PANGEA: Projection-Based Augmentation With Non-Relevant General Data for Enhanced Domain Adaptation in LLMs

Seungyoo Lee
KAIST
punctuate@kaist.ac.kr

Giung Nam
KAIST
giung@kaist.ac.kr

Moonseok Choi
KAIST
ms.choi@kaist.ac.kr

Hyungi Lee[†]
Kookmin University
lhk2708@kookmin.ac.kr

Juho Lee[†]
KAIST
juholee@kaist.ac.kr

Abstract

Modern large language models (LLMs) achieve competitive performance across a wide range of natural language processing tasks through zero-shot or few-shot prompting. However, domain-specific tasks often still require fine-tuning, which is frequently hindered by data scarcity, i.e., collecting sufficient domain-specific data remains a practical challenge. A widely adopted solution is to generate synthetic data using LLMs by augmenting a small set of available domain-specific examples. In this work, we first identify fundamental limitations of such approach in terms of both data diversity and quality, particularly when relying on only a handful of domain-specific examples. We then propose our method, PANGEA, which leverages large-scale, publicly available general-purpose data—entirely unrelated to the target domain—to generate more diverse and higher-quality synthetic data. Our extensive experiments on domain-specific benchmarks, including GSM8K, MedQA, and FinQA, as well as a custom domain-specific language task, validate the effectiveness of our approach.

1 Introduction

Large language models (LLMs) have achieved strong general-purpose capabilities via large-scale pre-training, handling tasks from natural language understanding to complex reasoning [11, 12, 31], yet they often struggle on domain-specific tasks which demand expert-level knowledge such as those in the medical or legal sectors [3, 13, 28]. A key bottleneck is the scarcity of high-quality domain data, as many real-world datasets (e.g., patient records or proprietary financial documents) are often inaccessible due to privacy constraints [1, 18, 19, 41]. Moreover, constructing expert-annotated datasets is costly and time-consuming due to the necessity for the limited availability of qualified annotators. As a result, niche domains often have far less training data than is available for web-scale corpus [6, 15, 47, 48].

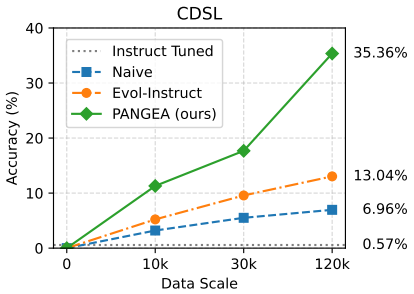

Figure 1: **Performance by data scale.** Result on customized dataset (cf. § 4.1).

[†]Equal corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

In practice, one may resort to fine-tuning APIs on small domain-specific datasets as an initial attempt. However, such direct tuning on a small amount of data is fundamentally prone to overfitting and catastrophic forgetting of previously learned general knowledge, particularly as LLM sizes continue to grow exponentially [17, 25]. Instead of updating the model weights, another line of work equips models with external knowledge at inference. Retrieval-augmented generation (RAG) techniques incorporate relevant documents from external knowledge base into the context, allowing LLMs to address domain-specific queries using up-to-date information [28, 46]. This can mitigate the scarcity of training-data by leveraging large text collections on the fly, though it still requires a large retrievable domain knowledge source. Another direction involves multi-stage alignment or self-refinement pipelines, often guided by human feedback [22, 29, 43]. These approaches enable gradual domain adaptation, though they typically induce training instability, limiting scalability and practical deployment when given a small number of data.

Recently, a promising trend has emerged which leverages LLMs themselves to synthesize new training data, reducing reliance on scarce human annotations [7, 20, 33, 35, 37]. Pioneering works showed that LLMs can bootstrap their instruction-following ability by creating a large set of synthetic instructions and answer pairs [21, 33]. Subsequent efforts extend to diverse domain-specific tasks, data modalities, and integration with external tools [8, 37, 42]. Across such diverse applications, synthetic data generation has proven to be remarkably effective for extending LLM capabilities. Despite ongoing advances, one major limitation persists: the synthetic data distribution is skewed or repetitive, as models tend to generate simpler examples which cannot capture the full complexity of genuine domain-specific data distribution [9, 37].

Our key contributions are summarized as follows:

- We present PANGEA (Projection-based Augmentation with Non-relevant General data for Enhanced domain Adaptation), a fully automated framework for domain-specific data generation that enhances diversity without any additional annotation cost.
- Unlike existing LLM-based synthetic data generators that rely on manually crafted seeds or prompts, our method leverages *a large-scale general-purpose dataset* $\mathcal{D}_g$ to bootstrap the synthesis of high-quality domain-specific examples. In essence, PANGEA extracts diversity from $\mathcal{D}_g$ and integrates it into the structure of *a small domain-specific dataset* $\mathcal{D}_s$, yielding a synthetic dataset that inherits the diversity of $\mathcal{D}_g$ while preserving the domain relevance of $\mathcal{D}_s$.
- As synthetic dataset size increases, our method maintains and even more improves diversity, outperforming existing approaches in both coverage and quality. In this paper, we validate PANGEA on domain-specific benchmarks spanning mathematics, medicine, finance, and a custom domain-specific language task, demonstrating consistently strong performance.

2 Related Works

Recent efforts adapt LLMs to domain expertise using specialized datasets, requiring the collection of sufficiently large domain-specific data. This section summarizes key methodologies for constructing and leveraging such datasets for domain adaptation. See § C.1 for additional related works.

Synthetic data generation. Recent works such as WizardLM [36], WizardMath [20], and WizardCoder [21] generate synthetic data for prompt-based reinforcement learning and instruction tuning. While effective, these methods depend exclusively on LLM generation guided only by source data, often resulting in off-domain or low-quality samples. Other methods like TinyStories [8] and Magpie [38] demonstrate efficient generation pipelines, but are limited by structural simplicity or a lack of domain specificity. Retrieval-augmented techniques such as CRAFT [49] and DataTune [9, 27] enhance generation via external data, yet their performance deteriorates without sufficient in-domain data. ELTEX [23] further emphasizes iterative extraction from raw data, but similarly presupposes the presence of relevant domain signals.

Scalable fine-tuning for domain adaptation. Domain adaptation typically involves fine-tuning LLMs on curated domain-specific datasets. Early scalable approaches introduced domain-specific instruction tuning to improve generalization across tasks [4, 26, 34]. However, acquiring high-quality and sufficiently diverse data remains a significant bottleneck, particularly in specialized domains. Later works involve actively collecting broader domain-specific datasets or constructing new human-annotated datasets [22, 30, 33, 44], yet such approaches are costly and labor-intensive.

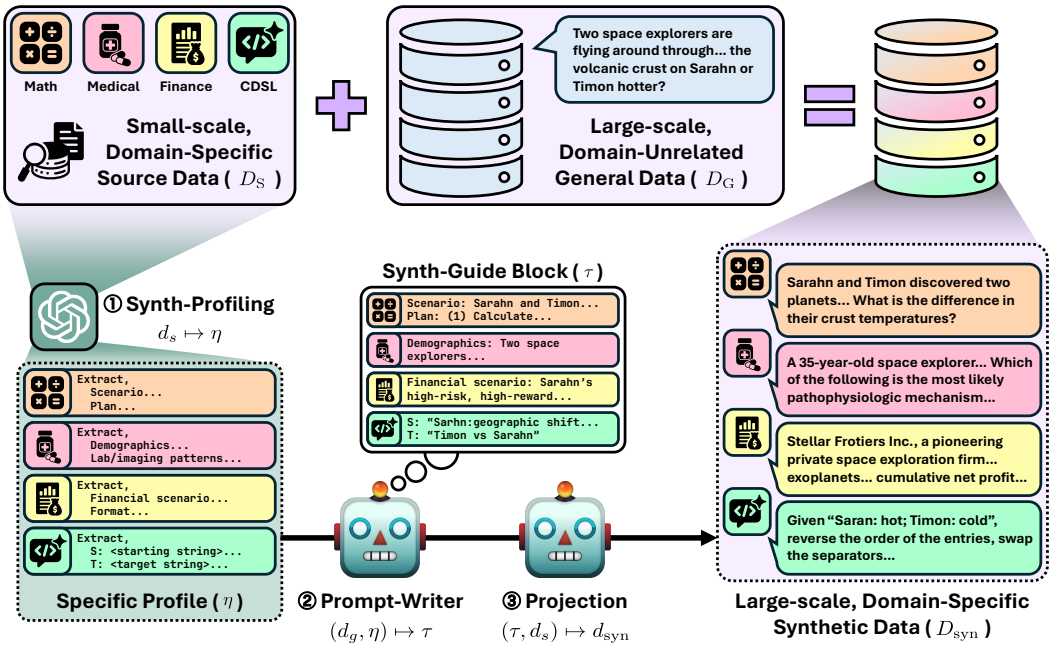

Figure 2: **Overview of PANGEA framework.** It effectively synthesizes domain-specific data using domain-unrelated open-source data, thus even applicable to areas where similar data is scarce.

3 Methodology

In this section, we introduce our method, called PANGEA, which is designed to augment high-quality synthetic data while addressing challenges in domain adaptation and data diversity. Traditional augmentation methods relying on LLMs are limited because they depend solely on the LLM's performance without proper guidance. When applied to augment scarce source data for domain-specific tasks, these methods suffer from issues such as quality degradation, reduced diversity, and high redundancy in the generated synthetic data. Moreover, techniques like Retrieval-Augmented Generation (RAG), which rely on augmenting data with similar or in-domain data, cannot be applied when such relevant data is unavailable, which is the case for most domain-specific tasks.

PANGEA, on the other hand, leverages easily accessible out-of-domain general data to supplement limited domain-specific data by applying guidance distilled from the source data. This strategy effectively enhances the diversity, complexity, and quality of the synthesized dataset, as demonstrated in the following sections. An overview of the proposed methodology is illustrated in Figure 2, which shows how domain-irrelevant data is used to generate synthetic domain-specific data.

3.1 Motivation: deconstructing step-by-step prompts for synthetic data generation

We explore a simple step-by-step prompting strategy as a proof of concept to leverage out-of-domain general data for augmenting domain-specific data. The process involves guiding the model to generate data through the following steps, formatted as a single prompt (see § A.1 for details):

Step 1: General data analysis. In this step, the model is instructed to analyze the given general data and extract relevant details, numerical information, and entities that can be mapped to the target domain. Even if the data itself is domain-irrelevant, such elements may still hold potential as components of synthetic domain-specific data. Which details from the general data should be used is determined based on the target domain, guided by patterns extracted from the source data. These elements are then identified for potential application to the domain.

Step 2: Referencing existing domain data structure. By referencing the existing data structure and details, the model is encouraged to generate data that does not replicate the original data but instead follows the problem's structure, format, length, and difficulty level.

Step 3: Projection and transformation of data. The elements extracted in Step 1 are projected onto the target domain. Guided by the references crafted in Step 2, this step ensures that the projected data is contextually rich and aligned with the target domain.

While this step-by-step prompting approach proved to be effective for generating domain-specific synthetic data using out-of-domain general data, its fixed-format prompt posed challenges in certain domains, such as MedQA. As illustrated in Fig. 3, the model either failed to extract the necessary elements from the general data—resulting in non-informative synthetic outputs—or generated data that lacked sufficient domain-specific information, leading to content irrelevant to the target domain.

As an ablation study, we present the results in Table 1. Compared to the full prompt, (1) simplifying Step 1, (2) simplifying Step 2, and (3) simplifying Step 3 each significantly degrade performance and data quality. We found that the first step—analyzing general data and mapping it to the domain—had a significant impact on performance across various domains. Building on this insight, our PANGEA methodology introduces the Prompt Writer component to further strengthen Step 1, which focuses on leveraging general data.

Table 1: **Ablation study.** Importance of each step measured by performance drop at 30k data scale when removed.

	GSM8K	MedQA	FinQA	Avg.
(1)	29.11	37.75	36.70	34.52
(2)	32.22	39.59	38.27	36.69
(3)	31.84	38.89	37.75	36.16
Full	35.48	39.12	40.19	38.26

3.2 PANGEA: three-stage generation process using the Prompt Writer

Stage 1: Synth profiling. When utilizing irrelevant general data to augment scarce domain-specific datasets, it is crucial to first systematically identify and extract meaningful domain-specific information. Without explicitly characterizing domain-specific attributes upfront, augmented data may suffer from poor alignment with the target domain and reduced effectiveness in the target task.

Algorithm 1 PANGEA

Require: Source dataset $\mathcal{D}_\text{s}$ and general dataset $\mathcal{D}_\text{g}$, where $|\mathcal{D}_\text{s}| \ll |\mathcal{D}_\text{g}|$.
Ensure: Synthetic dataset $\mathcal{D}_\text{syn}$.

$\mathcal{D}_\text{syn} \leftarrow \emptyset$
$\eta \leftarrow \text{LLM}_\text{profiling}(\mathcal{D}_\text{s})$
while enough-data-generated **do**
  Sample $d_\text{s} \in \mathcal{D}_\text{s}$ and $d_\text{g} \in \mathcal{D}_\text{g}$.
  $\tau \leftarrow \text{LLM}_\text{prompt-writer}(d_\text{g}, \eta)$
  $d_\text{syn} \leftarrow \text{LLM}_\text{projection}(\tau, d_\text{s})$
  $\mathcal{D}_\text{syn} \leftarrow \mathcal{D}_\text{syn} \cup \{(d_\text{syn})\}$
end while

To address this challenge, we start with an initial stage within our framework termed *synth profiling*. When the available source dataset is extremely limited—i.e., containing fewer than 100 examples—it becomes feasible to analyze the entire dataset by an LLM or human expert. Such analysis can provide valuable guidance on what kinds of information should be extracted from general datasets to generate synthetic data. Therefore, through our synth profiling framework, we aim to identify the critical informational elements in the target dataset—such as narrative structures or key entities—and use this insight to guide the generation of diverse synthetic data by leveraging various combinations and information scattered across general data sources.

Although information extraction can be performed by a human expert, in our experiments, we utilize a state-of-the-art LLM, OpenAI's o1 [12]. Formally, given a small amount of source dataset $\mathcal{D}_\text{s}$, the synth profiling step can be formulated as follows:

$$\eta = \text{LLM}_\text{profiling}(\mathcal{D}_\text{s}), \tag{1}$$

where η represents the structured domain-specific profile derived through the systematic analysis of the source data and $\text{LLM}_\text{profiling}$ denotes the o1 model which we used. And because η is derived from each specific source dataset, the resulting profiles can vary significantly. For example, in the GSM8K dataset, η encompasses scenario descriptions, symbolic representations, equations, and related reasoning structures. In contrast, for the MedQA dataset, η includes demographic information, clinical timelines, key medical events, laboratory results, and other domain-specific elements. These specialized profiles help guide the generation of more relevant and informative synthetic data. Refer to § A.2 to see how we prompt $\text{LLM}_\text{profiling}$ to generate η from $\mathcal{D}_\text{s}$.

By explicitly conducting the profiling stage, we provide a clear blueprint for how general data should be transformed for effective synthetic data generation. Here, the structured guidance derived from η facilitates the creation of high-quality synthetic samples, substantially improving both the relevance and diversity of the generated data.

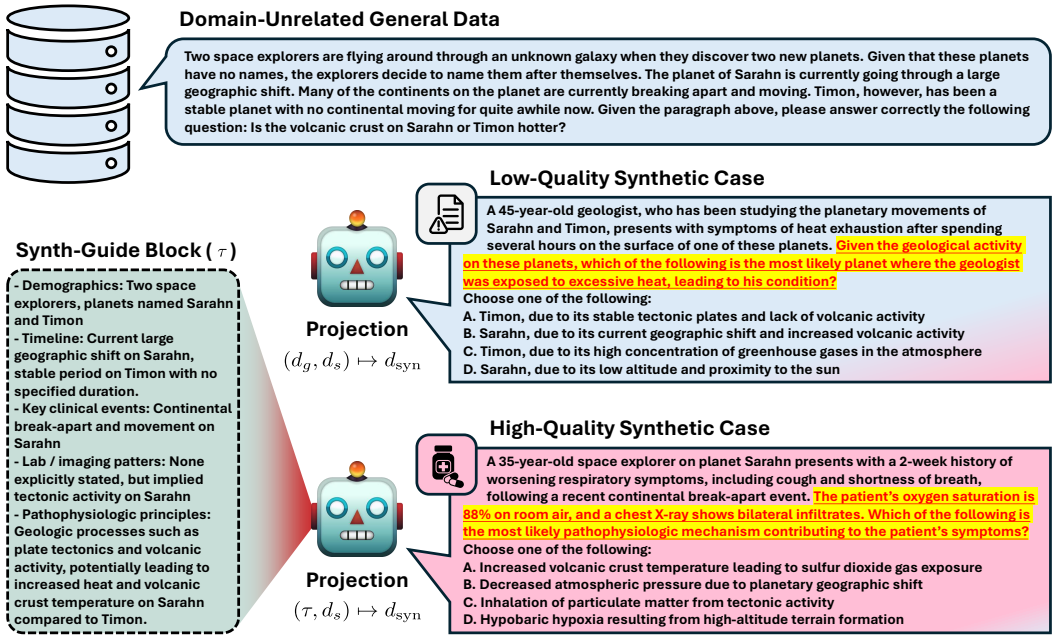

Figure 3: **End-to-end flow diagram of PANGEA Stages 2 and 3.** A general data passage is first distilled into a structured Synth-Guide Block (τ), and then projected into a domain-specific question format (e.g., MedQA). A naive projection of general data lacks clinical realism, whereas data generated through Stage 3 exhibits high realism and domain relevance.

Stage 2: General data analysis with the Prompt Writer. After extracting the domain-specific profile η in Stage 1, we employ a frozen LLM, referred to as the Prompt Writer ($\text{LLM}_{\text{prompt-writer}}$), to systematically extract structured, domain-specific Synth-Guide Blocks (τ) from irrelevant, out-of-domain general data instances (d_{g}), using η as guidance. Each instance d_{g} is drawn from the irrelevant general dataset $\mathcal{D}_{\text{g}}$ (i.e., $d_{\text{g}} \in \mathcal{D}_{\text{g}}$). Simply put, τ can be understood as the information within each d_{g} that either corresponds to or is relevant to η, organized into a prompt-like format. Since τ is generated from a wide range of different d_{g} samples, numerous and diverse combinations of τ are produced, which in turn ensures the diversity of the resulting synthetic data. Specifically, this process can be written as follow:

$$\tau = \text{LLM}_{\text{prompt-writer}}(d_{\text{g}}, \eta), \quad \text{where} \quad d_{\text{g}} \in \mathcal{D}_{\text{g}}. \tag{2}$$

Here, we utilized `Llama3.3-70B-Instruct` [10] as for $\text{LLM}_{\text{prompt-writer}}$ in our experiments. As described in Stage 1, since the profile η is closely tied to the characteristics of each source dataset, it can vary significantly across tasks. Consequently, the τ extracted from the general dataset is tailored to reflect the specific requirements of each target domain. This flexibility sets our method apart from traditional synthetic data generation approaches that rely on source-similar data [9, 49]. Instead, our framework leverages a single, broad general dataset to support a wide range of domain-specific tasks, offering a more scalable and adaptable solution.

Also, this explicit structuring of information—rather than relying on implicit prompting—offers clear and consistent guidance for data generation. As a result, this approach helps transform each d_{g} into data that is more suitable for learning in the target domain, reducing the risk of generating irrelevant content as observed in the failure cases discussed in § 3.1. It consequently improves both the efficiency and quality of the generated synthetic dataset. Refer to § A.3 to see how we prompt $\text{LLM}_{\text{prompt-writer}}$ to generate τ.

Stage 3: Domain-specific synthetic data generation. In the final stage—Stage 3—we generate synthetic data using the Synth-Guide Blocks τ produced in Stage 2. As previously mentioned, each τ is a summarized prompt that contains key information required for the target domain. By using the information in τ to generate data in the appropriate domain-specific format, we can produce synthetic samples that are more aligned with the learning needs of the downstream task. To ensure

Table 2: **Main results comparing synthetic data generation frameworks.** Accuracy (%) on four benchmarks is shown for increasing amounts of synthetic data (10k, 30k, and 120k). Our proposed PANGEA consistently outperforms the baselines across all data scales, with especially large margins at higher data volumes. All evaluations are conducted in a zero-shot setting.

# Synthetic	Method	Benchmarks				Avg. (impr.)
		GSM8K (↑)	MedQA (↑)	FinQA (↑)	CDSL (↑)	
-	Pre-trained	5.69	28.91	6.02	0.00	10.16
	Instruction-tuned	45.03	37.31	26.68	0.57	27.40
10k	Naive	26.91	35.42	24.06	3.20	22.40 (+12.24)
	Evol-Instruct	27.36	36.29	26.68	5.22	23.89 (+13.73)
	PANGEA (ours)	**32.52**	**37.78**	**36.44**	**11.30**	**29.51 (+19.35)**
30k	Naive	34.72	34.24	27.46	5.51	25.48 (+15.32)
	Evol-Instruct	32.51	38.09	29.64	9.57	27.45 (+17.29)
	PANGEA (ours)	**38.36**	**39.98**	**41.41**	**17.68**	**34.36 (+24.20)**
120k	Naive	42.68	38.02	32.43	6.96	30.02 (+19.86)
	Evol-Instruct	38.73	42.34	33.74	13.04	31.96 (+21.80)
	PANGEA (ours)	**48.61**	**44.62**	**50.22**	**35.36**	**44.70 (+34.54)**

that the generated data follows the correct format, we project each τ using both the domain-specific source data $d_s \in \mathcal{D}_s$ and a large language model, denoted as $\text{LLM}_{\text{projection}}$. At this point, to maximize the data diversity and improve training efficiency, we partition the collection of τ during the projection process so that each domain-specific sample d_s is assigned with $N = \frac{|\mathcal{D}_g|}{|\mathcal{D}_s|}$ different general data instances. Then, for each d_s with its partition $\mathcal{P}_s$ containing the N Synth-Guide Blocks, we perform projection using $\text{LLM}_{\text{projection}}$ as follows:

$$d^j_{\text{syn}} = \text{LLM}_{\text{projection}}(\tau_j, d_s), \quad \text{where} \quad \tau_j \in \mathcal{P}_s \text{ and } d_s \in \mathcal{D}_s. \tag{3}$$

In experiments, we utilized `Llama3.3-70B-Instruct` as for $\text{LLM}_{\text{projection}}$. After we generate a partial synthetic dataset using each d_s and its corresponding partition, we gather all of them to make our new synthetic dataset $\mathcal{D}_{\text{syn}}$ to train a domain-specific task. Fig. 3 illustrates examples of τ generated in Stage 2, as well as a qualitative comparison between synthetic data generated with and without the use of τ in Stage 3. The figure highlights how leveraging τ leads to more relevant and structured outputs, while directly using d_g without τ often results in misaligned synthetic data. As we proceed with instruction fine-tuning using $\mathcal{D}_{\text{syn}}$, we additionally generate corresponding labels for each synthetic instance. Unless otherwise specified, we reuse $\text{LLM}_{\text{projection}}$ to annotate the dataset. The entire process of the PANGEA framework is summarized in Algorithm 1. Additionally, example prompts used in our experiments are provided in § A.4 and § A.5.

4 Experiments

4.1 Main results

To evaluate the effectiveness of our proposed PANGEA framework, we conduct experiments using established benchmark datasets (GSM8K [5], MedQA [13], FinQA [3]). We consider the following scenarios: 1) A *data-scarce* setting where only 100 domain-specific examples are available. To simulate this, we randomly select 100 samples from the training split. 2) A setting where the general data is *entirely unrelated* to the target domain. To construct this, we manually exclude all math-, medical-, and finance-related entries from the CoT-Collection database [14]. We further explore an extreme domain-specific scenario by introducing a Custom Domain-Specific Language (CDSL), an artificially constructed programming language; the accuracy of pre-trained models on this language is indeed zero. Following the same design procedure as in the previous experiments, we generate synthetic data using 100 domain-specific examples along with a domain-unrelated general dataset. Please refer to § B.2 for more details on our experimental setup.

Table 2 summarizes the comparative results of synthetic data generation frameworks using the `Llama3.2-1B` model. When fine-tuned on 10k, 30k, and 120k synthetic data generated by each

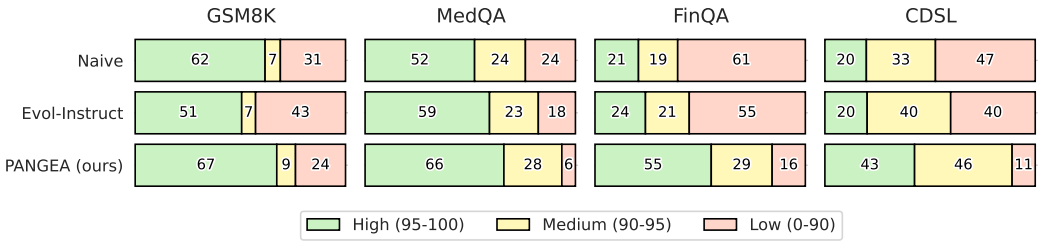

Figure 4: **o1 scores for data quality analysis.** The proportion (%) of generated data evaluated as High (95-100), Medium (90-95), or Low (0-90) quality by OpenAI's o1 model.

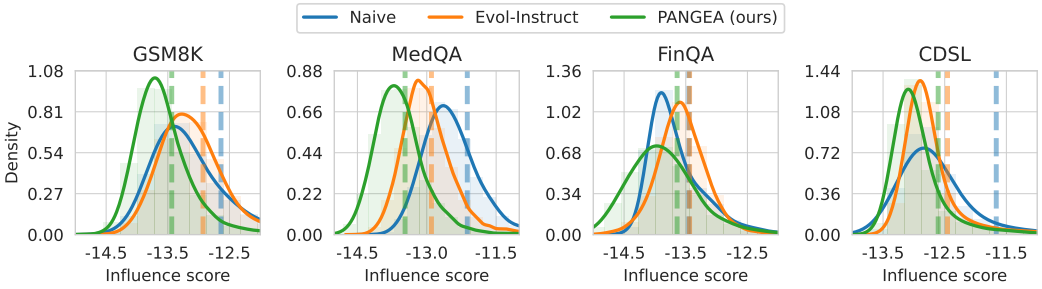

Figure 5: **Influence scores for data quality analysis.** Each histogram shows the distribution of data points for each augmentation method over influence score ranges. A kernel density estimation curve is overlaid as a solid line for better visualization, and the vertical dashed line denotes the mean.

methodology, our proposed PANGEA framework consistently outperforms the baselines across all benchmarks and data scales. At the 10k scale, PANGEA already demonstrates substantial improvements over the Naive and Evol-Instruct methods, with PANGEA achieving the highest performance. As the data scale increases to 30k and 120k, both PANGEA and PANGEA continue to outperform the baselines, particularly on the GSM8K and FinQA benchmarks, showing notable performance gains. Notably, the `Llama3.2-1B` model fine-tuned with our framework (35.36) even surpasses the instruction-tuned `Llama3.3-70B` (33.62) and `Qwen2.5-72B` (23.76) models on the CDSL benchmark, demonstrating its effectiveness in building domain-specific small language models.

Having demonstrated the strong performance of PANGEA, we now turn to an analysis of its success along two key aspects: *1) quality*, which assesses how realistic and high-quality the generated synthetic data is (cf. § 4.2), and *2) diversity*, which measures how novel the generated data is compared to the original sources (cf. § 4.3). Ideally, synthetic data should be both distinct from the original data and of high quality; extensive experiments confirm that PANGEA excels in both aspects.

4.2 Analysis of synthetic data quality

We conduct three complementary analyses: a qualitative review of key data characteristics, a quantitative influence-score analysis of training impact, and a human evaluation.

o1 score. We randomly sample 5,000 synthetic examples generated by each augmentation method and evaluated them using OpenAI's o1 model across five criteria: difficulty, formatting, problem relevance, language quality, and clarity. Consequently, each sample received a quality score ranging from 0 to 100, providing a detailed view of the overall quality of the generated data (cf. § B.3 for more experimental details). Fig. 4 summarizes the results across three score ranges: High (95–100), Medium (90–95), and Low (0–90). It clearly demonstrates that PANGEA consistently achieves higher scores across all benchmarks compared to the baseline methods, suggesting that o1 judged the generated data to be of higher domain-specific quality. The difference is especially pronounced in FinQA, where the data structure is more complex.

Influence score. We also measure the influence score [16], which quantifies the impact of each data point on the model's training. It allows us to measure how much each data point contributed to

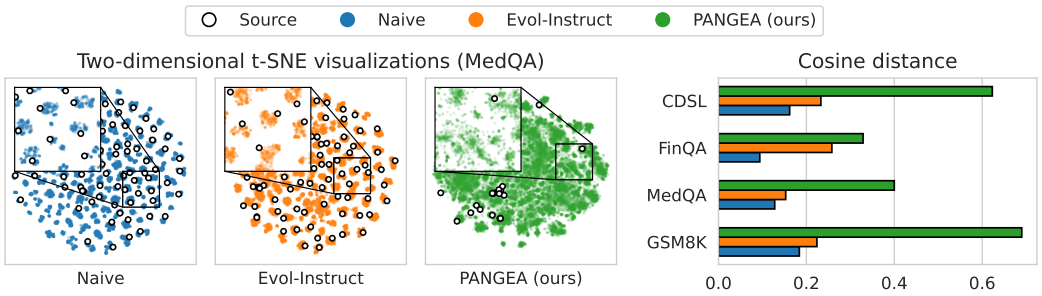

Figure 6: **Sentence embeddings for diversity analysis.** The first three plots show t-SNE visualizations of data generated by the Naive, Evol-Instruct, and PANGEA methods (colored dots), which are augmented from the source data (white dots), on the MedQA dataset. The bar plot on the right summarizes the average pairwise cosine distance of generated samples across datasets.

model performance. Fig. 5 illustrates the distribution of influence scores in a sign-log scale for each augmentation method across the datasets (GSM8K, MedQA, FinQA, and CDSL). Our proposed PANGEA approach consistently produces data points with higher influence scores, particularly in more specialized domains like CDSL, compared to the Naive and Evol-Instruct methods. It suggests that our framework generates data points that have a greater impact on model training, leading to more effective learning and improved performance in downstream tasks.

Human evaluation. To complement automatic metrics, we ran a human preference study with 57 graduate-level annotators. For each domain (GSM8K, FinQA, MedQA, CDSL), annotators ranked three synthetic candidates (Naive, Evol-Instruct, PANGEA) as *Best/Second/Worst* on 100 randomly sampled items. PANGEA received the highest *Best* preference on all four domains (avg. 50.6%), and differences were statistically significant by both χ^2 and Friedman tests ($p < 0.01$). Inter-annotator agreement was moderate (Krippendorff's $\alpha = 0.61$). Full protocol, per-domain distributions, and statistics are provided in § B.6.

4.3 Analysis of synthetic data diversity

PANGEA augments source domain-specific data by leveraging irrelevant general data, enabling it to generate more *diverse* synthetic data compared to augmentation methods that rely solely on limited source-domain samples. Recognizing the critical role of data diversity in model performance [2], we assess the diversity of the generated synthetic data using the `all-mpnet-base-v2` embedding model from the Sentence Transformers library [24].

t-SNE visualization. We begin by qualitatively examining the diversity of the synthetic data through a t-SNE [32] visualization of the embeddings. Fig. 6 shows two-dimensional embeddings of the synthetic data generated by each augmentation method (colored dots), alongside the original domain-specific source data (white dots), to assess how broadly the synthetic data points are distributed relative to their source. For the Naive and Evol-Instruct baselines, the synthetic data points are mostly concentrated around the source data, indicating limited diversity. In contrast, PANGEA produces more widely dispersed embeddings, reflecting a higher degree of diversity in synthetic data.

Cosine similarity. To quantitatively evaluate data diversity, we compute the average pairwise cosine distance between embeddings of the synthetic data. For each of the 100 source instances, we generate 100 synthetic samples and calculate the pairwise cosine similarity among them. As summarized in Fig. 6, PANGEA consistently achieves significantly higher average cosine distance than the baselines, indicating superior semantic diversity across all benchmark datasets. As a result, the data generated by PANGEA is not only more diverse but also of high quality (as previously discussed in § 4.2), demonstrating the overall effectiveness of the method.

4.4 Scalability & Robustness

Scalability & Robustness. We evaluate PANGEA along two axes, data scarcity and model capacity, to verify both robustness and scalability. First, to test robustness under extreme scarcity, we vary

Table 3: **Model and seed scale results.** (a) shows how performance changes as the number of seed data increases from 10 to 100 under a fixed 10k synthetic set, while (b) reports the 8B-scale results with `Llama-3.1-8B`, where PANGEA outperforms all baseline models.

(a) **Seed-size ablation**. Robust down to 20–40; drop only at 10 due to an under-specified profile.

Method (#seed)	GSM8K	MedQA	FinQA	CDSL	Avg.
Naive (100)	26.91	35.42	24.06	3.20	22.40
Evol-Instruct (100)	27.36	36.29	26.68	5.22	23.89
PANGEA (100)	**32.52**	**37.78**	**36.44**	**11.30**	**29.51**
PANGEA (80)	32.91	37.34	36.37	11.01	29.41
PANGEA (40)	31.84	36.10	35.21	10.15	28.33
PANGEA (20)	31.21	35.79	34.83	8.70	27.63
PANGEA (10)	29.28	33.27	28.31	3.77	23.66

(b) **8B-scale setup.** With 8B scale, PANGEA steadily surpasses all baselines across every benchmark.

Method	GSM8K	MedQA	FinQA	CDSL
Pre-trained	48.75	39.31	25.63	1.61
Instruct-tuned	85.62	64.10	64.95	2.32
Naive	81.35	55.93	48.78	15.65
Evol-Instruct	79.08	60.02	53.88	17.42
PANGEA	**86.47**	**64.51**	**65.31**	**25.91**

the number of seed data from 100 to 10 while keeping the synthetic set fixed at 10k. As shown in Table 3a, PANGEA remains strong with as few as 20 seeds and degrades gracefully only at 10, confirming that the quality of *Synth-Profiling* governs performance under limited seed data.

Second, to assess scaling with backbone capacity, we use `Llama-3.1-8B` and find that PANGEA yields substantial gains over the pre-trained model and consistently surpasses all baselines on every benchmark (Table 3b), showing that the framework remains effective even for larger models.

Domain-agnostic prompting & multilingual transfer. We replace task-specific prompts with unified, domain-agnostic templates across all stages and find that PANGEA (agnostic) continues to outperform non-projection baselines, showing only a small drop compared to the original domain-tailored prompts (Table 4). To evaluate cross-lingual robustness, we project English D_g into Korean to synthesize 10k CSAT items from official seeds. PANGEA achieves the highest overall score and is the only method that successfully solves *Hard* items (Table 5), demonstrating that the profiling–guide–projection pipeline remains effective even when the source and target languages differ.

Table 4: **Domain-agnostic result.** Unified templates beat baselines, with a small gap to original.

Method	GSM8K	MedQA	FinQA	CDSL
Pre-trained	5.69	28.91	6.02	0.00
Instruction-tuned	45.03	37.31	26.68	0.57
Naive	26.91	35.42	24.06	3.20
Evol-Instruct	27.36	36.29	26.68	5.22
PANGEA (agnostic)	31.70	36.39	34.74	9.25
PANGEA (original)	**32.52**	**37.78**	**36.44**	**11.30**

Table 5: **Korean CSAT.** PANGEA achieves the best total and uniquely solves *Hard* items.

Method	Easy	Normal	Hard	Total
Pre-trained	40.00	13.64	0.00	13
Instruction-tuned	60.00	31.82	0.00	27
Naive	60.00	13.64	0.00	15
Evol-Instruct	40.00	31.82	0.00	22
PANGEA	**60.00**	**36.36**	**5.26**	**34**

4.5 Discussion

Ablation of prompt writer. As part of our ablation study, we investigate the impact of the prompt writer component within the PANGEA pipeline, aiming to show that it plays a crucial role in effectively utilizing task-irrelevant general data. To this end, we evaluate PANGEA *without* the prompt writer, i.e., the proof-of-concept model introduced in § 3.1. Even without the prompt writer, leveraging general data itself yields strong performance, surpassing both the Naive and Evol-Instruct baselines that do not utilize general data. Moreover, integrating the prompt writer brings an additional gain. In the case of our 120k-scale experimental setup in Table 2, PANGEA improves average performance from 38.25% to 44.70%, further enhancing our core idea of leveraging general data by providing a guideline for effectively synthesizing domain-specific data from it. Details in § B.2.

Generalization across architectures. To verify the generalization capability of PANGEA across different model architectures, we conducted additional experiments using `Gemma2-2B` [31], `Qwen2.5-1.5B` [39], and the `DeepSeek-R1-Distill-Llama-8B` [11]. The reasoning model is evaluated under the 10k-scale setup, while the others are evaluated under the 30k-scale setup. As shown in Table 6, PANGEA consistently delivers substantial improvements over the pre-trained weights across all backbones.

Table 6: **Architecture generalization.** Benchmark results under our 30k-scale experimental setup (reasoning at 10k scale). PANGEA consistently displays significant performance improvements across all architectures. In particular, the reasoning model utilizes the `DeepSeek-R1` for labeling. The average performance improvement (+) is also shown for each model.

Model	Method	Benchmarks				
		GSM8K (↑)	MedQA (↑)	FinQA (↑)	CDSL (↑)	Avg. (impr.)
Gemma2-2B	Pre-trained	1.66	27.81	7.23	0.28	9.00
	PANGEA	37.16	37.47	38.46	15.87	32.74 (+23.74)
Qwen2.5-1.5B	Pre-trained	3.98	26.45	27.83	0.00	14.57
	PANGEA	41.92	32.25	40.84	16.48	32.87 (+18.30)
Llama3.2-1B	Pre-trained	5.69	28.91	6.02	0.00	10.16
	PANGEA	38.36	39.98	41.41	17.68	34.36 (+24.20)
DeepSeek-R1-Distill-Llama-8B	Pre-trained	85.29	57.09	61.29	6.96	52.66
	PANGEA	88.91	66.53	69.75	28.70	63.47 (+10.81)

Comparison to domain-specialized methods. To further highlight PANGEA's strength, we present comparative results with MuMath [40] and UltraMedical [45] datasets, which were constructed using domain-specific strategies for the mathematics and medical domains, respectively. Table 7 summarizes the results under our 30k-scale experimental setup. Notably, the proposed surpasses such domain-specialized baselines, further validating the high quality and diversity of the data it generates.

Table 7: **Comparative results.** Benchmark results on GSM8K and MedQA.

Method	GSM8K	MedQA
MuMath	30.25	-
UltraMedical	-	36.71
PANGEA (ours)	**32.52**	**37.78**

Further comparison with retrieve-then-transform approaches. It is worth noting that our extreme data scarcity scenario, combined with reliance on out-of-domain general data, poses significant challenges for generating domain-specific data. While we already conducted a comprehensive evaluation against the Naive and Evol-Instruct baselines in our main experiments, there exists another line of work that also leverages large-scale general-purpose data, similar to our approach. For example, DataTune [9] improves automatic dataset generation by utilizing existing, publicly available datasets. However, their method assumes that the general dataset is task-relevant but misaligned; that is, they retrieve examples from *a domain-relevant* $\mathcal{D}_g$ from the outset. This assumption does not hold in our scenario, where $\mathcal{D}_g$ is entirely *domain-unrelated*, making their approach less applicable. Indeed, on MedQA under our 10k-scale experimental setup, DataTune achieved only 34.95%, performing even worse than 35.42% of the Naive baseline. This clearly highlights the limitations of existing retrieval-based synthetic data generation methods in our problem setting—where $\mathcal{D}_s$ *is extremely scarce and there is no relevant examples in* $\mathcal{D}_g$. In contrast, our approach proves to be significantly more effective under such challenging conditions.

5 Conclusion

In this paper, we propose the PANGEA framework, which effectively generates high-quality synthetic data even in domain-specific environments with extremely scarce data. By transforming unrelevant general data into domain-specific synthetic data through a three-stage framework, PANGEA achieves superior diversity and quality compared to existing methods. Extensive benchmarking and quality evaluation experiments demonstrate that the data generated by PANGEA consistently maintains high quality and diversity, enabling efficient synthetic data generation and improved model performance even in highly specialized domain settings.

Limitation and future direction. Currently, PANGEA requires carefully designed prompts across its three stages, which entails a certain level of human effort. Reducing this manual overhead while evolving the framework into a more efficient and flexible solution for synthetic data generation presents a promising direction for future work.

Acknowledgement

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) ((No.RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST); No.RS-2024-00509279, Global AI Frontier Lab; No.2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics; No.RS-2025-02219317, AI Star Fellowship(Kookmin University); the MSIT(Ministry of Science, ICT), Korea, under the Global Research Support Program in the Digital Field program(IITP-2024-RS-2024-00417958) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation); and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2021-NR056917; NRF-2022R1A5A708390812). This work was supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea) & Gwangju Metropolitan City.

References

[1] Yung-Chieh Chan, George Pu, Apaar Shanker, Parth Suresh, Penn Jenks, John Heyer, and Sam Denton. Balancing cost and effectiveness of synthetic data generation strategies for llms. *arXiv preprint arXiv:2409.19759*, 2024.

[2] 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.

[3] Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, 2021.

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, and et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[6] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

[7] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

[8] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

[9] Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. Better synthetic data by retrieving and transforming existing datasets. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6453–6466, 2024.

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[11] 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.

[12] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[13] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[14] Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=D7omx8QyFP.

[15] Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36:47669–47681, 2023.

[16] Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[17] Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*, 2024.

[18] Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*, 2023.

[19] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv preprint arXiv:2406.15126*, 2024.

[20] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mMPMHWOdOy.

[21] Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=UnUwSIgK5W.

[22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[23] Arina Razmyslovich, Kseniia Murasheva, Sofia Sedlova, Julien Capitaine, and Eugene Dmitriev. Eltex: A framework for domain-driven synthetic data generation. *arXiv preprint arXiv:2503.15055*, 2025.

[24] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

[25] Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*, 2024.

[26] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*, 2022.

[27] Minju Seo, Jinheon Baek, James Thorne, and Sung Ju Hwang. Retrieval-augmented data augmentation for low-resource domain tasks. *arXiv preprint arXiv:2402.13482*, 2024.

[28] Zirui Song, Bin Yan, Yuhan Liu, Miao Fang, Mingzhe Li, Rui Yan, and Xiuying Chen. Injecting domain-specific knowledge into large language models: a comprehensive survey. *arXiv preprint arXiv:2502.10708*, 2025.

[29] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36: 2511–2565, 2023.

[30] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[31] Gemma Team, 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.

[32] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[33] 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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, 2023.

[34] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[35] 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. *arXiv preprint arXiv:2304.12244*, 2023.

[36] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CfXh93NDgH.

[37] Yiming Xu, Chenghao Zheng, Jialu Liu, and et al. Magpie: Synthetic data for instruction tuning with no human labels. *arXiv preprint arXiv:2402.08525*, 2024.

[38] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing, 2024. URL https://arxiv.org/abs/2406.08464.

[39] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[40] Weihao You, Shuo Yin, Xudong Zhao, Zhilong Ji, Guoqiang Zhong, and Jinfeng Bai. Mumath: Multi-perspective data augmentation for mathematical reasoning in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2932–2958, 2024.

[41] Xiao Yu, Zexian Zhang, Feifei Niu, Xing Hu, Xin Xia, and John Grundy. What makes a high-quality training dataset for large language models: A practitioners' perspective. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 656–668, 2024.

[42] Arthur Yuan and et al. Craft: Distilling tool use for llm reasoning. *arXiv preprint arXiv:2404.09855*, 2024.

[43] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

[44] Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, et al. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*, 2023.

[45] Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, et al. Ultramedical: Building specialized generalists in biomedicine. In *Conference on Neural Information Processing Systems*, 2024.

[46] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

[47] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. *arXiv preprint arXiv:2405.01470*, 2024.

[48] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P Xing, et al. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*, 2023.

[49] Ingo Ziegler, Abdullatif Köksal, Desmond Elliott, and Hinrich Schütze. Craft your dataset: Task-specific synthetic dataset generation through corpus retrieval and augmentation. *arXiv preprint arXiv:2409.02098*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The primary claims outlined in § 1 precisely align with our paper's contributions and scope, as substantiated by comprehensive empirical results and analyses detailed in § 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our method in § 5, particularly the requirement of human effort for precise prompt design, and we suggest directions for future improvements.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper primarily focuses on proposing and validating a novel synthetic data generation methodology through empirical experiments, without involving theoretical proofs or formal theoretical assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We fully disclose all necessary information for reproducing the main experimental results, including dataset descriptions, experimental setup, evaluation metrics, hyperparameters, and prompt details across the three-stage framework, clearly presented in § A and § B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide detailed experimental settings, including prompt specifications in § A, comprehensive methodological and algorithmic details as well as data descriptions used in our main experiments in § B.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experiments details are provided in § B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments do not explicitly include error bars or statistical significance metrics; instead, we focus primarily on comparative performance improvements and data quality evaluations across different methods and benchmarks, as detailed in § 4 and § B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational resource details are provided in § B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Potential positive and negative societal impacts of our work are discussed in § 5 and § B.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We describe appropriate safeguards for responsible use and release of our synthetic data generation framework, including controlled access and clear usage guidelines, detailed in § 5 and § C.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and models utilized are clearly listed in § B.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our research does not introduce any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our study does not include crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This study involves no human subjects, so IRB approval is not applicable.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our proposed PANGEA framework leverages LLMs for synthetic data generation. For writing, LLMs were solely used for grammar correction; all original ideas and core methodologies presented in this paper are entirely our own.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Supplementary materials

We provide detailed prompts utilized throughout the PANGEA framework. We first present the step-by-step prompts used in our initial motivation study (§ 3.1). Subsequently, we provide the structured prompts employed within our three-stage framework, leveraging the Prompt Writer (§ 3.2). Finally, we include the annotation prompts for each benchmark dataset evaluated in our experiments.

A.1 Step-by-step prompts for motivation study (§ 3.1)

Our initial proof-of-concept (Motivation) prompts were structured as follows. In this stage, data was generated in an end-to-end manner, where d_{g_i} represents **the irrelevant general data** instance and d_{s_i} denotes **the domain-specific source data** instance.

GSM8K prompt

```
You are an expert at transforming general questions into domain-specific,
math-related questions.  Your task is to generate only the transformed
math question without including any answers or solutions.
---
### STEP 1:  Analyze and Understand the General Question
- Fully understand the general question's context, key concepts, and
quantitative elements.
- Identify core topics (e.g., measurement, comparison, probability) and
specific details that inspire mathematical transformations.
---
### STEP 2:  Refer to Domain-Specific Question (for Inspiration Only)
- Use the domain-specific question to understand common formulations in
mathematical contexts.
- Extract core mathematical concepts and focus on transformation, not
duplication.
- Frame questions encouraging mathematical reasoning (e.g., multi-step
calculations, logic, real-world applications).
---
### STEP 3:  Generate the Transformed Math Question
- Create an original math question retaining general question context.
- Integrate scenario, characters, or objects clearly.
- Introduce mathematical challenges explicitly requiring calculations,
comparisons, or probabilistic/logical reasoning.
- Avoid solutions or numerical answers entirely.
---
Response Rule:  Generate only the T̈ransformed Domain Questionẅithout any
answer, explanation, or solution.
---
Output Format:
Transformed Domain Question:  [Write your transformed question here.]
---
Task:
- General Question:
"""
{d_{g_i}}
"""
- Domain Question: """
{d_{s_i}}
"""
```

Figure 7: **Motivation GSM8K prompt.** Prompt used for GSM8K Benchmark Generation.

MedQA prompt

```
You are an expert in transforming General Data and existing MedQA-style
clinical questions into high-quality, diverse, and complex MedQA-style clinical
questions.  Generate realistic USMLE-style multiple-choice questions.
---
### Step 1:  General Data Analysis & Clinical Scenario Transformation
- Convert General Data into clinical elements (demographics, lab values, timelines,
events, exposures, physiological processes).
- Transform key events into realistic clinical scenarios (accidents, exposures,
temporal shifts).
- Reinterpret events into clinical contexts (symptoms, history, diagnostics).
- Ensure clinical relevance (emergency medicine, infectious disease, cardiology).
### Step 2:  Referencing Existing Domain Question Structure
- Refer to existing MedQA questions (structure, relevance, complexity).
- Recreate and diversify scenario, symptom progression, diagnoses.
- Introduce new events, mechanisms, injuries, exposures, patient history.
### Step 3:  Generate Transformed Data (Scenario + MCQs)
1.  Develop Complex Clinical Scenario
- Merge General Data with MedQA style for realistic patient scenarios
(age, gender, travel, medication, family history, exposures).
- Add disease progression, comorbidities, side effects, diagnostic errors.
2.  Create New Multiple-Choice Questions
- Integrate General Data and MedQA clinical elements for unique scenario.
- Provide 4 answer options (1 correct, 3 relevant distractors).
Step 4:  Final Output (Restrictions)
- Ensure accuracy, originality, logical consistency.
- Strictly output New Clinical Question and 4 options only.
- No explanations, comments, answers, reasoning.
### Output Format:
New Clinical Question:
Question:  <Clinical question here>
Choose one of the following:
A. <Option 1>
B. <Option 2>
C. <Option 3>
D. <Option 4>
---
### Key Considerations:
- Convert General Data into medical data (timelines, labs, exposures).
- Reference existing MedQA questions but create original scenario.
- Diversify question types (diagnosis, treatment, prognosis, mechanisms).
- Ensure distractors are realistic and clinically relevant.
---
Task
General Question:
"""
{d_{g_i}}
"""
existing MedQA-style clinical question:
"""
{d_{s_i}}
"""
```

Figure 8: **Motivation MedQA prompt.** Prompt used for MedQA Benchmark Generation.

FinQA prompt

```
You are an expert in transforming General Data (general knowledge, everyday
questions, or non-financial text) into structured Financial Data and generating
high-quality, diverse, and complex financial reasoning problems.
--
Your Enhanced Approach Must Include:
### Step 1.  General Data → Financial Context Mapping
- Convert General Data into a finance-oriented scenario.
- Identify relevant financial events, business implications, investment scenarios,
or economic contexts that relate to the General Data.
- Ensure financial scenario is logically consistent, realistic, and distinct from
Existing FinQA-style financial reasoning question.
### Step 2.  Financial Context → Structured Financial Data Generation
- Create realistic financial data (revenue, costs, tax, investments, stocks).
- Include at least two different financial variables for complexity.
- Generate data solely from financial context derived from General Data.
### Step 3.  Financial Data → Unique FinQA-style Question Generation
- Formulate multi-step calculations, financial analysis, market evaluation,
investment decision-making, or risk assessment question.
- Ensure complete originality and no copying from existing FinQA questions.
- Use new business scenarios derived from General Data.
---
### Output Restriction:
- Strictly generate New Financial Context and Question only.
- Maintain similar length and structure as Existing FinQA-style question.
- Do NOT reference existing company names or financial details.
- Ensure financial scenario derived entirely from General Data.
- No explanations, comments, or notes.
--
### Output Format:
New Financial Context and Question:
Please answer the given financial question based on the context.
Context:  <Generated financial context with relevant numerical table data>
Question:  <Generated financial reasoning question>
---
### Enhanced Data Processing Flow:
- Extract key themes from General Data.
- Map themes to financial events (corporate, market, investments).
- Generate structured Financial Data (revenue, costs, indicators).
- Create unique financial reasoning problem (complexity, originality).
--
### Strict Constraints:
- DO NOT copy company names or figures from Existing FinQA question.
- All data derived solely from General Data.
- Ensure complete originality.
- Diversify numerical relationships and economic insights.
---
Task
General Data:
"""
{d_{g_i}}
"""
Existing FinQA-style financial reasoning question:
"""
{d_{s_i}}
"""
```

Figure 9: **Motivation FinQA prompt.** Prompt used for FinQA Benchmark Generation.

CDSL prompt

Step-by-Step Prompt Example for Custom DSL:
You are an expert in converting General Data and a Reference String-Manipulation Task into a fresh benchmark item that matches the reference style while remaining fully original.

Inputs:
General Data:
"""
{d_{g_i}}
"""
Reference String Task:
"""
{d_{s_i}}
"""

STEP 1 · General Data Analysis & Building-Block Construction *(keep private)*
1. Invent the initial string S -- 5-50 chars, quoted, contains $\geq$ 1 separator.
2. Write one Goal sentence describing the intended transformation of S.
3. Select 3-6 operations (split, join, reverse, str(n), reverse_tokens, replace, substr, upper, lower, append, prepend).
4. Draft the op chain (op(arg) in order).
5. Compute the target string T by mentally executing the chain.
Keep S, Goal, Ops, Chain, and T hidden; never expose them.

STEP 2 · Referencing Existing Task Structure
- Examine Reference Task for leading verb/phrase and sentence length & tone.
- Craft one instruction sentence $\leq$ 60 words, starting with that leading verb/phrase, quotes S exactly once, uses vivid verbs, no ops or intermediate math.
--
STEP 3 · Projection & Final Benchmark Generation
Goal: Publish final benchmark item in strict two-line format.
Header: New String-Transformation Task:
Problem-statement sentence (verbatim from Step 2).
Desired Output: "<T>" (directly after problem sentence).
No Extras: Exactly two lines shown, no comments or explanations.

HARD RULES
Reveal nothing from Step 1. Exactly three printed lines (header + two content lines). Task requires 3-6 implicit operations. Never copy wording or digits from the reference; mirror only opener and sentence length.

Figure 10: **Motivation CDSL prompt.** Prompt used for CDSL Benchmark Generation.

A.2 PANGEA: Profiling prompt (§ 3.2)

The prompt used in the initial profiling stage of our three-stage PANGEA framework (§ 3.2) is presented below. In this stage, we provided our scarce domain-specific source data ($\mathcal{D}_s$) to OpenAI's o1 model and requested a detailed analysis of the dataset. Specifically, the model was instructed to identify key characteristics, structural patterns, and thematic elements of the provided data.

Iteration 1: Initial analysis
"<Insert domain-specific source dataset $\mathcal{D}_s$ here>"

User:
You are an analyst tasked with transforming irrelevant general-domain data into {domain_name} - specific data. Carefully analyze the 100 provided domain-specific source examples. Identify their key characteristics, structural patterns, and thematic elements. Suggest what types of information should be extracted or transformed from unrelated general datasets to create useful domain_name data.

o1:

<model's initial analysis and suggestions>

Iteration 2: Refinement based on generated data

User:
Following your suggested approach, we generated this synthetic example:

"<first synthetic data example>"

Given this outcome, what additional information or refinements do you suggest extracting or transforming from general data to better align with the domain_name domain?

o1:

<model's refined analysis and further suggestions>

⋮

(The iterative process continues until desired data quality is achieved.)

Figure 11: **Synth-profiling prompt.** Example dialogue illustrating the iterative profiling process used in the PANGEA framework.

An illustrative example of the profiling prompt dialogue is presented in Figure 11. Based on the initial analysis, the model generated structured guidelines describing how irrelevant general-domain data could be effectively transformed into high-quality, domain-specific synthetic data. This profiling process was iterative: synthetic examples were first generated following the guidelines provided by the model, their quality was evaluated, and instructions were progressively refined based on feedback if the generated data did not sufficiently meet the desired criteria. This iterative, conversational approach ensured continuous improvement of instructions and consistent generation of high-quality synthetic data.

A.3 PANGEA: Prompt writer prompts (§ 3.2)

The prompt used in the second stage of the PANGEA framework (§ 3.2), which employs the Prompt Writer to generate structured Synth-Guide Blocks (τ), is presented below. Only irrelevant general data is provided as input to the Prompt Writer.

GSM8K Prompt-Writer's prompt

```
You are an expert at distilling everyday descriptions into concise,
GSM8K-ready
quantitative building blocks.  Your output feeds next stage, converting
these
blocks into a 2-4 sentence word problem solvable by a 5th- to 8th-grader
in
multiple arithmetic or logical steps.
### Inputs
General Question (raw text)
"""
{d_{q_i}}
"""
---------------------------------
### Choose a Kid-Friendly Scenario
- Restate one everyday clause students can picture
(shopping, chores, snacks, simple travel).
- Remove unnecessary details.
### Select 3-5 Meaningful Numbers
- Prefer numbers in General Question.
- Invent realistic values if needed, supporting clear multi-step
solution.
- Use everyday units (dollars, minutes, km, items, °C, simple percents).
### Define Symbols & Units
- Assign symbols A, B, C, (D, E) with brief label and explicit unit.
### Write 4-5 Simple Equations
- Use only +, -, ×, ÷, %, or single-step unit conversions.
- No new numbers; use chosen symbols or intermediate results.
### State the Target
- Specify final symbol to solve and its numeric meaning.
### Outline Step-by-Step Plan
- Provide one bullet per equation, in order (4-5 steps).
---------------------------------
### Output Format (use EXACTLY this template)
Quantitative Building Blocks:
- Scenario - <one kid-friendly clause>
- Symbols - <A = value unit - label>, <B = ...>, <C = ...>[, <D = ...>[,
<E = ...>]]
- Equations - (1) ...  = ...
(2) ...  = ...
(3) ...  = ...
(4) ...  = ...
[(5) ...  = ...]
- Target - <T = description of numeric goal>
- Plan - (1) ...  → (2) ...  → (3) ...  → (4) ...  [→ (5) ...]
### Hard Restrictions & Key Considerations
- Scenario familiar, everyday, age-appropriate; no fantasy/sci-fi.
- All numbers appear in Scenario and Equations.
- Exactly 4 or 5 equations, requiring at least four steps.
- Do NOT write/solve full word problem.
- No examples, explanations, or commentary.
- Begin exactly with "Quantitative Building Blocks:" and follow template
strictly.
```

Figure 12: **Prompt Writer prompt for GSM8K.** This prompt instructs the model to analyze irrelevant general data and explicitly extract structured information relevant to arithmetic reasoning, including scenarios, equations, symbols, and solution plans for GSM8K.

MedQA Prompt-Writer's prompt

```
You are an expert at transforming General Data into clinically useful Clinical
building blocks.

## General-to-Clinical Mapping
General Data:
"""
{d_{g_i}}
"""
---
### 1.  Analyze General Data and extract every relevant
- numbers & ranges
- logical/causal relationships
- temporal progressions
- physical principles, legal scenarios, or major events.
### 2.  Convert these elements into clinical building blocks, such as
- patient demographics, timelines, physiologic/metabolic processes
- lab values, vital-sign ranges, exposures, accidents, disease
progressions.
### 3.  Output ONLY the transformed clinical building blocks in the exact
format below:
---
### Output Format
Clinical Building Blocks:
- Demographics - ...
- Timeline - ...
- Key clinical events - ...
- Lab / imaging patterns - ...
- Pathophysiologic principles - ...

### Hard Restrictions
- Do NOT generate questions, answer options, or explanations.
- Output must begin exactly with "Clinical Building Blocks:"
followed by the bullet list above.
```

Figure 13: **Prompt Writer prompt for MedQA.** This prompt guides the model to analyze unrelated general data and extract structured clinical elements tailored specifically for MedQA.

FinQA Prompt-Writer's prompt

```
You are an expert in translating General Data (non-financial facts or everyday
scenarios) into a realistic, detailed, logically consistent financial scenario.
---------------------------------
### Task:
1.  Carefully analyze the provided General Data.
2.  Clearly translate the General Data into a concise financial scenario, explicitly
highlighting:
- Specific and realistic financial implications or investment contexts.
- Potential corporate decisions, economic outcomes, or market
environments logically derived from the data.
3.  Scenario must be distinct, detailed, clear enough for later use in Stage 2
(structured financial data and FinQA-style questions).
4.  Do NOT produce numerical tables, numeric values, or financial questions now.
5.  Avoid closely replicating illustrative examples or specific contexts from prior
instructions; generate entirely original scenario uniquely based on provided
General Data.
---------------------------------
### Output (follow EXACTLY):
Financial Scenario:
<Clearly describe your original financial scenario derived from the provided
General Data.  Include explicit descriptions of financial events, investment
opportunities, economic contexts, or corporate implications that are realistic
and logically consistent.>
---------------------------------
### Provided General Data:
"""
{d_{g_i}}
"""
```

Figure 14: **Prompt Writer prompt for FinQA.** This prompt directs the model to carefully analyze irrelevant general-domain data and extract a financial scenario specifically for FinQA.

CDSL Prompt-Writer's prompt

```
You are an expert at distilling everyday descriptions into concise
String-Flow Building Blocks.  Your output will feed the next stage, which
converts these blocks into a complete String-Flow DSL "recipe"
(code) that the interpreter can run.
--
## Stage 1 - General Description → String-Flow Building Blocks
General Text (raw):
"""
{d_{g_i}}
"""
### 1.  Define the Initial String S
- Single quoted string, 5-50 chars.
- Include at least one separator or symbols.
- Remove irrelevant details.
### 2.  State the Desired End-State T (plain words)
- Describe in one sentence the intended final form of S
(e.g., "reverse the items and join them with hyphens").
### 3.  Select 3-6 Essential Operations
- Choose from split · join · reverse_str · reverse_tokens · replace
("" to delete) · substr · upper · lower · append · prepend.
- Briefly list each operation with arguments if needed
(delimiter, replacement text, start/length, etc.).
- set (loading S) always implied, omit from the list.
### 4.  Draft an Operation Chain
- List operations in order:  (1) ...  → (2) ...  → (3) ...  (3-6 steps).
- Each node:  operation(arg).
### 5.  Give the Target String T (concrete)
- Write exact resulting string after all operations (Stage 2
correctness checking).
---
### Output Format (use exactly this)
String-Flow Building Blocks:
- S - "<initial string>"
- Goal - <plain-language description of desired result>
- Ops - <op1(arg)>, <op2(arg)>, <op3(arg)>[, ...]
- Chain -(1) ...  → (2) ...  → (3) ...  [→ (4) ...  → (5) ...  → (6) ...]
- T - "<target string>"
### Hard Rules & Tips
- Ops and Chain must match exactly in order and content.
- No unnecessary operations or arguments.
- substr indices:  0-based [start, length].
- upper/lower convert ASCII A-Z/a-z only.
- Everyday, age-appropriate contexts; no complex jargon.
- Do not add examples, explanations, or commentary.
- Begin your answer with "String-Flow Building Blocks:"
exactly as above, nothing else.
```

Figure 15: **Prompt Writer prompt for CDSL.** This prompt explicitly instructs the model to analyze unrelated general-domain data and extract structured elements required by the CDSL benchmark.

A.4 PANGEA: Projection prompts (§ 3.2)

Using the following prompts, we performed the projection described in § 3.2. Here, d_{s_i} denotes **the domain-specific source data** instance, and the **Synth-Guide Block** (τ) generated by the Prompt Writer is provided as block, facilitating the generation of the final synthetic data.

GSM8K projection prompt

```
You are an expert in crafting GSM8K-style math word problems that require
clear, multi-step quantitative reasoning.

Quantitative Blocks → Transformed Math Question

Inputs
- Quantitative Building Blocks
"""
{block}
"""
- Reference GSM8K Question (for tone & length only)
"""
{d_{s_i}}
"""
---
### Adapt Structure & Inject Realism
- Mirror only tone, length, complexity of reference; never copy
content.
- Length:  exactly 2 to 4 crisp sentences.
- Seamlessly weave Scenario, Entities, Numbers, Required relationships
from Building Blocks into natural story.
- Embed each numeric value exactly once; avoid repeating numbers.
- No intermediate calculations shown or mentioned.

### Compose the New Math Problem (Projection)
1.  Preserve Context
- Keep original setting/characters (or logical variants).
2.  Embed Every Given Number
- All numeric values from Blocks must appear and be essential.
3.  Demand Clear Multi-Step Challenge
- Solver must implicitly perform at least five distinct steps.
4.  Avoid Superfluous Data
- Do not invent extra numbers; recombine given numbers if needed.
5.  Maintain Numerical Consistency
- Respect units; conversions must be inferable (e.g., min→h, cm→m).
---
### Final Output & Hard Restrictions
- Output ONLY the problem text--no answers, hints, explanations.
- Begin exactly with "Transformed Domain Question:".
- Do NOT reuse exact wording or numeric values from reference GSM8K.
- Problem must require at least five distinct steps to solve.
- Length:  exactly 2 to 4 concise sentences.
- Never repeat numbers; never expose arithmetic/intermediate results.
---
### Output Template
Transformed Domain Question:  <sentence 1> <sentence 2> <sentence 3
(optional)> <sentence 4 (optional and must be explicit question)>
```

Figure 16: **Projection prompt for GSM8K.** This prompt explicitly directs the model to project the Synth-Guide Block onto GSM8K-specific source data, ensuring the generated synthetic examples.

MedQA projection prompt

```
You are an expert in crafting complex, USMLE-style (NBME-format) multiple-choicemedical &
clinical questions.

## Inputs
- Clinical Building Blocks
"""
{block}
"""
- Existing MedQA-style clinical question (structure reference only)
"""
{d_{s_i}}
"""
---
## Reference & Diversify Existing Question Structure
- Use existing question for format, difficulty, diagnostic flow only;
do NOT copy storyline.
- Recreate/diversify clinical context, symptom progression, diagnoses.
- Add clinically relevant events (injury mechanisms, toxic exposures,
detailed history) from Clinical Building Blocks.

### Projection & MCQ Creation (USMLE Blueprint)
1.  Develop Complex Clinical Scenario
- Concise patient vignette (age, sex → chief-complaint → HPI → PE/labs).
- Insert ≥2 essential numeric data (vitals, labs, imaging).
- Integrate USMLE clues (pathognomonic signs, imaging findings, signature lab patterns,
POD-complications, drug toxicity).
- Include dynamic elements (progression, comorbidities, side-effects).
- Exposure history details (≥2:  intensity mg/m³, duration, frequency,
route, protective equipment).

2.  Create New Multiple-Choice Question
- Stem:  ask either (a) most likely diagnosis or (b) next best step
/ appropriate management (only one).
- Provide exactly 4 answer options (A-D):
- One correct answer (current guidelines).
- Three high-quality realistic distractors (differentials or plausible
suboptimal choices).
- All options USMLE-standard medical concepts/actions.

### Final Output & Hard Restrictions
- Verify clinical accuracy, originality, data coherence, consistency.
- Output Restriction:  ONLY New Clinical Question & 4 answer options.
- NO explanations, rationales, answer keys, extra text.

### Output Format
New Clinical Question:
Question:  <clinical question text>
Choose one of the following:
A. <Option 1>
B. <Option 2>
C. <Option 3>
D. <Option 4>
---
### Key Considerations (USMLE Focus--do not output)
- Embed guiding numbers (Na+ 128 mEq/L, TSH 12 µIU/mL, EF 35%).
- Diagnosis distractors:  common look-alike diseases.
- Management distractors:  reasonable suboptimal or contraindicated
therapies.
- Follow current USMLE/NBME consensus guidelines.
```

Figure 17: **Projection prompt for MedQA.** This prompt explicitly instructs the model to transform the Synth-Guide Block into synthetic examples aligned with MedQA.

FinQA projection prompt

```
You are an expert in creating original, realistic, and complex FinQA-style financial reasoning
problems based on provided inputs.
---
### Task:
Using these inputs:
- Financial Scenario (from General Data)
- Existing FinQA-style Question (REFERENCE ONLY)

Generate a completely new, high-quality financial context and multi-step reasoning question
explicitly projecting Stage 1 Financial Scenario into a financial reporting narrative matching
Existing FinQA-style Question's style, complexity, and detail.
---
### STRICT Requirements:
(1) Main Narrative Context:
- Emulate Existing Question's narrative complexity with lengthy, complex
sentences and detailed explanations.
- Detailed definitions/explanations of ≥2 financial terms or acronyms.
- Use formal, passive-voice financial-reporting language ("was recognized,"
"were recorded," "is anticipated to be realized").
- Include ≥1 explicit note-reference clause ("net of tax," "see Note X").
- Describe financial events qualitatively, without repeating numbers.
(2) Financial Table (Markdown Format):
- Precisely replicate Existing Question's table dimensions (rows, columns).
- Use identical unit descriptor ("$ in millions," "$ in billions").
- Original yet realistic financial labels/numerical data closely
matching Existing Question's magnitude and complexity.
- Explicit markdown formatting matching Existing Question.
(3) Follow-up Narrative (Mandatory):
- Exactly match Existing Question's length, complexity, style, depth.
- Provide detailed qualitative insights without repeating table numbers.
- Explicitly address profitability trends, cost control effectiveness,
strategic impacts, or risk management considerations.
(4) Advanced, Multi-step Financial Reasoning Question:
- Require ≥2 sophisticated calculation steps (ratios, multi-year growth,
weighted averages, financial adjustments).
- Exactly match Existing Question's complexity, brevity, and clarity.
---
### STRICTLY AVOID:
- Copying exact wording, company names, numbers from Existing Question.
- Overly simplistic or ambiguous single-step questions.
- Repeating illustrative examples; all content must be original.
- Explanations, solutions, answers, additional commentary.
- Omitting or generalizing Follow-up Narrative.
---
### OUTPUT FORMAT:
New Financial Context and Question:
Context:  <Explicitly structured narrative context exactly matching Existing FinQA Question's
detailed sentence complexity, narrative length, passive voice, financial explanations, note
references, and qualitative financial descriptions.>
<Markdown-formatted financial table precisely matching dimensions, complexity, numerical
magnitude, digit length, exact unit descriptor, with entirely original data from Stage 1
scenario.>
<Detailed Follow-up Narrative explicitly matching Existing Question's length, complexity,
style, analytical depth; provide financial insights without repeating table numbers.>
Question:  <Explicitly stated advanced-level multi-step financial
reasoning question exactly matching Existing Question's brevity,
complexity, clarity, requiring ≥2 calculation steps.>
---
### Provided Inputs:
Financial Scenario:
"""
{block}
"""
Existing FinQA-style question (REFERENCE ONLY):
"""
{d_{s_i}}
"""
```

Figure 18: **Projection prompt for FinQA.** This prompt instructs the model to project the Synth-Guide Block onto FinQA-specific source data to generate high-quality synthetic data for FinQA.

CDSL projection prompt

```
You are an expert at crafting concise, multi-step string-manipulation
tasks
for benchmark datasets.
### Inputs
- String-Flow Building Blocks
"""
{block}
"""
- Existing Generic String Task (style and format reference only)
"""
{d_{s_i}}
"""
---
### Use the Reference ONLY for Format & Tone
- Match length, phrasing style, and difficulty, but do NOT copy
storyline or wording.
- Recreate transformation using the Building Blocks.
+Style-Mirroring Rules
- Begin your sentence with the same leading verb or phrase
used in the reference task.
### Paraphrase Requirement
- Turn bland "Goal" line into vivid instruction (≤ 60 words).
- Mention initial string S exactly as given (including quotes,
separators, etc.).
- Do not reveal internal operation names.
- Prefer concrete verbs (strip, reorder, reverse, join).
---
### Create the New String-Transformation Task
1.  Problem Statement
- One sentence satisfying Paraphrase Requirement.
2.  Desired Output
- Present T in quotes on its own line, labeled
"Desired Output:".
### Output Restrictions
- Output only lines shown below--no explanations, extra text.
- Begin with exact header "New String-Transformation Task:".
- Preserve line breaks exactly.
---
### Output Format (copy exactly)
New String-Transformation Task:
<Problem Statement>

Desired Output:  "<T>"
```

Figure 19: **Projection prompt for CDSL.** This prompt explicitly guides the model in projecting the Synth-Guide Block onto CDSL-specific source data to produce high-quality synthetic data for CDSL.

A.5 Annotation prompts (§ 3.2)

The annotation prompt is used to collect responses for the generated synthetic data. By specifying a structured output format, this prompt facilitates both the model's chain-of-thought reasoning and the parsing of responses, thereby simplifying performance evaluation.

GSM8K annotation prompt

```
You are an expert in mathematical problem solving.  Your task is to
provide a detailed and logically sound solution to the given math
problem.  Each response must be accurate and based on rigorous
mathematical reasoning.

### Response Rule
- Provide clear step-by-step solutions, including relevant
mathematical principles and theorems.
- Explore multiple solution methods if applicable; compare their
efficiency.
- Verify final answers through appropriate validation methods.
- Conclude clearly with "The final answer is:  [Final Answer]"
under the header **Answer**.

### Output Format
- **Answer**:  [Provide detailed mathematical explanation here,
including step-by-step derivations, logical reasoning, and
verification of calculations.  Conclude explicitly with
"The answer is:  [Final Answer]".]

---
### Your Task:
- **Question:** {Question}
```

Figure 20: **Annotation prompt for GSM8K.** Prompt used to collect annotation responses for the GSM8K synthetic data.

MedQA annotation prompt

```
You are an expert in medical question answering.  Your task is to provide
a detailed, evidence-based response to the given multiple-choice medical
question, accurate and aligned with up-to-date medical guidelines.

### Response Rule
- Provide comprehensive explanation including relevant clinical
reasoning.
- Analyze all answer choices, explaining correctness or incorrectness.
- Conclude clearly with final answer format:
"The answer is:  [Answer Letter].  [Answer Option]" under the header
**Answer**.

### Output Format
- **Answer**:  [Provide detailed medical explanation here, including
clinical reasoning, differential diagnosis, and evidence-based
references.
Conclude explicitly with "The answer is:  [Answer Letter].  [Answer
Option]".]

---
### Your Task:
- **Question:** {Question}
```

Figure 21: **Annotation prompt for MedQA.** Prompt used to collect annotation responses for the MedQA synthetic data.

FinQA annotation prompt

```
You are an expert in financial question answering.  Provide detailed,
evidence-based responses to given financial questions.  Each response
must be accurate, concise, and based on financial principles, accounting
standards, and quantitative analysis.

### Response Rule
- Provide step-by-step breakdown of calculations, relevant formulas,
and financial reasoning.
- Clearly explain each step and how data points derive the final answer.
- Explicitly state and justify necessary assumptions.
- Conclude with Final Answer clearly boxed:  \boxed{{}}.

### Output Format
- **Answer**:
[Provide detailed financial explanation here, including step-by-step
calculations, financial reasoning, and key insights.]

The answer is:  \[ \boxed{{[Final Answer]}} \]

---
### Your Task:
- **Question:** {Question}
```

Figure 22: **Annotation prompt for FinQA.** Prompt used to collect annotation responses for the FinQA synthetic data.

CDSL annotation prompt

```
You are an expert programming assistant familiar with a custom
interpreter
called StringFlowInterpreter (also known as SousChef).  This interpreter
uses cooking-themed commands to manipulate strings step-by-step.

Interpreter Command Quick Reference:
- `pour` "string":  Set main string (broth) to specified string.
- `slice "delimiter"`:  Split broth into tokens (ingredients) by
delimiter.
- `stir "delimiter"`:  Join tokens (ingredients) into broth using
delimiter.
- `flip`:  Reverse broth string.
- `toss`:  Reverse order of tokens (ingredients).
- `season "old" "new"`:  Replace occurrences of substring old with new.
- `fillet start length`:  Extract substring from broth (start, length).
- `flambe`:  Convert broth to uppercase.
- `simmer`:  Convert broth to lowercase.
- `garnish "string"`:  Append specified string to broth.
- `plate "string"`:  Prepend specified string to broth.
- `taste_then "substring" label`:  If broth contains substring, jump to
label.
- `move_to label`:  Unconditionally jump to specified label.
- `label LABEL_NAME`:  Define jump destination label.
- `serve`:  Print current broth and end execution.
---
Given cooking-themed instruction, follow these steps:
1.  Analyze and generate final command sequence according to
StringFlowInterpreter rules.
2.  Explain step-by-step in detail how final command sequence was
derived.
3.  Conclude with clearly indicated final generated command sequence:

### output format:

<your step by step explanation>

The answer is:
"""
<line by line final command sequence>
"""

Important:
- Do NOT include original provided example; only step-by-step
explanation clearly followed by final command sequence.
- In each command line, Do NOT include comments.

### Your Task:
- **Question:** {Question}
```

Figure 23: **Annotation prompt for CDSL.** Prompt used to collect annotation responses for the CDSL synthetic data. An additional detailed task description was provided to ensure accurate annotation by the model due to the novel and domain-specific nature of CDSL tasks.

B Experimental details

B.1 Models and datasets

Models. The pre-trained weights used in our experiments are summarized below.

- **`Llama3.2-1B`**: https://huggingface.co/meta-llama/Llama-3.2-1B, licensed under Llama3.2 Community License.[3]
- **`Qwen2.5-1.5B`**: https://huggingface.co/Qwen/Qwen2.5-1.5B, licensed under Apache 2.0 License.[4]
- **`Gemma-2B`**: https://huggingface.co/google/gemma-2b, licensed under Gemma License.[5]
- **`Llama3.3-70B-Instruct`**: https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct, licensed under Llama 3.3 Community License.[6]
- **`Qwen2.5-72B-Instruct`**: https://huggingface.co/Qwen/Qwen2.5-72B-Instruct, licensed under Apache 2.0 License.[4]
- **`DeepSeek-V3-0324`**: https://huggingface.co/deepseek-ai/DeepSeek-V3-0324, licensed under MIT License.[7]

Both our motivation study and the proposed three-stage PANGEA framework employed the `Llama3.3-70B-Instruct` model for generating synthetic data. To ensure fair comparisons, we generated synthetic data for all baseline methods (Naive, Evol-Instruct, and DataTune) using the same `Llama3.3-70B-Instruct` model.

Annotations of the synthetic data were also performed using `Llama-3.3-70B-Instruct`, except for the CDSL. For CDSL, annotation accuracy with `Llama3.3-70B-Instruct` reached only 33.62%, despite detailed task instructions. Thus, we specifically employed the `DeepSeek-V3` model to annotate synthetic data for the CDSL benchmark.

Due to budget constraints, subsequent experiments utilized smaller pre-trained base models: `Llama3.2-1B`, `Qwen2.5-1.5B`, and `Gemma-2B`. We conducted supervised fine-tuning (SFT) using the Alpaca template [30], training all models with full-parameter updates.

Datasets. The datasets used in our experiments are summarized below:

- **COT-Collection** [14] consists of 1.84 million Chain-of-Thought (CoT) augmented samples across 1,060 tasks derived from the Flan Collection, designed to induce CoT reasoning capabilities into language models.
- **GSM8K** [5] contains 8,792 arithmetic reasoning questions requiring multi-step logical reasoning and numerical calculations.
- **MedQA** [13] consists of over 12,000 standardized medical exam (USMLE) questions, assessing models' medical knowledge and clinical decision-making abilities.
- **FinQA** [3] includes over 8,000 complex financial reasoning problems extracted from real-world financial reports, testing numerical analysis and financial comprehension.
- **CDSL** is introduced to evaluate models under severe data scarcity and extreme domain novelty. It involves transforming input strings into target outputs using a novel interpreter, `StringFlowInterpreter`, with entirely unseen commands and functions. The dataset contains only 100 manually curated training examples and 345 test instances.

Since our experimental setting involves an extreme scenario of data scarcity in highly specialized domains without any related general-domain data available, we constructed our irrelevant general dataset ($\mathcal{D}_g$) from the CoT-Collection. To ensure strict irrelevance, we manually removed samples related to mathematics, medical, finance, and coding tasks by using

[3] https://www.llama.com/llama3_2/license/
[4] https://www.apache.org/licenses/LICENSE-2.0
[5] https://ai.google.dev/gemma/terms
[6] https://www.llama.com/llama3_3/license/
[7] https://github.com/deepseek-ai/DeepSeek-V3-0324/blob/main/LICENSE

`Llama-3.3-70B-Instruct` and `Qwen/Qwen2.5-72B-Instruct` across five different random seeds.

For the GSM8K, MedQA, and FinQA benchmarks, we randomly selected 100 samples from each dataset's training split to form our domain-specific source dataset. In the case of the CDSL, which includes 100 manually curated training examples and 343 test samples specifically constructed for evaluating models in novel and specialized domain settings, we directly utilized the entire set of 100 training samples as our domain-specific source dataset ($\mathcal{D}_s$).

Our proposed benchmark, Custom Domain-Specific Language (CDSL), is specifically designed to evaluate language models' capabilities in highly specialized domains and extreme data scarcity settings. The task involves transforming an input string into a desired output by generating executable code conforming to a novel interpreter (`StringFlowInterpreter`, detailed in Algorithm 2). Unlike conventional programming languages or commonly used scripting syntax, our interpreter features uniquely crafted cooking-themed commands, enhancing domain specificity and novelty.

This cooking metaphor-based interpreter maintains internal state representations (such as `broth` for strings and `ingredients` for tokens) and executes instructions through commands named after culinary actions (e.g., `slice`, `stir`, `flambe`). This distinctive structure not only emphasizes domain novelty but also ensures the synthetic generation process must capture highly specific instructions and contextually relevant transformations, effectively evaluating the adaptability of synthetic data augmentation methodologies in novel environments.

Specifically, for annotating the CDSL benchmark, we employed the `Deepseek-V3` model. The generated code, enclosed within triple backticks (```), was extracted and compiled. Instances where the code failed to compile or produced outputs inconsistent with the labels were filtered out.

Algorithm 2 CDSL: Interpreter

```python
class StringFlowInterpreter:
    def __init__(self):
        self.broth = ""          # str_val
        self.ingredients = []    # tokens
        self.stations = {}       # labels
        self.pc = 0
        self.recipe = []         # program
        self.cooking = True      # running

    def prep(self, recipe_str):              # parse → prep
        for line in recipe_str.strip().split('\n'):
            dish = line.strip()
            if not dish:
                continue
            if re.match(r'^[\w]+:$', dish):
                self.stations[dish[:-1]] = len(self.recipe)
            else:
                step, *args = self.dice(dish)
                self.recipe.append((step, args))

    def dice(self, line):                    # _split_line → dice
        return re.findall(r'\"[^\"]*\"|\S+', line)

    def cook(self):                          # run → cook
        self.pc = 0
        while self.cooking and self.pc < len(self.recipe):
            step, args = self.recipe[self.pc]
            self.saute(step, args)           # execute → saute
            self.pc += 1
        return self.broth

    def saute(self, step, args):
        if step == "pour":                   # SET
            self.broth = self.peel(args[0])

        elif step == "slice":                # SPLIT
            delim = self.peel(args[0])
            self.ingredients = self.broth.split(delim)

        elif step == "stir":                 # JOIN
            delim = self.peel(args[0])
            self.broth = delim.join(self.ingredients)

        elif step == "flip":                 # REVERSE STR
            self.broth = self.broth[::-1]

        elif step == "toss":                 # REVERSE TOKENS
            self.ingredients.reverse()

        elif step == "season":               # REPLACE
            old, new = map(self.peel, args[:2])
            self.broth = self.broth.replace(old, new)

        elif step == "fillet":               # SUBSTR
            start, length = map(int, args)
            self.broth = self.broth[start:start + length]

        elif step == "flambe":               # UPPER
            self.broth = self.broth.upper()

        elif step == "simmer":               # LOWER
            self.broth = self.broth.lower()

        elif step == "garnish":              # APPEND
            self.broth += self.peel(args[0])

        elif step == "plate":                # PREPEND
            self.broth = self.peel(args[0]) + self.broth

        elif step == "taste_then":           # IFSTRCONTAINS
            substr, label = self.peel(args[0]), args[1]
            if substr in self.broth:
                self.pc = self.stations[label] - 1

        elif step == "move_to":              # GOTO
            self.pc = self.stations[args[0]] - 1

        elif step == "serve":                # PRINT
            print(self.broth)
            self.cooking = False

        else:
            raise ValueError(f"Unknown step: {step}")

    def peel(self, s):                       # _strip_quotes → peel
        return s[1:-1] if s.startswith('"') and s.endswith('"') else s
```

B.2 Main experiments detail & example

In our main paper, we selected Naive and Evol-Instruct as baseline methods for comparison, as other approaches such as DataTune [9] showed limited effectiveness in our scenario (e.g., achieving only 34.95% accuracy on MedQA at the 10k scale when using irrelevant general data). Thus, we assessed how effectively these selected baselines could generate diverse and high-quality synthetic data from general-domain data.

We generated synthetic datasets at scales of 10k, 30k, and 120k, subsequently evaluating improvements in model performance across various benchmarks. Additionally, we also evaluated our initial proof-of-concept (motivation study) across the same three data scales. All benchmarks were tested under a zero-shot setting, where performance was assessed by directly parsing model-generated responses and comparing them to the ground-truth labels.

Table 8: **Comparison of synthetic data generation frameworks.** Accuracy (%) on four benchmarks across synthetic data scales (10k, 30k, 120k). Our PANGEA framework consistently outperforms baselines, with larger improvements at increased data scales. All evaluations are performed in a zero-shot setting.

# Synthetic	Method	Benchmarks				
		GSM8K (↑)	MedQA (↑)	FinQA (↑)	CDSL (↑)	Avg. (impr.)
-	Pre-trained	5.69	28.91	6.02	0.00	10.16
	Instruction-tuned	45.03	37.31	26.68	0.57	27.40
10k	Naive	26.91	35.42	24.06	3.20	22.40 (+12.24)
	Evol-Instruct	27.36	36.29	26.68	5.22	23.89 (+13.73)
	Motivation (ours)	32.14	36.76	32.08	8.89	27.47 (+17.31)
	PANGEA (ours)	**32.52**	**37.78**	**36.44**	**11.30**	**29.51 (+19.35)**
30k	Naive	34.72	34.24	27.46	5.51	25.48 (+15.32)
	Evol-Instruct	32.51	38.09	29.64	9.57	27.45 (+17.29)
	Motivation (ours)	35.48	39.12	40.19	10.15	31.24 (+21.08)
	PANGEA (ours)	**38.36**	**39.98**	**41.41**	**17.68**	**34.36 (+24.20)**
120k	Naive	42.68	38.02	32.43	6.96	30.02 (+19.86)
	Evol-Instruct	38.73	42.34	33.74	13.04	31.96 (+21.80)
	Motivation (ours)	46.02	42.58	43.07	21.31	38.25 (+28.09)
	PANGEA (ours)	**48.61**	**44.62**	**50.22**	**35.36**	**44.70 (+34.54)**

As demonstrated in Table 8, our proposed three-stage **PANGEA** framework consistently achieves superior performance across all benchmarks and data scales compared to baseline methods, exhibiting particularly significant improvements as the data scale increases. Notably, even our initial proof-of-concept (Motivation) outperforms the Evol-Instruct baseline on most benchmarks, indicating its robustness and effectiveness in leveraging irrelevant general-domain data. However, the performance gap between Motivation and Evol-Instruct is marginal in some cases, highlighting the enhanced capability and stability introduced by our structured three-stage PANGEA framework.

Resource & hyperparameters. For our main experiments, we utilized four NVIDIA H100 GPUs. All models were trained with Fully Sharded Data Parallel (FSDP) using the AdamW optimizer with a learning rate of 1×10^{-5}, a total batch size (batch size $\times$ gradient accumulation steps) of 64, a sequence length of 4096, and a cosine scheduler with a warm-up ratio of 0.03 over three epochs. Inference with `Deepseek-V3` was conducted using its external API, whereas inference for Llama models was performed locally.

Example. Here, we present examples of synthetic data generated using our proposed three-stage PANGEA framework. Table 9 provides examples of the general-domain data alongside the corresponding **Synth-Guide Blocks** generated for each benchmark by the Prompt Writer. Table 10 illustrates how these Synth-Guide Blocks are subsequently projected into **synthetic** examples.

Table 9: **Examples of Synth-Guide Blocks (τ).** Synth-Guide Blocks generated from general data, tailored specifically to each benchmark using our proposed three-stage PANGEA framework.

General Data: Two space explorers discover two unnamed planets in an unknown galaxy and name them after themselves. Sarahn is undergoing major tectonic shifts, while Timon is geologically stable. *Question*: Which planet's volcanic crust is hotter?	
GSM8K	**Scenario:** Sarahn and Timon are comparing the crust temperature of their planets: Sarahn's crust is 48 °C, Timon's crust is half as hot (0.5 ×), and they note that the temperature difference equals twice (2 ×) Timon's crust temperature. **Symbols:** - A = 48 °C – Sarahn's crust temperature - B = 0.5 – "half" factor - C = 2 – "twice" factor **Equations:** (1) D = A × B (2) T = C × D (3) E = A – D (4) T = E **Target:** T = difference between the planets' crust temperatures (°C) **Plan:** (1) Get A → (2) Find D by multiplying A and B → (3) Compute T by doubling D → (4) Calculate T = A – D
MedQA	**Demographics:** Two space explorers, planets named Sarahn and Timon **Timeline:** Current large geographic shift on Sarahn, stable period on Timon with no specified duration **Key events:** Continental break-apart and movement on Sarahn **Imaging/labs:** Continental break-apart and movement on Sarahn **Pathophysiology:** Geologic processes such as plate tectonics and volcanic activity, potentially leading to increased heat and volcanic crust temperature on Sarahn compared to Timon
FinQA	A private space exploration firm, Stellar Frontiers Inc., has discovered two previously unknown planets in a distant galaxy, naming them Sarahn and Timon after their lead explorers. Initial scans reveal stark geological differences between the two planets, presenting unique investment opportunities and risks for resource extraction ventures. - **Sarahn** is undergoing significant tectonic activity, with continents fracturing and shifting rapidly. This instability suggests high volcanic activity, making the planet a prime candidate for mining rare geothermal energy resources or extracting valuable minerals formed under extreme heat and pressure. However, the volatile crust poses high operational risks, requiring advanced (and costly) drilling technology and insurance against geological disruptions. - **Timon** in contrast, exhibits remarkable geological stability, with no continental movement detected. This makes it an attractive site for long-term infrastructure investments, such as establishing permanent research bases or low-risk mining operations for stable mineral deposits. The lack of tectonic activity implies lower energy yields from geothermal sources but offers predictable costs and safer exploration conditions. **Market implications:** - Sarahn: Short-term, high-reward ventures tied to volatile resources, requiring risk management strategies. - Timon: Appeals to conservative investors seeking steady returns. **Corporate decisions:** Stellar Frontiers Inc. must decide whether to license exploration rights to third parties (favoring Timon's stability) or vertically integrate operations to exploit Sarahn's high-risk, high-reward potential. Regulatory bodies may also intervene, imposing environmental safeguards for Sarahn's fragile ecosystem. This scenario sets the stage for analyzing risk-adjusted returns, capital allocation strategies, and sector-specific market reactions.
CDSL	**S –** "Sarahn: hot; Timon: cold" **Goal –** Swap temperature descriptors. **Ops –** `split("; ")`, `reverse_tokens()`, `join(", ")` **Chain –** (1) split → (2) reverse tokens → (3) join **T –** "Timon: cold, Sarahn: hot"

Table 10: **Example of projection output.** Domain-specific synthetic data generated by projecting Synth-Guide Blocks onto scarce domain source data using our proposed PANGEA framework.

GSM8K	Sarahn and Timon discovered two planets, where Sarahn's crust temperature is 48 °C, and Timon's crust is half as hot due to lower geographic activity. If the temperature difference between their planets is twice Timon's crust temperature, what is the difference in their crust temperatures?
MedQA	A 35-year-old space explorer on planet Sarahn presents with a 2-week history of worsening respiratory symptoms, including cough and shortness of breath, following a recent continental break-apart event. The patient's oxygen saturation is 88% on room air, and a chest X-ray shows bilateral infiltrates. Which of the following is the most likely pathophysiologic mechanism contributing to the patient's symptoms? Choose one of the following: A. Increased volcanic crust temperature leading to sulfur dioxide gas exposure B. Decreased atmospheric pressure due to planetary geographic shift C. Inhalation of particulate matter from tectonic activity D. Hypobaric hypoxia resulting from high-altitude terrain formation.
FinQA	**Context:** Stellar Frontiers Inc., a pioneering private space exploration firm, has recently announced the discovery of two exoplanets, Sarahn and Timon, located in a distant galaxy. Initial geological assessments have revealed divergent characteristics that present distinct investment opportunities and operational challenges for resource extraction ventures. **Sarahn**, named after lead explorer Dr. Elena Sarahn, exhibits intense tectonic activity, with continental plates undergoing rapid fracturing and displacement. This geological volatility suggests the presence of rare minerals formed under extreme heat and pressure, such as superconductors and platinum-group metals, which are highly sought after in advanced manufacturing. However, the instability of the planet's crust necessitates the deployment of specialized drilling equipment, which carries significant capital expenditure (CapEx) and operational risk, including potential project delays due to seismic events. These risks are further compounded by the need for comprehensive insurance coverage against geological disruptions, as outlined in Note 3 to the financial statements. **Timon**, named after geologist Marcus Timon, displays remarkable geological stability, with no detectable tectonic movement. This environment is conducive to long-term infrastructure investments, such as modular research facilities or low-risk mining operations for stable mineral deposits like iron and silicon. The predictability of Timon's conditions reduces both CapEx and operational expenditure (OpEx), making it an attractive proposition for conservative investors, including pension funds and aerospace conglomerates seeking sustainable returns. The table below summarizes the projected financial implications of resource extraction ventures on Sarahn and Timon over a five-year period, adjusted for inflation and risk premiums (see Note 4 for detailed risk-adjustment methodology). All figures are presented in $ billions, net of tax and exploration licensing fees. 2024 2025 2026 2027 2028 *Sarahn Revenue* $1.2 $2.8 $4.5 $6.1 $7.9 *Sarahn OpEx* $1.5 $2.3 $3.0 $3.8 $4.6 *Timon Revenue* $0.8 $1.6 $2.4 $3.2 $4.0 *Timon OpEx* $0.6 $1.0 $1.4 $1.8 $2.2 The financial projections underscore the divergent risk-return profiles of the two exoplanets. Sarahn's revenue growth trajectory is steep, reflecting the high-value nature of its volatile resources, but this is offset by escalating operational costs tied to risk mitigation and technological adaptation. Timon, while offering slower revenue growth, demonstrates superior cost efficiency due to its stable environment, resulting in more predictable margins. Notably, Sarahn's OpEx as a percentage of revenue remains persistently high, indicating the ongoing financial burden of geological risk management. In contrast, Timon's OpEx ratio declines steadily, highlighting its suitability for long-term, low-volatility investment strategies. Regulatory considerations, particularly environmental safeguards for Sarahn's fragile ecosystem (as discussed in Note 5), may further influence capital allocation decisions. **Question:** What is the cumulative net profit (revenue minus OpEx) for Stellar Frontiers Inc. across both exoplanets from 2024 to 2026, and how does this compare to the net profit in 2027 alone?
CDSL	Given "Saran: hot; Timon: cold", reverse the order of the entries, swap the separators to commas, and keep the temperature descriptors unchanged. Desired Output: "Timon: cold, Saran: hot"

B.3 o1 score & influence Score

o1 score. To quantitatively assess the quality of synthetic data generated by each augmentation method, we conducted an extensive evaluation using OpenAI's `o1` model. We randomly sampled 5,000 synthetic examples from each method (Naive, Evol-Instruct, and PANGEA) and evaluated each example based on five key criteria: *difficulty*, *formatting*, *problem relevance*, *language quality*, and *clarity*. Each criterion was scored from 0 to 20 points, yielding an overall quality score between 0 and 100 per sample. The detailed evaluation rubric tailored specifically for the GSM8K benchmark is presented in Fig. 24.

Representative examples from each augmentation method, categorized into three distinct quality levels—High (95–100), Medium (90–95), and Low (0–90)—along with their respective o1 evaluation scores, are shown in Tables 11, 12, and 13. These examples qualitatively illustrate the variations in synthetic data quality across methods, supporting the quantitative results summarized in Fig. 4.

```
### GSM8K Grading Rubric

**Candidate Item**
"""
{questions}
"""
---
*(Five components, equally weighted; 0-20 pts each)*

1.  **Problem-Solving Potential**
- Does prompt give every fact required to compute
single, unique answer?
- Prompt must not include worked solution steps.

2.  **Presentation Clarity**
- Is wording clear, unambiguous to average reader?
- Sections/tables optional; ease of understanding key.
- Prompt must *not* contain answer or derivation.

3.  **Mathematical Soundness**
- Are numbers, relationships, conditions consistent
and error-free?
- Single-step or multi-step reasoning acceptable if
correct.

4.  **Conciseness & Readability**
- Is wording free of unnecessary jargon/filler?
- Minor formatting quirks not penalized if comprehension
intact.

5.  **Determinacy & Appropriateness**
- Does problem guarantee exactly one well-defined
answer?
- Is difficulty suitable for GSM8K (basic arithmetic,
percentages, ratios, etc.)?

**Scoring:** Sum of five component scores → total 0-100.
```

Figure 24: **Evaluation rubric for synthetic data quality (GSM8K).** This rubric, utilized by OpenAI's `o1` model, assesses each synthetic data sample across five equally-weighted criteria, with each criterion scored from 0 to 20 points, resulting in a total quality score from 0 to 100.

Table 11: **Naive Method Examples.** Representative examples of synthetic data generated by the Naive method, grouped by o1 evaluation scores (High: 95–100, Medium: 90–95, Low: 0–90).

Result	Generated Synthetic Data
Low	If Dan Deacon was 30 years old when he released his third album in 2009, how old was he when he was born in Baltimore, Maryland?
Midium	A freight train rolls 120 mi on flat prairie and then 80 mi through hills. On flat track it normally burns 2 gal of diesel per mile. The trip's fuel gauge shows 300 gal used in total (this already covers the tiny fuel spent during brief station stops—ignore that). About how many gallons per mile does the locomotive use while on the hilly stretch?
High	At the start of a 30-day month, Li Beibei receives 60 yuan in pocket money and, spending exactly 1.5 yuan on lunch each day, how much money will she have left by the end of the month?

Table 12: **Evol-Instruct Examples.** Representative examples of synthetic data generated by the Evol-Instruct method, grouped by o1 evaluation scores (High: 95–100, Medium: 90–95, Low: 0–90).

Result	Generated Synthetic Data
Low	Ethan once sorta-borrowed $2,500-ish from Lucas at approximately 15% interest that may or may not exist. Ethan flung $300 back sometime while Lucas was napping. It's now both exactly and not exactly one year later. How much random cash should Ethan hurl at Lucas so they can declare the debt mysteriously settled?
Midium	Tom is repainting two rooms in his office building: first, a rectangular conference room whose floor measures 20 ft by 30 ft and whose walls rise 12 ft high, and in that room he intends to cover only the two longer walls—each of which contains a single 4 ft × 7 ft doorway that will be left unpainted—and second, a perfectly square meeting room 15 ft on a side with walls of the same height, where he plans to coat all four walls except for three identical 3 ft × 4 ft windows (one window set into each of three different walls while the fourth wall has no openings). Ignoring ceilings, floors, trim, and any other fixtures, how many square feet of wall surface will actually receive paint?
High	Lily's phone plan costs $45 per month. For the first three months she pays a promotional rate of $25 per month. In the 6th month she incurs an international roaming fee of $18. Assuming all other months are billed at the regular rate and taxes are included in the prices, what is her total payment for the first 9 months?

Table 13: **PANGEA Examples.** Representative examples of synthetic data generated by the PANGEA method, grouped by o1 evaluation scores (High: 95–100, Medium: 90–95, Low: 0–90).

Result	Generated Synthetic Data
Low	Carrying the $40 he saved over the summer, a student heads to the campus bookstore and picks out three textbooks—the first carries a price tag of $18, the second rings up at $4 less than that, and the third, sitting on the new-releases shelf, is exactly $2 more than the second; with tax waived for the back-to-school promotion and a free bookmark tossed in, what is the price of the third textbook?
Midium	A runner's GPS watch shows she covered 10 miles in 1 hour 30 minutes; given this, first convert the total time to minutes, then use that time to find her average speed in miles per hour, and finally determine (to the nearest second) her average pace per mile—how many minutes and seconds does each mile take?
High	Alex saved $150 to cover all the costs of getting his driver's license: the state imposes a $50 base license fee and a $30 road-test fee, each of which is increased by a 10% sales tax that is applied once to the respective fee, and he must also pay $12 for the required ID photo, which is not taxed. After paying every fee and the applicable tax, how many dollars remain from Alex's $150 budget?

The examples provided in Tables 11, 12, and 13 clearly illustrate qualitative differences in the synthetic data generated by each method. While the Naive method often generates overly simplistic or poorly structured questions, Evol-Instruct tends to create scenarios that, although slightly improved, still frequently include irrelevant details or overly complex phrasing. In contrast, PANGEA consistently produces clear, contextually coherent, and precisely structured examples. Specifically, PANGEA-generated problems demonstrate more realistic scenarios, appropriate complexity levels, and more accurate and concise wording, resulting in higher overall evaluation scores by OpenAI's o1 model. This qualitative distinction highlights the effectiveness of PANGEA's structured, three-stage approach in generating synthetic data that is both domain-relevant and diverse, addressing key limitations observed in other methods.

Influence score. We measure the usefulness of the synthetic training examples generated by our PANGEA framework via influence scores estimated with the **DataInf** method of Kwon et al. [16]. We denote the domain-specific source dataset by $\mathcal{D}_{\text{s}}$ and use it exclusively as the validation set when evaluating how each synthetic example affects model performance. After training our baseline model, we perform an additional round of low-rank adaptation (LoRA) fine-tuning on the union of the synthetic set $\mathcal{D}_{\text{syn}}$ and $\mathcal{D}_{\text{s}}$, and the resulting parameters are used only to compute gradients for the influence-score estimator.

Classical influence-function approaches require the inverse of the Hessian $H(\theta)$, whose exact computation or iterative approximation is prohibitive for modern LLMs, while DataInf circumvents this bottleneck by approximating the Hessian inverse in closed form using first-order gradients and a Sherman–Morrison correction, reducing the cost to $\mathcal{O}\big(\sum_l n\, d_l\big)$ time and $\mathcal{O}\big(\max_l d_l\big)$ memory while retaining high correlation with exact scores.

Let the synthetic training set be $\mathcal{D} = \mathcal{D}_{\text{syn}} = \{(x_i, y_i)\}_{i=1}^{n}$, the validation set $\mathcal{D}^{\text{val}} = \mathcal{D}_{\text{s}} = \{(x_j^{\text{val}}, y_j^{\text{val}})\}_{j=1}^{m}$, and f_θ the LoRA-adapted model; the influence of a synthetic example (x_k, y_k) on validation performance is

$$I_{\text{DataInf}}(x_k, y_k) = \sum_{l=1}^{L} \frac{1}{\lambda_l} \left[\frac{1}{n} \sum_{i=1}^{n} \frac{(v_l^\top g_{l,i})\,(g_{l,i}^\top g_{l,k})}{\lambda_l + g_{l,i}^\top g_{l,i}} - v_l^\top g_{l,k} \right], \tag{4}$$

where $g_{l,i} = \nabla_{\theta_l} \ell\big(y_i, f_\theta(x_i)\big)$ and $v_l = \frac{1}{m} \sum_{j=1}^{m} \nabla_{\theta_l} \ell\big(y_j^{\text{val}}, f_\theta(x_j^{\text{val}})\big)$, and where the damping term is $\lambda_l = 0.1\, \frac{1}{n d_l} \sum_{i=1}^{n} g_{l,i}^\top g_{l,i}$ with d_l the parameter dimension of layer l.

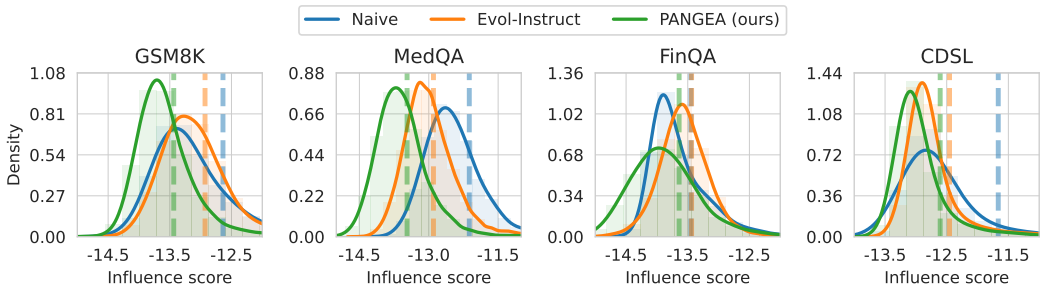

Figure 25: **Influence scores for data quality analysis.** Each histogram shows the distribution of data points for each augmentation method over influence score ranges. A kernel density estimation curve is overlaid as a solid line for better visualization, and the vertical dashed line denotes the mean.

As shown in [16], *the sign of $I_{DataInf}$ matches the direction of the validation-loss change*; hence $I_{\text{DataInf}}(x_k, y_k) < 0$ implies that the sample *lowers* the validation loss and is therefore beneficial. Large-magnitude negative scores thus pinpoint the most influential, high-quality data.

Figure 25 plots the signed log-scaled distribution of the estimated scores, showing that PANGEA-generated synthetic examples yield larger negative influence values than all baselines and therefore lower the validation loss on $\mathcal{D}_{\text{s}}$ more effectively.

B.4 Embedding visualization with Cosine similarity

Embedding visualization. To qualitatively analyze the diversity of synthetic data generated by our framework, we visualized embeddings of 30k synthetic data points along with original source data across all benchmarks using the `all-mpnet-base-v2` model and t-SNE projection (Fig. 26), clearly illustrating improved diversity and coverage.

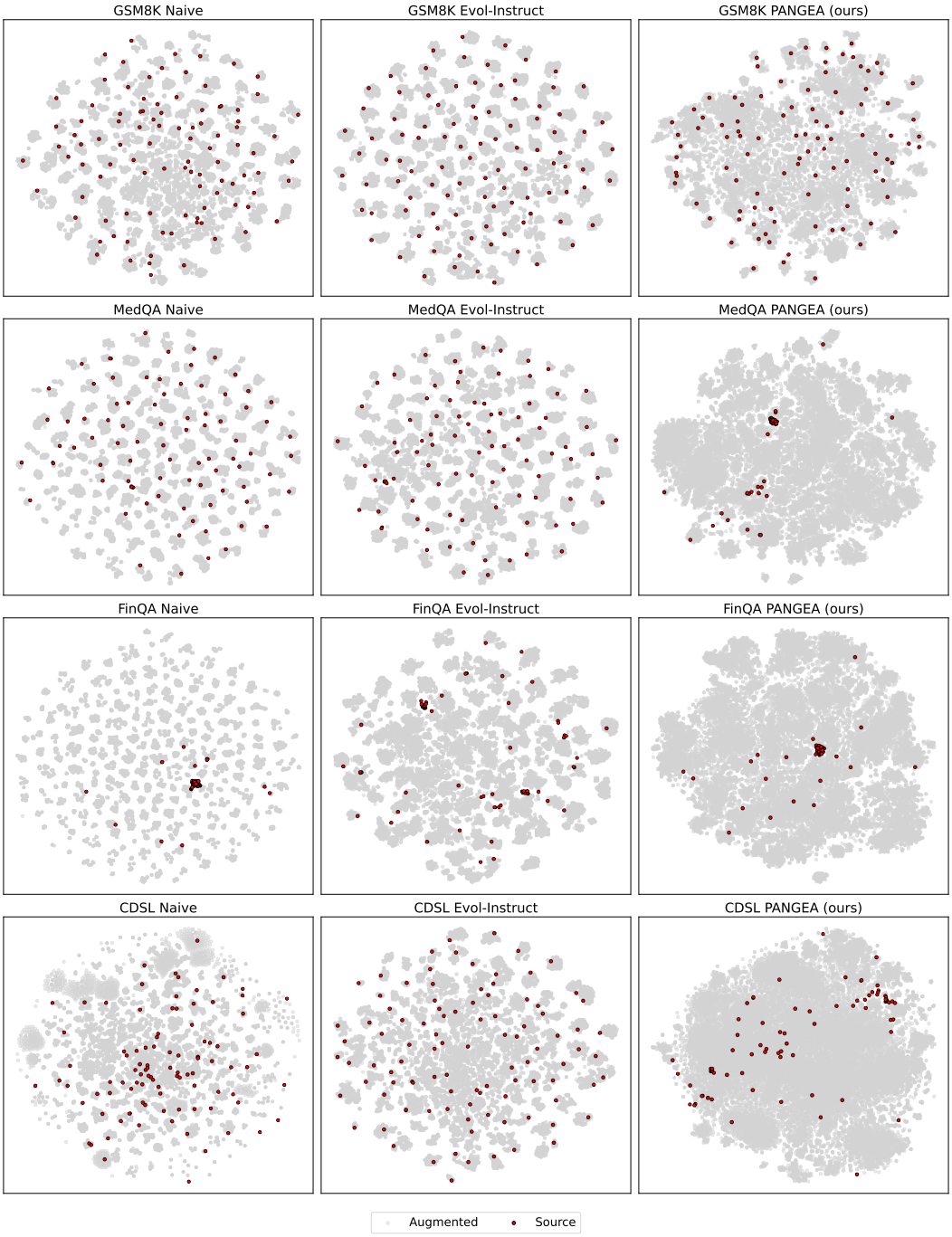

Figure 26: **Embedding Visualization of Synthetic Data Diversity.** t-SNE visualization of embeddings for 30k synthetic data points generated by Naive, Evol-Instruct, and PANGEA methods, along with the original domain-specific source data.

Table 14: **Cosine similarity for measuring synthetic data diversity.** We compute pairwise cosine similarities between synthetic samples generated from the same source instance.

Method	GSM8K (↓)	MedQA (↓)	FinQA (↓)	CDSL (↓)
Naive	0.816±0.004	0.872±0.003	0.906±0.003	0.838±0.003
Evol-Instruct	0.776±0.003	0.847±0.002	0.742±0.002	0.767±0.008
PANGEA	**0.310**±0.000	**0.600**±0.004	**0.671**±0.001	**0.377**±0.001

Cosine similarity. Furthermore, to quantitatively assess the diversity of synthetic data, we computed average pairwise cosine similarities between synthetic samples generated from each of the 100 source instances, with each source instance producing 100 synthetic samples. Table 14 summarizes these results, where lower cosine similarity values indicate higher intra-source diversity. PANGEA consistently achieves substantially lower similarities across all benchmarks compared to baseline methods, clearly demonstrating its superior capability to generate numerically diverse and distinctive synthetic samples.

B.5 DataTune failed example

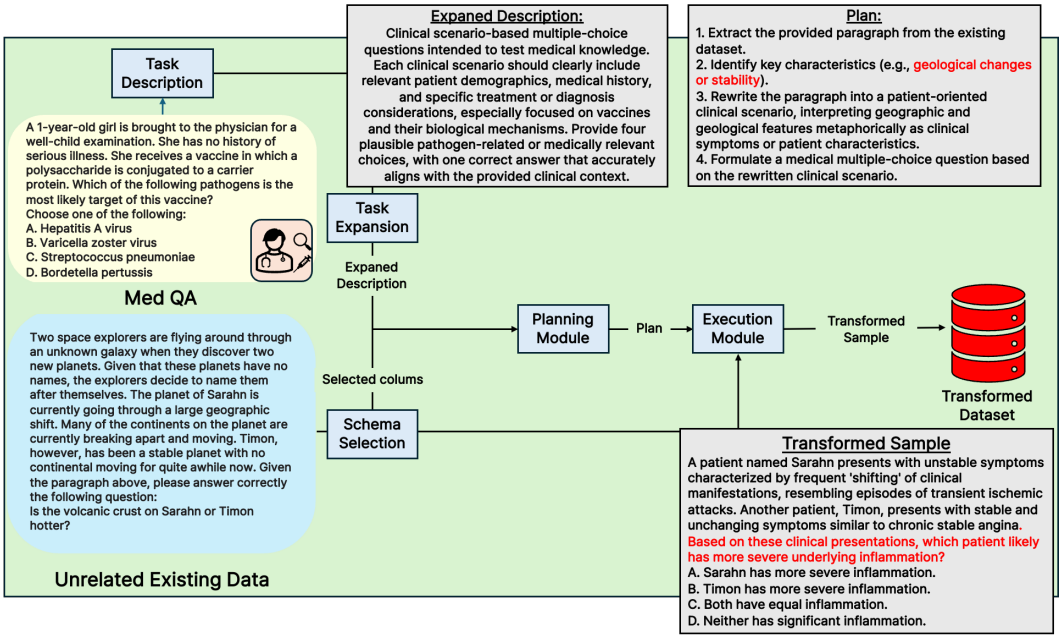

Figure 27: **Failed synthetic data examples generated by DataTune.** These examples illustrate typical issues encountered when DataTune was applied using unrelated general-domain data in our severely data-scarce setting, highlighting poor domain alignment and low-quality outcomes.

To assess the performance of DataTune[9], a baseline method that augments synthetic data using related existing datasets, we generated synthetic data through five module expansions as illustrated in Figure 27. While DataTune typically leverages related but possibly misaligned data, our experimental setup involves an extreme scenario in which domain-specific source data ($\mathcal{D}_s$) is scarce, and no related existing general-domain data ($\mathcal{D}_g$) is available. Consequently, we had to use unrelated data for augmentation, making the application of DataTune challenging and often resulting in low-quality synthetic examples, as illustrated in Fig. 27.

Empirically, when we evaluated DataTune-generated data on the MedQA benchmark at the 10k scale, the resulting model achieved an accuracy of only 34.95%, performing even worse than the Naive method, which solely relies on LLM-generated data without explicit alignment. This clearly demonstrates the limitations of employing DataTune under conditions of severe data scarcity and domain mismatch.

B.6 Human evaluation

Table 15: **Human evaluation.** Fifty-seven annotators ranked three generators (A: Naive, B: Evol-Instruct, C: PANGEA) as *Best/Second/Worst* for 100 randomly sampled items per domain. Values are preference proportions (%). Bold indicates the maximum within each Best/Second/Worst block. Final columns report p-values from χ^2 and Friedman tests.

	Best (%)			Second (%)			Worst (%)			Significance (p)	
Domain	A	B	C	A	B	C	A	B	C	χ^2	Friedman
GSM8K	24.22	28.96	**46.82**	19.57	**46.62**	33.81	**56.21**	24.42	19.37	4.0×10^{-5}	5.96×10^{-3}
FinQA	20.34	26.22	**53.44**	17.24	**46.49**	36.27	**62.42**	27.29	10.29	2.2×10^{-8}	1.64×10^{-5}
MedQA	24.46	24.25	**51.29**	24.29	**37.87**	37.84	**51.25**	37.88	10.87	6.3×10^{-5}	7.65×10^{-4}
CDSL	28.93	20.41	**50.66**	**34.15**	31.72	34.13	36.92	**47.87**	15.21	1.9×10^{-3}	2.19×10^{-3}

We randomly sampled 100 synthetic items from each benchmark (GSM8K, FinQA, MedQA, CDSL). Fifty-seven graduate-level annotators (after a brief pilot) evaluated, in random order, three candidates per item—**Naive (A)**, **Evol-Instruct (B)**, and **PANGEA (C)**—using the same rubric as in the supplement, and ranked them *Best/Second/Worst*. Across all domains, PANGEA (C) received the largest share of *Best* preferences, while Naive (A) was most frequently *Worst* and Evol-Instruct (B) typically ranked in between.

To assess significance, we applied a χ^2 test of independence on the 3×3 contingency table of counts and a non-parametric Friedman test on per-item ranks; all domains showed $p < 0.01$. Inter-annotator agreement was moderate (Krippendorff's $\alpha = 0.61$). Percentages are computed over item-level tallies per domain (one Best/Second/Worst per item). Rubric details and example judgments are provided in the supplement.

B.7 Domain-agnostic prompting and multilingual transfer

Table 16: **Domain-agnostic prompting Result.** We provide unified templates for Synth-Profiling, Prompt-Writer, and Projection to remove domain-specific cues while preserving structure. Empirically, Table 4 shows PANGEA(agnostic) still outperforms baselines, validating scalability when expert prompt engineering is unavailable.

Method	GSM8K	MedQA	FinQA	CDSL
Pre-trained	5.69	28.91	6.02	0.00
Instruction-tuned	45.03	37.31	26.68	0.57
Naive	26.91	35.42	24.06	3.20
Evol-Instruct	27.36	36.29	26.68	5.22
PANGEA (agnostic)	31.70	36.39	34.74	9.25
PANGEA (original)	**32.52**	**37.78**	**36.44**	**11.30**

We remove domain-specific cues while preserving structure by using unified templates for Synth-Profiling, Prompt-Writer, and Projection; the same slot layout is instantiated across tasks and the learned profile η supplies domain signals. With 10k synthetic samples per task and otherwise identical training, **PANGEA (agnostic)** still outperforms non-projection baselines with only a small gap to domain-tailored prompts (Table 16).

To test cross-lingual robustness, we project English D_g into Korean and synthesize 10k CSAT items using official seeds, evaluate on a held-out 2025 set, and observe that **PANGEA** attains the highest total score and is the only method that solves *Hard* items (Table 17). In practice, keeping the Prompt-Writer in English for D_g parsing and enforcing Korean only at Projection reduced literal-translation artifacts. When the seed pool is extremely small (e.g., 10), η becomes under-specified (fewer equations, weaker tone/difficulty constraints), mildly increasing verbosity or scenario realism issues—consistent with the seed-size trend in § B.8.

Table 17: **Korean CSAT Result.** We synthesize 10k Korean CSAT items using English D_g and evaluate on the held-out 2025 set. Results in Table 5 show that PANGEA yields the highest total score and uniquely solves *Hard* items, indicating cross-lingual projection is feasible when the generator LLM is multilingual.

Method	Easy	Normal	Hard	Total
Pre-trained	40.00	13.64	0.00	13
Instruction-tuned	60.00	31.82	0.00	27
Naive	60.00	13.64	0.00	15
Evol-Instruct	40.00	31.82	0.00	22
PANGEA	**60.00**	**36.36**	**5.26**	**34**

B.8 Seed-size ablation

We vary the number of seeds from 100 to 10 while fixing the synthetic set at 10k; for each level we re-derive η and sample from the same D_g working subset to isolate seed effects. Results in Table 18 show robustness down to 20–40 seeds and a drop at 10, where η under-specifies style and difficulty. Qualitatively, 10-seed profiles request fewer steps and looser unit constraints than 100-seed profiles; in practice, $\sim$40 well-chosen seeds suffice to unlock most gains.

Table 18: **Seed-size ablation.** Seeds vary 100→10 with the synthetic set fixed at 10k. PANGEA is robust to 20–40 seeds and degrades only at 10 due to an under-specified profile.

Method (#seed)	GSM8K	MedQA	FinQA	CDSL	Avg.
Naive (100)	26.91	35.42	24.06	3.20	22.40
Evol-Instruct (100)	27.36	36.29	26.68	5.22	23.89
PANGEA (100)	**32.52**	**37.78**	**36.44**	**11.30**	**29.51**
PANGEA (80)	32.91	37.34	36.37	11.01	29.41
PANGEA (40)	31.84	36.10	35.21	10.15	28.33
PANGEA (20)	31.21	35.79	34.83	8.70	27.63
PANGEA (10)	29.28	33.27	28.31	3.77	23.66

B.9 Data filtering and verifier analysis

We explored two strategies beyond projection. *(i) Verifier for D_g pre-filtering:* a linear probe on `Llama-3.1-8B` embeddings trained on $\sim$20k (D_s, D_g) pairs labeled by our automatic judge to predict whether a pair would yield high-quality outputs. Despite balanced labels, positive precision hovered at 50–60%, yielding too few usable pairs and requiring domain-specific retuning. *(ii) Quality-based filtering of synthetic data:* on GSM8K, training only on low-scored vs. high-scored subsets underperformed training on the full 30k set (Table 19). These results suggest that retaining diverse "imperfect" samples is beneficial when generation is anchored by η and τ, whereas hard pre-filtering incurs a precision–yield trade-off.

Table 19: **Quality-based filtering on GSM8K.** Using only high- or low-scored subsets underperforms the full 30k set. This indicates that diversity in synthetic data, anchored by η and τ, is more valuable than strict filtering.

Training data	Acc.
10k Low-quality only	28.66
20k High-quality only	35.47
30k Whole set	**37.72**
Baselines	
Naive	26.91
Evol-Instruct	27.36

B.10 Broader impact

Our proposed synthetic data generation framework can positively impact society by effectively addressing data scarcity in specialized domains, enabling improved performance and broader applicability of AI systems. However, we acknowledge potential negative impacts, such as the risk of generating synthetic data that unintentionally reflects biases or misinformation from general-domain sources. We therefore emphasize responsible data use, careful evaluation, and continuous monitoring of synthetic outputs to mitigate these risks.

C Miscellaneous

C.1 Related works

Synthetic Data Generation Methods. Recent studies propose various techniques to effectively augment datasets for domain-specific tasks and LLM fine-tuning. Approaches such as WizardLM [36], WizardMath [20], and WizardCoder [21] leverage prompt-based reinforcement learning and instruction-tuning to improve LLMs' capability to follow intricate instructions, including mathematical reasoning, and code generation. However, these methods primarily rely on the generative capabilities of LLMs alone, often leading to the production of off-domain or lower-quality synthetic samples.

TinyStories [8] demonstrated the training efficiency of smaller models using synthetic narratives generated by large-scale LLMs such as GPT-3.5 and GPT-4. Despite this effectiveness, its generated data lacks structural variety, restricting scalability and limiting applicability in specialized or complex scenarios.

Magpie [38] proposes an innovative method to synthesize alignment data exclusively through LLM prompting, without the need for existing aligned datasets. While advantageous for iterative and self-augmenting data improvements, this approach predominantly suits general use-cases, showing limited effectiveness in highly specialized domains. Additionally, it inherently struggles to generate data involving concepts or scenarios absent from the LLM's original training corpus.

Retrieval-augmented approaches like CRAFT [49] and DataTune [9, 27] utilize external knowledge bases or related datasets to enhance data quality. CRAFT retrieves and refines relevant text segments via instruction-tuned LLMs, while DataTune [9] identifies closely related public datasets and augments data through structured modules such as Task Description, Task Expansion, Schema Selection, Planning, and Execution. However, both face substantial limitations when domain-relevant datasets or resources are insufficient or unavailable.

ELTEX [23] addresses cybersecurity data augmentation by iteratively extracting explicit domain indicators from real-world raw data. Nonetheless, this method fundamentally relies on the presence of initial domain-specific datasets, constraining its effectiveness when such initial domain signals are absent.

In task-specific settings, researchers have begun tailoring the entire data-synthesis pipeline to the unique structure of a single downstream task rather than a topical domain. For example, DISCO [50] generates phrasal perturbations with GPT-3, retaining only those pairs verified by a strong NLI teacher, significantly enhancing robustness and out-of-distribution (OOD) generalization.

Similarly, CORE [51] employs a learned retriever coupled with GPT-3-driven minimal editing, extracting label-flipping excerpts from a large unlabeled corpus to improve OOD performance substantially (up to 4.5 pp on NLI and 6.2 pp on cross-domain sentiment tasks), modifying only a small fraction of the training data. However, these task-specific approaches inherently require either strong, domain-aligned teacher models or extensive unlabeled in-domain corpora, constraining their applicability in highly specialized or extremely data-scarce settings.

C.2 Safeguards

In this work, we introduced a synthetic data generation framework designed to effectively leverage general-domain data for generating diverse and high-quality domain-specific datasets. Throughout the data generation and model training processes, we ensured that our methodology does not create or utilize datasets containing sensitive, personal, or otherwise ethically problematic information. Instead, we exclusively used publicly available, openly accessible datasets, thereby circumventing privacy concerns and ethical risks.

Beyond privacy, we recognize that synthetic data at scale introduces several other potential risks:

(i) *Bias and representativeness.* Projecting information from general-domain D_g and a generator LLM can propagate or amplify societal stereotypes, and profile mis-specification may cause semantic drift that disadvantages certain groups or topics.

(ii) *Misinformation and factuality.* Plausible but incorrect content can be produced, which is unacceptable in safety-critical domains (e.g., medicine, finance) without expert review.

(iii) *Benchmark contamination and data pollution.* Unlabeled synthetic data can leak into public corpora or benchmarks, inflating scores and creating feedback loops if re-used for training.

(iv) *Transparency and provenance.* Mixing synthetic with real data without disclosure undermines downstream auditing and accountability.

(v) *Intellectual property and privacy.* While our pipeline avoids private domain data by design, the generator might regurgitate licensed text or rare memorized strings; licenses and de-duplication audits remain necessary.

(vi) *Dual-use and misuse.* Domain-style generation can be misapplied for spam or to produce harmful content.

(vii) *Environmental impact.* Large-scale generation and judging consume compute and energy.

To address these risks, we have implemented several mitigations and safeguards. As our primary technical mitigation for bias (risk i), we anchor generation to structured profiles (*Synth-Profiling* $\rightarrow$ Synth-Guide Block τ) to reduce semantic drift. The effectiveness of this approach was confirmed through experimental validation using automatic quality checks and human preference assessments (§ B.6). We also recommend optional manual audits for high-stakes domains to address factuality (risk ii).

To manage contamination, transparency, and IP (risks iii, iv, v), we enforce license compliance for all sources and perform de-duplication against seed/benchmark sets.

Furthermore, recognizing the potential for misuse (risk vi), we have implemented a responsible-release policy. This includes:

(a) reporting all prompts and data flows;

(b) releasing a *sanitized* subset with controlled access mechanisms, detailed documentation, and clear usage guidelines;

(c) discouraging the redistribution of unlabeled synthetic data; and

(d) clearly tagging provenance when releasing any assets.

This documentation is aimed at ensuring that users understand both the capabilities and limitations of our synthetic data and associated models. Our guidelines particularly emphasize responsible and ethical use, aiming to minimize misuse risks and maximize positive societal impacts.

Finally, to mitigate environmental impact (risk vii), we document compute (GPU type, steps, sequence length. cf. § B.2). As our methodology relies on model inference for data generation, we recommend carbon-aware scheduling, token-efficient profiling, and reuse of D_g partitions. Compared to training from scratch, the dominant cost is LLM inference for projection/judging, which can be amortized by reusing η across scales and domains.