee et al., 2023; Wei et al., 2023), addressing issues such as data scarcity and privacy (Liu et al., 2024a; Villalobos et al., 2022; Maini et al., 2024b). For instance, Microsoft's Phi-3 (Microsoft, 2024) model, trained on heavily filtered publicly available web data and synthetic data, has outperformed much larger models on both academic benchmarks and internal testing. MagicoderS-CL-7B (Wei et al., 2023), a 7B parameter code model trained on synthetic code problems and answers generated by LLMs, even surpasses the prominent ChatGPT on many coding benchmarks. However, synthetic data is not without flaws. Several critical issues have been identified, particularly concerning model performance and data distribution integrity. One significant concern is the phenomenon of model collapse (Shumailov et al., 2024; Seddik et al., 2024), where training on model-generated data leads to the disappearance of the tails of the original content distribution. Furthermore, the recursive use of synthetic data in training generative models can amplify artifacts and biases, ultimately degrading model performance, as demonstrated by the concept of Model Autophagy Disorder (MAD) (Alemohammad et al., 2023). Task-specific synthetic data often lacks diversity and exhibits regional biases (Yu et al., 2023b), with effectiveness varying by task nature (Li et al., 2023).

LLM Unlearning. Unlearning in LLMs involves the elimination of specific undesired targets while preserving overall performance (Liu et al., 2024b). Strategies vary from specific data points to higher-

Position Embedding	Hidden Size	FFN Size	Heads	Layers	Context Length
RoPE (Su et al., 2023)	$2,048$	$5,504$	32	32	$4,096$

Table 1: The architecture details of BaseLM.

level concepts such as harmful language or specific knowledge domains (Jang et al., 2022; Lu et al., 2022; Eldan and Russinovich, 2023). Effective unlearning requires robustness and generalization (Patil et al., 2024; Maini et al., 2024a; Shi et al., 2023) with efficient handling of computational costs (Pawelczyk et al., 2023). Existing unlearning methods leverage various fine-tuning techniques, including gradient ascent, parameter-efficient fine-tuning, and KL-divergence-based methods, each with unique strengths and limitations regarding run-time and memory costs (Yao et al., 2024; Jang et al., 2022; Eldan and Russinovich, 2023). While unlearning methods have been utilized to manage harmful data and reduce hallucinations in models, their application to synthetic data remains under-explored. Our research aims to fill this gap by applying unlearning strategies to mitigate the adverse effects of synthetic data on LLMs.

3 Experimental Setup

In this section, we outline the experimental design, including dataset selection, model configurations, and evaluation benchmarks.

Datasets. We utilize five distinct datasets:

- *NonSynth data*: A comprehensive non-synthetic dataset collected from diverse sources (Soldaini et al., 2024; Penedo et al., 2023; Soboleva et al., 2023), including webpages, books, research papers, and codebases.
- *SynthQA data*: Synthetic Q-A pairs generated by GPT-4, based on a variety of sources including webpages, books, and other textual materials, covering topics such as mathematics, coding, and general knowledge.
- *MixedIns data*: Instructions consisting of general knowledge, mathematics, and coding, primarily generated by GPT-4 and human contributors.
- *U33B data* (Yuan et al., 2023): Aggregated synthetic dataset of diverse reasoning paths generated from GSM8K dataset by multiple LLMs to enhance mathematical reasoning capabilities.
- *OpenHermes-2.5 data* (Teknium, 2023): An extension of the OpenHermes-1 dataset, primarily consisting of synthetically generated instruction and chat samples.

Models. We use the following models in our experiments:

- *BaseLM*: A Llama-like (Touvron et al., 2023) 2B model trained from scratch. We set the learning rate to 1.0×10^{-4} and adopt a cosine learning rate schedule, training on a total of 1 trillion tokens. The details of hyperparameters are listed in Table 1.
- *BaseLM-Chat (MixedIns/OpenHermes-2.5)*: Chat models obtained by performing SFT on BaseLM using MixedIns or OpenHermes-2.5 data. We set the learning rate to 2.0×10^{-5}, the number of epochs to 2, the context length to $4,096$, and the batch size to 64.

Benchmarks. We evaluate the capabilities of models using the following benchmarks:

- *Bilingual Capabilities*: Evaluated using the MMLU (Hendrycks et al., 2021), CMMLU (Li et al., 2024) and C-Eval (Huang et al., 2023) benchmarks to assess the models' proficiency in handling both English and Chinese tasks.
- *Coding Proficiency*: Assessed with the HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) benchmarks, which measure the models' ability to generate correct and efficient code snippets based on given problems.
- *Mathematical Reasoning*: Measured using the GSM8K (Cobbe et al., 2021) benchmark, which tests the models' ability to solve complex mathematical problems.
- *Instruction-Following Capability*: Analyzed through FollowBench (Jiang et al., 2024) and MT-bench (Zheng et al., 2023), evaluating the models' ability to understand and follow complex instructions.

4 Defect Analysis of Synthetic Data

In this section, we systematically analyze the flaws of synthetic data, specifically synthetic Q-A pairs, by examining their data distribution differences and pattern overfitting observed in LLMs. This analysis is crucial to understand how synthetic data impacts the LLMs' foundational abilities on benchmarks and instruction-following capabilities.

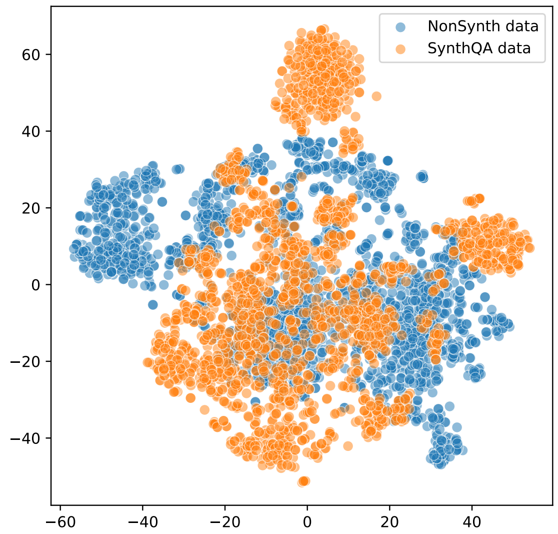

Figure 2: t-SNE visualization of data distributions. The clusters of NonSynth and SynthQA data show considerable non-overlap.

4.1 Data Distribution Differences

One of the primary concerns with synthetic data is the potential mismatch between its distribution and that of real-world data. This discrepancy can result in models that perform well on synthetic data but fail to generalize effectively to real-world scenarios.

Data Characteristic Differences. Synthetic data generated by LLMs often exhibits distinct distributional characteristics compared to non-synthetic data. To illustrate these differences, we sample $2,000$ entries from both NonSynth and SynthQA data. Using the embeddings from the last hidden state of BaseLM, we apply t-SNE (Van der Maaten and Hinton, 2008) for dimensionality reduction and visualize the data distributions in Figure 2. The t-SNE visualization reveals that the clusters of NonSynth and SynthQA data have considerable areas of non-overlapping, which indicates that SynthQA data does not perfectly replicate the characteristics of NonSynth data. Such differences may lead to misinterpretations of real-world scenarios by LLMs trained on synthetic data.

Simplified Data Patterns. Synthetic data often contains repetitive and structurally predictable elements, which simplify the complexity of real-world interactions and patterns. This simplification can result in data that fails to capture the intricacies of human language and interaction. To explore this, we again sample $2,000$ entries from both NonSynth and SynthQA data and calculate

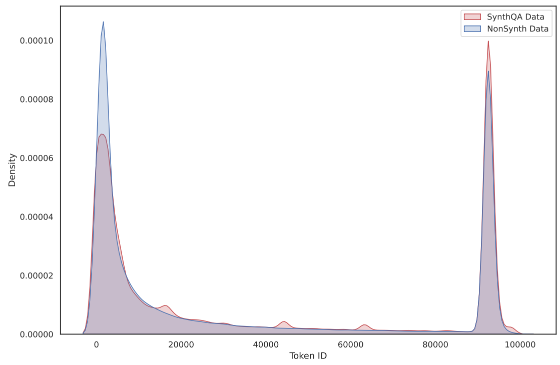

Figure 3: Kernel density estimation of token IDs for NonSynth and SynthQA data. The token frequency distribution for SynthQA data shows several small peaks, indicating high structural consistency for specific tokens compared to NonSynth data.

the token frequencies based on the tokenizer of BaseLM. Figure 3 presents the kernel density estimation (KDE) (Parzen, 1962) plot of token IDs. We observe that the distribution of token frequencies for SynthQA data exhibits several noticeable small peaks compared to NonSynth data. We find that these peaks correspond to tokens with a high degree of structural consistency within SynthQA data. Specifically, tokens like "question" (ID: 44246), "answer" (ID: 63264), and "summary" (ID: 16752) contribute to these observable peaks. The presence of these structural tokens indicates a repetitive pattern in SynthQA data, reflecting its inherent simplicity and lack of variability compared to NonSynth data. By over-representing certain tokens, synthetic datasets risk failing to encapsulate the full spectrum of linguistic diversity found in non-synthetic data, which may lead to models trained on such data being less robust and adaptable.

4.2 Pattern Overfitting

In this part, we investigate the detrimental effects of synthetic data on instruction-following capabilities and output distributions of LLMs. Our analysis highlights how synthetic data, specifically synthetic Q-A pairs, can cause overfitting to specific patterns observed in Section 4.1, potentially affecting the performance of chat models.

Instruction-Following Capability Decline. While synthetic data has shown considerable potential in enhancing the foundational abilities on benchmarks for LLMs in the pre-training stage, our work identifies significant challenges when these models undergo SFT. Specifically, we observe a notable decline in the instruction-following capabilities of

Models	C-Eval	CMMLU	MMLU	HumanEval	MBPP	GSM8K	Avg.
BaseLM	39.05	38.83	38.08	9.76	12.00	15.09	25.47
SynthLM	47.71	47.56	47.27	18.90	18.40	16.60	32.74
RefineLM	46.79	47.15	45.82	17.07	18.30	13.42	31.42
UnlearnLM	48.09	47.29	47.53	20.73	18.60	11.45	32.28

Table 2: Evaluation results of base models with continued pre-training and unlearning. SynthLM is obtained by training BaseLM with a dataset containing 300 billion tokens, of which 2% are from the SynthQA data. RefineLM is derived from SynthLM by further training with an additional 300 billion tokens of NonSynth data. UnlearnLM is obtained by performing our unlearning strategy on SynthLM using 1 billion tokens from the SynthQA data.

Models	FollowBench		MT-Bench	C-Eval	CMMLU	MMLU	HumanEval	MBPP	GSM8K
	SSR	HSR							
BaseLM-Chat	39.95	27.58	5.45	39.92	40.16	41.55	18.29	17.80	14.33
SynthLM-Chat	38.29	24.00	5.39	49.50	48.37	49.06	21.95	22.60	22.21
RefineLM-Chat	39.60	25.22	5.43	47.71	47.40	47.08	17.68	23.60	22.37
UnlearnLM-Chat	42.00	27.87	5.85	49.12	48.83	48.82	20.12	21.80	21.99

Table 3: Evaluation results of chat models with continued pre-training and unlearning. Models with the suffix "-Chat" represent chat models derived from their corresponding base models in Table 2 through SFT on the MixedIns data.

chat models, underscoring critical limitations associated with the use of synthetic Q-A pairs. To investigate this issue, we design a series of experiments. We mix 2% SynthQA data with NonSynth data to create a dataset containing 300 billion tokens and perform continued pre-training on BaseLM with a fixed learning rate of 5.0×10^{-5}. The evaluation results, presented in Table 2 (SynthLM v.s. BaseLM), show that the foundational abilities of BaseLM has significantly improved after training with synthetic Q-A pairs. We validate the role of synthetic data through ablation experiments in Section 6. However, following SFT, we notice a severe decline in instruction-following capabilities in the resulting chat model, as shown in Table 3 (SynthModel-Chat v.s. BaseLM-Chat).

Output Distribution Changes. Due to simplified data patterns in synthetic data, a critical concern is its propensity to cause overfitting. To investigate this effect, we sample $2,000$ entries each from OpenHermes-2.5 and MixedIns data. We then calculate their perplexity using BaseLM and SynthLM. Figure 4 shows the KDE plot of perplexity values for these two types of data. We can clearly observe that the perplexity distribution for SynthLM exhibits a noticeable shift and reduced variance compared to BaseLM, which is similar to the phenomenon of model collapse (Shumailov et al., 2024). This suggests a tendency for the model to overfit to the patterns present in the synthetic data, reducing its ability to deal with real-world variability.

5 Unlearning-Based Mitigation Strategy

In this section, we introduce our unlearning strategy and describe the experiments conducted to implement this approach.

5.1 Unlearning Strategy

To address the identified flaws in synthetic data, we propose a mitigation strategy based on unlearning techniques. Typically, unlearning is applied to remove harmful data or reduce model hallucinations. In this context, we leverage unlearning to recalibrate the LLM's understanding, mitigating the adverse effects of synthetic data while preserving its beneficial attributes.

Task Description. In the task where the LLM predicts the next token y_i based on an existing token sequence $y_{<i} = [y_1, y_2, \ldots, y_{i-1}]$, let $p(y_{<i}; \theta)$ denote the predicted probability of y_i. Formally, this can be expressed as:

$$p(y_{<i}; \theta) = P(y_i \mid y_{<i}; \theta),$$

where θ represents the parameters of the LLM. The prediction accuracy is evaluated using the cross-entropy loss function. Specifically, the loss for

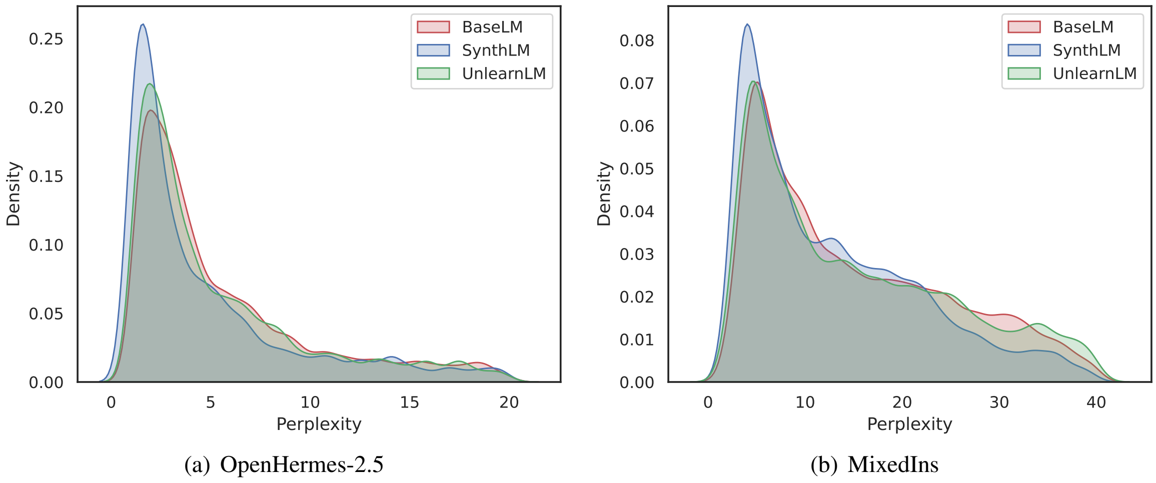

Figure 4: Kernel density estimation of perplexity values for OpenHermes-2.5 and MixedIns data using BaseLM, SynthLM and UnlearnLM. SynthLM shows a noticeable shift and reduced variance, while UnlearnLM corrects the distribution shift.

predicting y_i is given by $l(p(y_{<i};\theta), y_i)$, where $l(\text{input}, \text{target})$ denotes the cross-entropy loss between the predicted probability distribution and the actual target token.

Unlearning Loss. Following previous work (Yao et al., 2024), the unlearning loss function we designed consists of three parts:

• *Lower-Bounded Forgetting Loss*: This component focuses on forgetting the biased distribution of specific synthetic data. Unlike previous methods that apply gradient ascent (Thudi et al., 2022) (i.e., adding a negative sign to the cross-entropy loss to introduce irrelevant elements into the predictions), we have observed that this method has uncontrolled loss due to the logarithm approaching zero without a lower bound. Therefore, we designed a simple yet effective lower-bounded forgetting loss by inverting the model prediction probabilities in the cross-entropy loss. This retains the original forgetting loss function's features while adding a lower bound (*i.e.,* 0). We validate the effectiveness of our forgetting loss approach through ablation experiments in Section 6. The designed lower-bounded forgetting loss L_{fgt} can be defined as:

$$L_{\text{fgt}} = \sum_{i=1}^{|y^{\text{syn}}|} l(1 - p(y_{<i}^{\text{syn}};\theta), y_i^{\text{syn}}).$$

• *Replay Loss*: We sample a portion of the data from the trained non-specific synthetic data for replay, using the cross-entropy loss to allow the model to retain memory of historical knowledge. The replay loss L_{rpy} can be defined as:

$$L_{\text{rpy}} = \sum_{i=1}^{|y^{\text{non-syn}}|} l(p(y_{<i}^{\text{non-syn}};\theta), y_i^{\text{non-syn}}).$$

• *Bias Mitigation Loss*: After unlearning, we aim to ensure that the LLM's output distribution on the trained non-specific synthetic data does not change excessively. Therefore, we calculate the KL divergence between the current model and the original model on the data used for replay, as the bias mitigation loss L_{mtn} to preserve the original performance:

$$L_{mtn} = \sum_{i=1}^{|y^{\text{non-syn}}|} \text{KL}(p(y_{<i}^{\text{non-syn}};\theta_{\text{ori}}) \parallel p(y_{<i}^{\text{non-syn}};\theta_i)), \quad (1)$$

where θ_{ori} represents the parameters of the original model.
Finally, we obtain the total unlearning loss function as follows:

$$L_{\text{unlearn}} = w_{\text{fgt}} \cdot L_{\text{fgt}} + w_{\text{rpy}} \cdot L_{\text{rpy}} + w_{\text{mtn}} \cdot L_{\text{mtn}},$$

where w_* denotes the weights corresponding to each part of the loss L_*.

5.2 Unlearning Experiments

In this part, we detail the experimental process of applying unlearning techniques. Our objective is mitigate the adverse effects on models trained with synthetic data. Specifically, we aim to enhance the

Models	FollowBench		MT-Bench	GSM8K
	SSR	HSR		
BaseLM-Chat (O.H.)	40.25	27.27	5.76	34.27
SynthLM* (U33B)-Chat (O.H.)	39.95	25.13	5.61	43.06
UnlearnLM* (U33B)-Chat (O.H.)	40.21	27.26	5.87	42.00

Table 4: Evaluation results of chat models with continued pre-training on U33B data and subsequent unlearning. SynthLM*(U33B) is the base model trained with 40 billion tokens including 2% U33B data. UnlearnLM*(U33B) is derived from SynthLM*(U33B) by applying our unlearning strategy. Models with the suffix "-Chat(O.H.)" represent chat models derived from their corresponding base model through SFT on the OpenHermes-2.5 data.

instruction-following capabilities of models while preserving their foundational abilities.

Basic Implementation. We utilize NonSynth data containing 300 billion tokens to perform continued pre-training on SynthLM in Table 2, with the aim of recovering the model's instruction-following capabilities. We utilize a fixed learning rate of 5.0×10^{-5} during the training process. From the results in Table 2 and 3, we can clearly observe that extensive training with non-synthetic data leads to enhanced instruction-following capabilities (RefineLM-Chat v.s. SynthLM-Chat) at the cost of a decline in overall base model performance (RefineLM v.s. SynthLM). However, this approach does not completely eliminate the negative impact of the synthetic data on the model.

Unlearning Strategy Implementation. We propose employing the unlearning strategy on SynthLM. We apply lower-bounded forgetting loss on texts from the SynthQA data with 1 billion tokens. Concurrently, we perform replay loss and bias mitigation loss on the trained NonSynth data alongside the unlearning process. We use a fixed learning rate of 5.0×10^{-5} and set the weights $w_{\text{fgt}} = 0.01$, $w_{\text{rpy}} = w_{\text{mtn}} = 1$. As can be seen from Table 2 and 3, although unlearning leads to a slight decrease in foundational abilities of base (UnlearnLM v.s. SynthLM) and chat (UnlearnLM-Chat v.s. SynthLM-Chat) models, especially math abilities, there is a considerable improvement in instruction-following capabilities (UnlearnLM-Chat v.s. BaseLM-Chat).

Distribution Shift Correction. The unlearning process partially corrects the output distribution shift of the LLM. Following the experiments in Section 4.2, we include the perplexity distribution of UnlearnLM on OpenHermes-2.5 and MixedIns data in Figure 4. It can be observed that the distribution shift has been effectively corrected after unlearning, indicating a significant reduction in pattern overfitting.

It's worth noting that the instruction-following capabilities of UnlearnLM-Chat after unlearning with just **1 billion tokens** surpass the performance of both RefineLM-Chat trained on **300 billion tokens** and BaseLM-Chat. Additionally, the foundational abilities of UnlearnLM are comparable to those of RefineLM, suggesting that the beneficial effects of synthetic data on model performance have been preserved. This underscores the efficacy of our method in achieving **more robust and efficient LLM training at significantly lower training costs**.

6 Ablation Study

6.1 Effectiveness of Unlearning Strategy

To explore the effectiveness of our unlearning strategy across different types of synthetic data, we conduct experiments using the U33B data. We first perform continued pre-training on the BaseLM with 40 billion tokens of data, including 2% U33B data, resulting in SynthLM*(U33B). We utilize a fixed learning rate of 5.0×10^{-5} during the training process. Following this, we apply our unlearning strategy to mitigate the adverse effects of U33B data on instruction-following capabilities while preserving its positive impact on foundational abilities, particularly in mathematics. Specifically, we employ the same unlearning parameters as described in Section 5.2, resulting in UnlearnLM*(U33B). We conduct SFT on the resulting models using OpenHermes-2.5 data. The evaluation results are presented in Table 4. The results indicate that while the model trained with U33B data improves its mathematical abilities, it exhibits a decline in instruction-following capabilities. However, after applying our unlearning strategy, the instruction-following capabilities are restored, while retaining the enhancements in mathematical abilities provided by the U33B data. These findings suggest

Models	C-Eval	CMMLU	MMLU	HumanEval	MBPP	GSM8K	Avg.
BaseLM	39.05	38.83	38.08	9.76	12.00	15.09	25.47
MixSynthLM	44.63	44.12	45.00	18.29	19.40	14.95	31.07
NonSynthLM	42.33	40.46	40.88	18.29	17.80	12.21	28.66

Table 5: Evaluation results of BaseLM with continued pre-training on synthetic and non-synthetic data. MixSynthLM is BaseLM trained with 40 billion tokens including 2% SynthQA data. NonSynthLM is BaseLM trained with 40 billion tokens of NonSynth data.

Models	C-Eval	CMMLU	MMLU	HumanEval	MBPP	GSM8K	Avg.
SynthLM	47.71	47.56	47.27	18.90	18.40	16.60	32.74
UnlearnLM (GA)	26.58	25.08	39.28	11.59	9.60	6.82	19.82
UnlearnLM (**Ours**)	48.09	47.29	47.53	20.73	18.60	11.45	32.28

Table 6: Evaluation results of SynthLM with different unlearning strategies applied. UnlearnLM (GA) is derived from SynthLM by applying traditional gradient ascent loss. UnlearnLM (Ours) is derived by applying our lower-bounded forgetting loss.

that our unlearning strategy could be extended to other types of open-source synthetic data.

6.2 Impact of Synthetic Data on Model Performance

To verify that SynthQA data, rather than NonSynth data, contributes to the significant performance improvements in BaseLM, we conduct a controlled ablation experiment. We evaluate two models: NonSynthLM, which is the BaseLM trained with 40 billion tokens of NonSynth data, and MixSynthLM, which is the BaseLM trained with 40 billion tokens of data including 2% SynthQA data. To ensure a fair comparison and better verify the impact of synthetic data, the NonSynth data used to train both NonSynthLM and MixSynthLM is the same high-quality data corpus used to generate the SynthQA data. The evaluation result is shown in Table 5. We can see that MixSynthLM exhibits markedly superior performance enhancements. This confirms that synthetic data plays a critical role in boosting base model performance.

6.3 Efficacy of Bounded Forgetting Loss

When introducing our unlearning strategy in Section 5.1, we use the lower-bounded forgetting loss to forget the biased distribution of specific synthetic data. To evaluate the effectiveness of this approach compared to the traditional gradient ascent loss, we conduct a comparative experiment where the SynthLM in Table 2 undergo unlearning using both the lower-bounded forgetting loss and the traditional gradient ascent loss. As shown in Table 6, we can clearly observe that the model subjected to traditional gradient ascent loss exhibits severe performance degradation. This may be due to the uncontrolled magnitude of negative loss during training. Conversely, the lower-bounded forgetting loss results only in a partial decline in mathematical abilities.

7 Conclusion

In this work, we have systematically explored the potential issues associated with synthetic data, particularly focusing on synthetic Q-A pairs, and their impact on the performance of LLMs. Our analysis has identified inherent flaws in synthetic data, such as pattern overfitting and significant shifts in output distribution, which can degrade the instruction-following capabilities of LLMs. To mitigate these adverse effects, we have proposed an innovative unlearning-based strategy. This strategy employs a lower-bounded forgetting loss, which is controllable and superior to traditional unlearning approaches at significantly lower training costs. The empirical results demonstrate that our strategy effectively addresses the limitations of synthetic data and corrects the output distribution shift, thereby enhancing the instruction-following capabilities while preserving foundational capabilities of LLMs on benchmarks. Our work has demonstrated a viable path to leverage the advantages of synthetic data without being adversely affected by its limitations, enhancing the robustness and efficiency of LLM training.

8 Limitations

Despite our substantial efforts, several limitations warrant further consideration. Firstly, while our unlearning-based strategy has shown promise in mitigating the negative effects of synthetic data, it may still cause degradation in specific model capabilities, such as mathematical reasoning. Moreover, its scalability to much larger models remains untested. As LLMs continue to grow in size and complexity, the computational efficiency and practical applicability of this strategy require further validation. Additionally, this study primarily focuses on the flaws and mitigation strategies related to Q-A pair synthetic data. Although we have demonstrated the effectiveness of our unlearning strategy on the open-source synthetic dataset U33B, many other forms of synthetic data remain unexplored. Furthermore, the quality of synthetic data generated by GPT-4 used in this study may not fully represent the entire spectrum of synthetic data quality. Different synthetic data generation techniques and tools can produce data with varying degrees of imperfections, potentially impacting the effectiveness of our mitigation strategy. Further investigation into more advanced unlearning techniques is necessary to minimize these side effects. We will continue to refine and enhance our method in future work.

References

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. 2023. Self-consuming generative models go mad. *Preprint*, arXiv:2307.01850.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *Preprint*, arXiv:2310.02238.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Preprint*, arXiv:2305.08322.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Followbench: A multi-level fine-grained constraints following benchmark for large language models. *Preprint*, arXiv:2310.20410.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. *arXiv preprint arXiv:2310.07849*.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024a. Best practices and lessons learned on synthetic data for language models. *Preprint*, arXiv:2404.07503.

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

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. *Preprint*, arXiv:2205.13636.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

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

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024b. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380*.

Microsoft. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Sergey I. Nikolenko. 2019. Synthetic data for deep learning. *https://arxiv.org/pdf/1909.11512*.

NVIDIA. 2024. Nemotron-4 340b technical report. *Technical Report*.

Emanuel Parzen. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2024. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *ICLR*.

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

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Debbah. 2024. How bad is training on synthetic data? a statistical analysis of language model collapse. *Preprint*, arXiv:2404.05090.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024. The curse of recursion: Training on generated data makes models forget. *Preprint*, arXiv:2305.17493.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. *Preprint*, arXiv:2109.13398.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

P. Villalobos, J. Sevilla, L. Heim, T. Besiroglu, M. Hobbhahn, and A. Ho. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Empowering code generation with oss-instruct. *arXiv preprint arXiv:2312.02120*.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning. *Preprint*, arXiv:2310.10683.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023a. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023b. Large language model as attributed training data generator: A tale of diversity and bias. *Preprint*, arXiv:2306.15895.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Examples of Data Utilized in This Work

In Section 3, we introduce the various datasets employed in our research. To provide a clear understanding of the data characteristics and content diversity, we present examples for each dataset type in Table 7.

Dataset	Type	Source	Sample
NonSynth	Non-synthetic Data	Webpage	As an independent Nissan service repair centre and our aim is to provide our customers with an alternative to the high servicing and repair costs associated with large Nissan dealerships.
		Book	The sovereign or heir of Moscow was to succeed Yan Kazimir, details of boundaries and succession to be settled by the Diet, both sides to refrain from hostilities till the Swedes were expelled, and neither to make peace with Sweden separately.
		Research Paper	\section*Introduction\n\nClubbing is a central part of many young adults' lives. In Norway, the club culture is alcohol driven and drinking to intoxication is a common phenomenon in the Nordic countries (Mäkelä et al., 2001), as in other parts of the Western world (Measham \\& Brain, 2005; Moore, 2010).
		Codebase	from openpyxl import load_workbook\nimport numpy as np \n\ndef read():\n #Load data from workbook\n wb = load_workbook('DL03_Teste01_Dados.xlsx')\n sheet = wb['Planilha1']\n\n #Recover data\n datasheet = []\n for row in sheet.iter_rows(): \n newRow = list()\n for cell in row:\n newRow.append(cell.value)
SynthQA	Synthetic Q-A Pair	Mathematic	Question:\n\nWhich of the following represents the derivative of the function $f(x) = x^2$?\n\nA) 2\n\nB) 2x\n\nC) x^2\n\nD) x^3\n\n**Answer:**\n\nTo find the derivative of $f(x) = x^2$, we apply the power rule of differentiation. If $f(x) = x^n$, then $f'(x) = nx^{n-1}$.\n\nApplying the rule to $f(x) = x^2$, we have:\n\n $f'(x) = 2x^{2-1} = 2x$\n\nSo, the correct answer is **B) 2x**.
		Coding	Create a test suite of quantum physics questions categorized by difficulty.\\n\nBest Answer:\nGiven the complexity, this task would typically be broken into multiple stages and require a team of experts ... if __name__ == "__main__":\n app.run()
		General Knowledge	探讨迷茫心理、求助的重要性与技巧、心理学视角下的人际交流和资源交换原则。\n从心理学来讲,人生之所以迷茫,是对未知的无能为力。 ... 明确对方需求,才能获得你想要的帮助。
MixedIns	Instruction	Mathematic	{"role": "user", "content": "帮助用户解决以下数学问题。\n问题: 计算 45 ÷ 5 + 6 × 2"}, {"role": "assistant", "content": "根据算术运算法则, 先计算乘除再计算加减: \n45 ÷ 5 + 6 × 2 = 9 + 12 = 21"}
		Coding	{"role": "user", "content": "创建一个包含10个随机数字的列表，并按升序排列。\n"}, {"role": "assistant", "content": "以下是Python语言的代码实现：\n"'python\nimport random\n# 生成一个包含10个随机数字的列表\nmy_list = [random.randint(1, 100) for i in range(10)]\n# 使用sort()方法按升序排列\nmy_list.sort()\n# 输出\nprint(my_list)"}
		General Knowledge	{"role": "user", "content": "给定以下两种动物的名字，请确定它们属于哪个动物大类；鲨鱼和海马。\n鲨鱼和海马\n"}, {"role": "assistant", "content": "鲨鱼属于鱼类，而海马属于海龙目，也就是海马目动物。"}
U33B	Synthetic Q-A Pair	Mathematic	{"query": "Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?", "response": "Natalia sold 48/2 = «48/2=24»24 clips in May.\nNatalia sold 48+24 = «48+24=72»72 clips altogether in April and May.\n#### 72"}
OpenHermes-2.5	Instruction and Chat Example		{"from": "human", "value": "Do you know any jokes about animals with ailments?"}, {"from": "gpt", "value": "What's worse than a centipede with sore feet? A giraffe with a sore throat"}

Table 7: Illustrative examples for six datasets used in this work.