AIDE: Attribute-Guided MultI-Hop Data Expansion for Data Scarcity in Task-Specific Fine-tuning

Jiayu Li[1,3,*], **Xuan Zhu**[2], **Fang Liu**[2], **Yanjun Qi**[2,3]

[1]Syracuse University, [2]AWS Bedrock Science

[3]**Correspondence:** jli221@data.syr.edu, yanjunqi@amazon.com

Abstract

Fine-tuning large language models (LLMs) for specific tasks requires diverse, high-quality training data. However, obtaining sufficient relevant data remains a significant challenge. Existing data synthesis methods either depend on extensive seed datasets or struggle to balance task relevance and data diversity. To address these challenges, we propose **A**ttribute-guided mult**I**-hop **D**ata **E**xpansion (AIDE), a novel data synthesis framework that uses a multi-hop process to expand very few seed data points while ensuring data diversity and task relevance. AIDE extracts the main topic and key knowledge attributes from the seeds to guide the synthesis steps. The process repeats for K hops, using the generated data as seeds. To prevent irrelevant data generation as the hop depth increases, AIDE incorporates a residual connection mechanism. Our empirical results show that AIDE enables fine-tuning of Mistral-7B, Llama-3.1-8B and Llama-3.2-3B from 10 seeds, surpassing the models fine-tuned on human curated data. Furthermore, AIDE outperforms state-of-the-art data synthesis methods, such as Evol-Instruct, by over 30% in task-specific fine-tuning. Code is available at `https://github.com/Code4Graph/AIDE`.

1 Introduction

Fine-tuning with task-specific training data is essential because it allows a pre-trained model to adapt and optimize for a specific task, resulting in better performance in that domain. However, task-specific data is insufficient or unavailable for many use cases, and manually curating the data is labor intensive (Gandhi et al., 2024).

To overcome the limitation, an approach from (Wei et al., 2022; Xu et al., 2022) samples task-specific training data from public NLP datasets, but the sampling often covers limited information. Another category of recent methods leverages the capabilities of LLMs to automatically generate large-scale synthetic data, enabling the training of advanced models in specific task domains. For example, Prompt2Model (Viswanathan et al., 2023) and DataTune (Gandhi et al., 2024) rely on several candidate datasets to synthesize task-specific data for fine-tuning LLMs. However, these methods either require a large set of seed data for rewriting or produce synthetic data that lacks task relevance and diversity, as they do not maintain sufficient control over the synthesis process.

To address these challenges, we propose **AIDE** (**A**ttribute-guided mult**I**-hop **D**ata **E**xpansion), a novel data synthesis framework that generates abundant training data from a small set of seed inputs, as shown in Figure 1. Our framework focuses on maintaining high task relevance, diversity, and quality in the synthetic data for specific tasks. AIDE uses LLMs as key players via a multi-hop synthesis process. Each hop in AIDE begins by extracting the main topic and important knowledge attributes from a seed sample using a LLM. This builds knowledge triplets, and AIDE traverses these triplets (each consisting of a topic, relationship, and attribute) to synthesize new data points. In the next hop, each newly generated data point becomes a seed, and the process repeats until reaching a depth of K hops. This multi-hop mechanism allows for recursive data synthesis along all paths of a process tree, enabling the generation of large-volume data from just a few seeds. Extracted attributes act as control nodes in the multi-hop tree, ensuring the generated data points remain relevant to the target task. We also introduce personas as new key attributes, enhancing the generation of diverse data. As the depth of the recursive synthesis increases, the relevance of the synthetic data may diminish. To address this, we propose a residual connection mechanism to reduce irrelevance.

To validate AIDE, we conduct experiments with three pretrained models (Mistral-7B, Llama-3.1-

[*]Work was done during an internship at AWS.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 1083–1101

8B, and Llama-3.2-3B). We evaluate the performance of these models when fine-tuned with synthetic data generated by AIDE, comparing the results against models fine-tuned with human-curated (gold) data and synthetic data from state-of-the-art (SOTA) methods. Our evaluations span a range of tasks from well-known benchmarks, including industrial datasets like MedQA (Jin et al., 2020) and FinBen (Xie et al., 2024), as well as BIG-Bench (bench authors, 2023), MMLU (Hendrycks et al., 2021), ARC-Challenge (Clark et al., 2018), GSM8K (Cobbe et al., 2021), and TruthfulQA (Lin et al., 2022). For comparison, we include SOTA data synthesis methods such as Evol-Instruct (Xu et al., 2024), DataTune (Gandhi et al., 2024), and Prompt2Model (Viswanathan et al., 2023). Our main contributions are as follows:

- We introduce AIDE, a novel data synthesis framework that has a multi-hop synthesis, guided by attributes and personas, to generate abundant, task-relevant, diverse, and high-quality data from only a few of seed inputs.
- We design a residual connection mechanism to mitigate the irrelevance as the depth of hop increases during the multi-hop synthesis.
- In zero-shot prompting, Mistral-7B fine-tuned with synthetic data from AIDE achieves average relative improvements of over 6% and 30% across tasks, compared to Mistral-7B fine-tuned with gold training data and SOTA data synthesis methods. Additionally, AIDE enhances the performance of Llama-3.1-8B and Llama-3.2-3B, yielding average relative improvements of approximately 0.7% and 1.5% across tasks, respectively, compared to fine-tuning with gold data.

2 Related Work

Data synthesis for fine-tuning LLMs targets two primary problems. The first is open-domain generation, which synthesizes data across a wide range of topics and complexity levels. The second is task-specific generation, where synthetic data is tailored to a particular task. One can use the synthetic data in fine-tuning LLMs through techniques, such as instruction tuning, preference tuning, and their variations. This paper focuses on synthesizing training data for instruction tuning to enhance the performance of LLMs for specific tasks. We discuss related methods for data synthesis in both open and task-specific domains in Appendix A.

Our approach AIDE differs from related methods as follows: For each data point, AIDE extracts a topic, attributes, and their relationships in the form of knowledge triplets. These triplets then guide the generation of synthetic data relevant to a specific task. AIDE also has a residual connection mechanism to maintain the relevance of synthetic data as synthesis depth increases. Additionally, AIDE introduces personas to expand attributes, and uses a self-reflection technique to improve diversity and quality of the synthetic task-specific data.

3 Proposed Method: Attribute-Guided Multi-Hop Data Expansion (AIDE)

In the section, we discuss the details of AIDE. We define the seed data in a specific task as $D_{\text{seed}} = \{(X_i, Y_i)\}_{i=1}^{n}$ where n is the number of data points in D_{seed}, X_i is the i-th question and Y_i is the corresponding answer to X_i. We aim to automatically synthesize abundant data within the specific domain by expanding D_{seed} into $D = \{(X_i, Y_i)\}_{i=1}^{m}$, where $n \ll m$ and m is the size of synthetic dataset. We use the synthetic dataset to fine-tune a model, improving its performance in the specific domain.

3.1 Multi-Hop Synthesis

To synthesize abundant data, we propose a multi-hop synthesis approach, with an example illustrated in Figure 8 of Appendix B.

Definition 3.1 (Multi-hop synthesis)**.** *Given a seed data point $X_i^{(0)}$ where $1 \le i \le n$, multi-hop synthesis involves recursively generating data from $X_i^{(0)}$ until reaching depth K. At depth K, m_K denotes the number of K-hop neighbors $X^{(K)}$ of $X_i^{(0)}$, where $X^{(K)} = \{X_1^{(K)}, X_2^{(K)}, ..., X_{m_K}^{(K)}\}$. Each $X_i^{(K)}$ for $1 \le i \le m_K$ is a synthetic data point. The total size of synthetic data after multi-hop synthesis is $m = n(m_1 + m_2 + ... + m_K)$, where m_1, m_2 and m_K correspond to the number of synthetic data at the depth 1, 2, K, respectively.*

3.2 Multi-Hop Synthesis Guided by Attributes and Persona

During the multi-hop synthesis, we need to ensure the generated data remains relevant to the seed data within the specific task domain. One approach is to use operations as paths in the multi-hop synthesis to create data by rewriting the previous data. However, manually enumerating all possible paths is

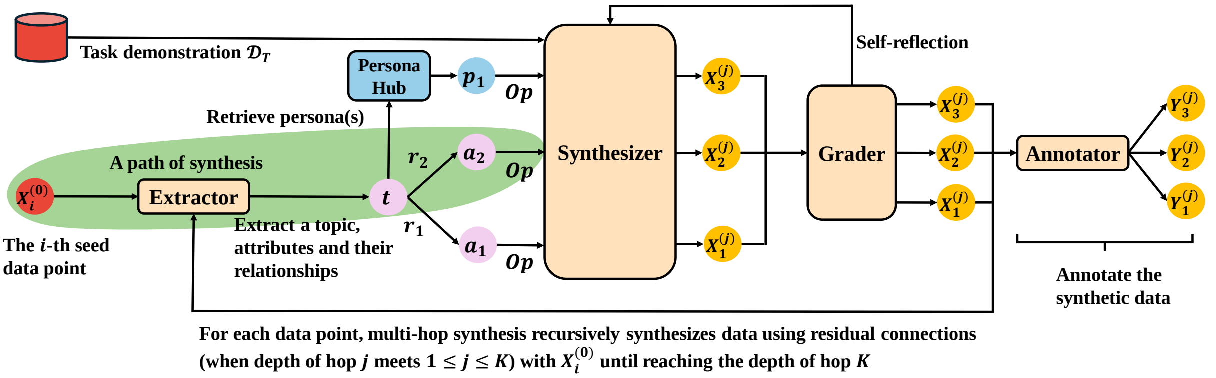


Figure 1: Overview of the workflow of AIDE. $X_i^{(0)}$ denotes the i-th task-related seed data point. AIDE includes four steps. (1) a LLM extractor extracts a topic t, knowledge attributes a_1 and a_2 with relationships r_1 and r_2 of a data point. (2) During the multi-hop synthesis at the depth of hop j, a LLM acts as a synthesizer with task demonstrations $\mathcal{D}_T$ to generate data $X_1^{(j)}$, $X_2^{(j)}$ and $X_3^{(j)}$ along paths of synthesis with a predefined operation Op (i.e., adding constraints). (3) To enhance the diversity of synthesis, we expand attributes by retrieving a persona p_1 from a persona hub with t. Finally, a LLM as an annotator generates the label of synthetic data. We describe the technical details of AIDE in Section 3.

Variables	Content
$X_i^{(0)}$	Generate a list of ten items a person might need for a camping trip.
Task demonstration $\mathcal{D}_T$	What are the packages pepole needs to prepare for a bike ride through parks or countryside?
$< t_1, r_1, a_1 >$	<Outdoor activities, Involves, Camping>
$< t_1, r_2, a_2 >$	<Outdoor activities, Needs, Camping gears>
Persona p_1	An adventurous senior citizen who can recall some related experiences of living in high elevation.
Predefined operation Op	Adding constraint
$X_1^{(1)}$	What are the top essential items recommended by a survival expert for a successful camping trip in harsh weather conditions?
$X_2^{(1)}$	Generate a list of ten essential items required for a multi-day camping expedition, ensuring that the list includes both shelter and food.
$X_3^{(1)}$	Generate a list of ten essential items a person might need for a camping trip, ensuring each item is crucial for outdoor activities and aligns with basic camping gear requirements.

Table 1: The 1-hop synthesis in Figure 9 of Appendix C uses an input data point $X_i^{(0)}$ to generate a representation of the data point $\mathcal{A}_i^{(0)}$ with triplets $< t_1, r_1, a_1 >$ and $< t_1, r_2, a_2 >$. We retrieve the persona P_1 according to t_1. Through the triplets, task demonstrations $\mathcal{D}_T$, the persona p_1 and the predefined operation Op, we synthesize $X_1^{(1)}$, $X_2^{(1)}$ and $X_3^{(1)}$ by combining $X_i^{(0)}$ with its corresponding task category and related examples.

infeasible, limiting the volume of synthetic data. Furthermore, introducing operations without controlling content along the paths can lead to irrelevant data. To address this, we propose a multi-hop synthesis method guided by attributes and personas, introduced in Sections 3.2.1 and 3.2.2, which enhances data diversity while maintaining relevance to the task-related seed data.

3.2.1 Multi-Hop Synthesis Guided by Attributes for Relevance

For a given seed data point, we can extract its main topic, related attributes, and their relationships. Using in-context learning (ICL) (Wen et al., 2024; Melnyk et al., 2022; Jin et al., 2023), a LLM can represent a data point $X_i^{(K)}$ as $\mathcal{A}_i^{(K)} = \{\langle t, r, a\rangle_i^{(K)} | r \in R; t, a \in E\}$, where t, r and a represent the topic, relations and attributes, respectively. R is the set of relations while E contains the topic and attributes. The process of extracting the $\mathcal{A}_i^{(K)}$ for the i-th data $X_i^{(K)}$ is as follows,

$$\mathcal{A}_i^{(K)} = \text{LLM}(X_i^{(K)}). \tag{1}$$

We show the prompt of how to extract $\mathcal{A}_i^{(K)}$ from $X_i^{(K)}$ in Appendix I. Using $X_i^{(K-1)}$ and a triplet $\langle t, r, a\rangle_i^{(K-1)}$ from $\mathcal{A}_i^{(K-1)}$ based on Eq. (1), a LLM synthesizes $X_i^{(K)}$ with task demonstrations $\mathcal{D}_T$. The task demonstrations $\mathcal{D}_T$ includes task-related examples to guide the process of synthesis. To improve data complexity, we apply operations Op (i.e., adding constraints, reasoning, and concreteness) during synthesis to enhance the quality of synthetic data (Xu et al., 2024). This process is summarized as:

$$X_i^{(K)} = \text{LLM}(X_i^{(K-1)}, \langle t, r, a\rangle_i^{(K-1)}, Op, \mathcal{D}_T). \tag{2}$$

Prompts for the synthesis process are shown in Appendix J. A multi-hop synthesis example is demonstrated in Figure 9 in Appendix C and Table 1.

3.2.2 Multi-Hop Synthesis Guided by Personas for Diversity

Song et al. (2024) shows that fine-tuning LLMs with diverse data improves performance. However, generating diverse data at scale by LLMs requires varied prompts (Chan et al., 2024). To address this, we leverage Persona Hub (Chan et al., 2024) to enhance synthetic data diversity. For each data point, we retrieve the top-P personas by using cosine similarity between its topic embedding and personas embeddings. The retrieved personas $p_i \in P$ guide multi-hop synthesis paths. Given a persona p_i, a data point $X_i^{(K-1)}$, task demonstrations $\mathcal{D}_T$, and a predefined operation Op, we synthesize $X_i^{(K)}$ as,

$$X_i^{(K)} = \text{LLM}(X_i^{(K-1)}, t, p_i, Op, \mathcal{D}_T). \quad (3)$$

Prompts for persona-guided synthesis are shown in Appendix K. Combining multi-hop synthesis with attributes and personas increases the volume of diverse, task-relevant synthetic data.

3.3 Residual Connection Mechanism for Maintaining Task Relevance

Multi-hop synthesis guided by attributes and personas generates diverse, relevant data, but relevance decreases as hop depth K increases. For instance, synthesizing 10-hop neighbors introduces unrelated themes (Figure 15 in Appendix L). To address this drift from the original input at deeper synthesis depths, we introduce residual connections between a seed data point and its neighbors. Specifically, for any depth d where $1 < d \leq K$, we build the connections when $d \leq L$ where L is the depth of residual connection within the range $(1, K]$,

$$X_i^{(d)} = \begin{cases} \text{LLM}(X_i^{(d-1)}, \langle t, r, a \rangle_i^{(d-1)}, Op, \mathcal{D}_T), & L < d \\ \text{LLM}(X_i^{(d-1)}, \langle t, r, a \rangle_i^{(d-1)}, Op, \mathcal{D}_T, X_i^{(0)}), & d \leq L. \end{cases}$$

We illustrate the detail of residual connection in Appendix D. Figure 16 demonstrates a 10-hop synthesis using residual connections. Compared to Figure 15, the 10-hop neighbor in Figure 16 remains focused on the relevant topic.

4 Experiment

We evaluate AIDE to answer the following research questions (RQs): **(RQ1)** Can AIDE enable the fine-tuning of pretrained models that outperform those fine-tuned on human-curated data and data generated by SOTA synthesis methods? **(RQ2)** How does AIDE affect pretrained models' performance under different settings? **(RQ3)** Does the data from AIDE maintain relevance and diversity?

4.1 Experiment Setup

Datasets. We evaluate all methods across 5 tasks from BIG-Bench, 5 tasks from MMLU, 1 task from FinBen, as well as MedQA, ARC-Challenge, GSM8K, and TruthfulQA. Details of the benchmarks and statistics of the synthetic data from AIDE are provided in Appendix H and F.

Baselines. We use fine-tuned Mistral-7B, Llama-3.1-8B, and Llama-3.2-3B with human-generated (gold) data as baselines for comparison with the models fine-tuned using synthetic data from AIDE. We also compare AIDE with SOTA synthesis methods (Evol-Instruct, DataTune, and Prompt2Model) by fine-tuning Mistral-7B. A fine-tuned Mistral-7B using 250K synthetic data from Evol-Instruct [1] is utilized as Mistral-7B with Evol-Instruct. Details about the setups are provided in Appendix E.

Metrics. We evaluated all models using zero-shot accuracy as the primary metric on the benchmarks. For GSM8K, we report 8-shot maj@8 performance using prompts from Wang et al. (2023).

4.2 Performance and Analysis (RQ1)

In Table 2, the pretrained models fine-tuned with AIDE demonstrate comparable or superior performance to those fine-tuned with gold data. For example, on MMLU tasks, models fine-tuned with AIDE data outperform those trained on gold data by an average of $> 1.4\%$. In the CFA task, synthetic data from AIDE improves Mistral-7B and Llama-3.1-8B by at least $> 1.6\%$ compared to gold data. On ARC-Challenge, the Llama series fine-tuned with AIDE surpasses their counterparts fine-tuned on gold data. In GSM8K, pretrained models fine-tuned with AIDE perform comparably to those fine-tuned with gold data. On TruthfulQA, models fine-tuned with AIDE exceed those trained on gold data by an average of $> 15.0\%$. Similarly, on MedQA, AIDE improves pretrained models by more than $> 8.2\%$ on average. In Table 3 (BIG-Bench without training sets), Mistral-7B with AIDE significantly outperforms itself fine-tuned using Evol-Instruct, Prompt2Model and DataTune by $> 20.0\%$, and its pretrained model by $> 40.0\%$.

[1] https://huggingface.co/dreamgen/WizardLM-2-7B

	Fine-tuning with Data Source	MMLU					FinBen	ARC-Challenge	GSM8K	TruthfulQA	MedQA	Avg. (↑)	Avg. Δ (↑)
		Bio.	CS	Phi.	EE	Market.	CFA						
# Seed Data Points in **AIDE**		10	10	10	10	10	10	10	10	10	10	10	
Pretrained Mistral-7B	**AIDE (Ours)**	75.5%	57.0%	72.2%	60.7%	89.3%	41.0%	74.7%	59.1%	69.2%	44.0%	64.3%	7.0%
	Gold training data	73.2%	56.0%	71.1%	60.0%	85.9%	35.0%	79.4%	53.4%	49.9%	37.0%	60.1%	NA
Pretrained Llama-3.1-8B	**AIDE (Ours)**	74.2%	47.0%	63.0%	49.7%	82.1%	62.0%	69.8%	65.8%	69.2%	56.0%	63.9%	0.7%
	Gold training data	74.7%	48.1%	60.5%	50.1%	82.3%	61.0%	69.6%	68.2%	66.1%	54.0%	63.7%	NA
Pretrained Llama-3.2-3B	**AIDE (Ours)**	58.7%	43.4%	56.6%	54.5%	71.4%	54.0%	56.8%	45.1%	67.6%	51.0%	55.9%	1.5%
	Gold training data	60.2%	45.0%	55.6%	48.3%	70.7%	54.0%	56.5%	45.5%	64.9%	50.0%	55.1%	NA

Table 2: AIDE-generated data vs. human-curated training data for fine-tuning. We evaluate the performance of various zero-shot learning methods across MMLU, FinBen, ARC-Challenge, GSM8K (8-shot with maj@8), TruthfulQA, and MedQA. We highlight the **best** and **runner-up** performances. "Avg." represents the average performance across all benchmarks. For GSM8K, we fine-tune the models using 3.2K gold training data, matching the amount of synthetic data from AIDE. Results are obtained using the same parameter settings. Avg. $\Delta(\uparrow)$ represents the relative average improvement of models compared to those fine-tuned with gold data. "NA" indicates no difference from models fine-tuned with gold data.

Pretrained Model	Fine-tuning with Data Source	BIG-Bench					Avg. (↑)
		Code	C&E	Impl.	Math	Time	
Mistral-7B	**AIDE (Ours)**	91.7%	99.2%	67.9%	21.0%	90.3%	74.2%
	Prompt2Model	84.5%	41.2%	48.0%	4.7%	2.0%	36.1%
	DataTune	73.4%	33.8%	44.0%	8.1%	16.9%	35.2%
	Evol-Instruct	73.3%	73.2%	65.1%	14.1%	45.2%	54.2%
	Pretrained Model	46.7%	47.7%	61.1%	11.6%	1.4%	33.7%

Table 3: AIDE vs. SOTA Data Synthesis Methods. We compare the performance of various zero-shot learning approaches in Mistral-7B fine-tuned with AIDE and SOTA synthesis methods across five BIG-Bench tasks. The table follows a setup similar to Table 2. Notably, Evol-Instruct fine-tunes Mistral-7B with 250K synthetic data points.

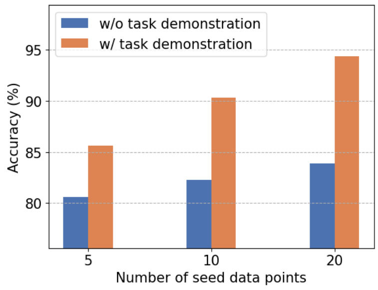


Figure 2: The effect of varying the number of seed data w/ and w/o task demonstration on the Time task from BIG-Bench.

Attributes	Personas	Residual Connections	Fine-tuned Mistral-7B
✔	✘	✘	60.1%
✘	✔	✘	49.3%
✔	✔	✘	72.2%
✔	✘	✔	75.0%
✔	✔	✔	**90.3%**

Table 4: Different core components of AIDE contribute to the synthetic data, improving the performance of Mistral-7B on the Time task from BIG-Bench. We highlight the **best** performance and the base performance is in Table 3.

This is because Prompt2Model focuses on generating task-specific data with limited diversity, whereas Evol-Instruct, despite its multi-hop synthesis structure, generates data without targeting a specific task.

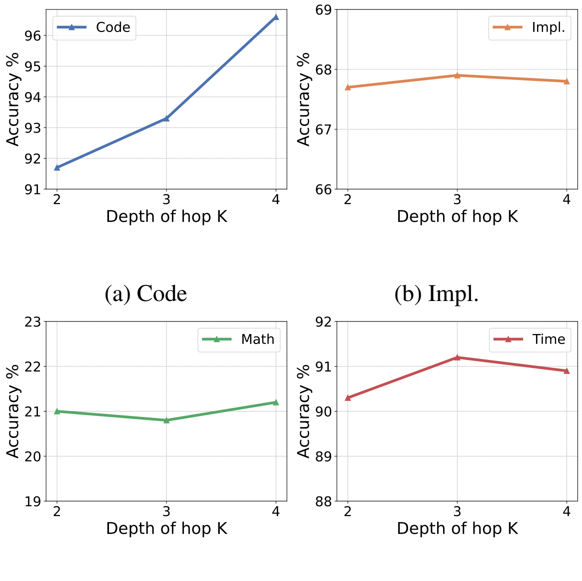
4.3 Ablation and Sensitivity Studies (RQ2)

We conduct ablation studies to empirically explore AIDE with pretrained models.

Effectiveness of Core Designs. Table 4 (Time task) demonstrates how AIDE's core components - attributes, personas, and residual connection - boost Mistral-7B's performance by enhancing the relevance and diversity of synthetic data. To preserve synthesis paths in multi-hop synthesis, we include either attributes or personas. Using only attributes or personas increases Mistral-7B's accuracy from 1.4% to 60.1% and 49.3%, respectively. With all three components combined, AIDE enables Mistral-7B to achieves 90.3% accuracy, the best performance by preserving synthesis paths and enhancing the relevance of synthetic data.

Effect of Seed Data and Task Demonstration. The amount of seed data affects initial synthetic data diversity, while task demonstration provide task-related examples to guide synthesis. Therefore, we analyze how the amount of seed data and inclusion of task demonstrations impact AIDE's synthetic data quality by fine-tuning Mistral-7B on equal amounts of data. In Figure 2, we show that increasing seed data in AIDE improves Mistral-7B's performance on the Time task through fine-tuning. Furthermore, including task demonstration in AIDE boosts Mistral-7B's accuracy by $> 10\%$ through fine-tuning, compared to using AIDE without task demonstrations.

Scaling with Data Quantity using Different Depth K. The multi-hop depth K determines the amount of AIDE's synthetic data, directly influencing fine-tuned model performance. Figure 3 shows increasing K from 2 to 4 significantly enhances Mistral-7B's performance on the code task after fine-tuning on AIDE data. However, for other tasks, performance gains gradually decrease with higher



(a) Code (b) Impl.

(c) Math (d) Time

Figure 3: Effect of data quantity with different number of K values in multi-hop synthesis based on the BIG-Bench.

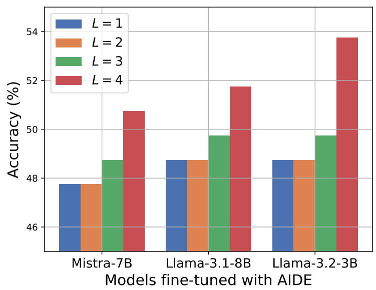


Figure 4: The effect of varying the depth of residual connections (L) when we fix the hop depth K as 4.

K values due to the inherent ability gap between the pretrained model and the LLM synthesizer.

Effect of Residual Connection. We use a contract task from LegalBench (Guha et al., 2023), setting the hop depth K to 4 while varying the depth of residual connections L. By synthesizing 5,682 training data points from 6 seeds, we analyze their impact on fine-tuning models. Figure 4 shows that as the multi-hop synthesis depth increases, a higher residual connection depth L improves the task relevance of the synthetic data, resulting in better model performance during fine-tuning.

Effect of Capability of LLMs. We investigate the impact of using different LLMs as components in AIDE by conducting experiments on 5 BIG-Bench tasks, using Claude Sonnet 3.5 and GPT-3.5-Turbo separately to synthesize data. As shown in Table 5, fine-tuning Mistral-7B with AIDE's synthetic data, generated with either Claude Sonnet 3.5 or GPT-3.5-Turbo as components, enhances the pretrained model Mistral-7B's performance by $> 40.0\%$.

Model	Synthetic method	BIG-Bench Benchmark					Avg.
		Code	C&E	Impl.	Math	Time	
Mistral-7B	**AIDE (Ours)** Claude Sonnet 3.5	**91.7%**	**99.2%**	67.9%	21.0%	**90.3%**	74.0%
	AIDE (Ours) GPT-3.5-Turbo	**91.7%**	86.3%	**82.5%**	**34.6%**	85.2%	**76.1%**
	-	46.7%	47.7%	61.1%	11.6%	1.4%	33.7%

Table 5: The performance of Mistral-7B fine-tuned with synthetic data from AIDE using different LLMs as synthesizer.

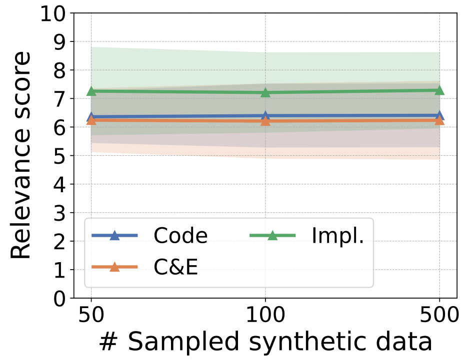


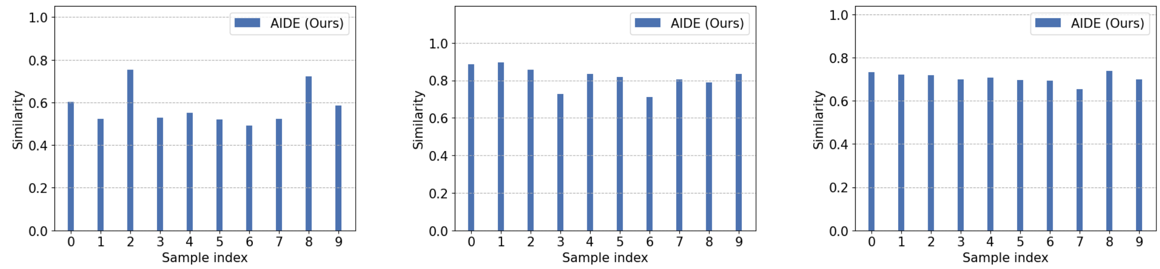
Figure 5: The relevance score related to the sampled synthetic data and task-related seed data from the Code task, the C&E task and the Impl. task.

4.4 Relevance and Diversity (RQ3)

We empirically investigate the relevance and diversity of synthetic data from AIDE. Appendix G provides details on synthetic data complexity.

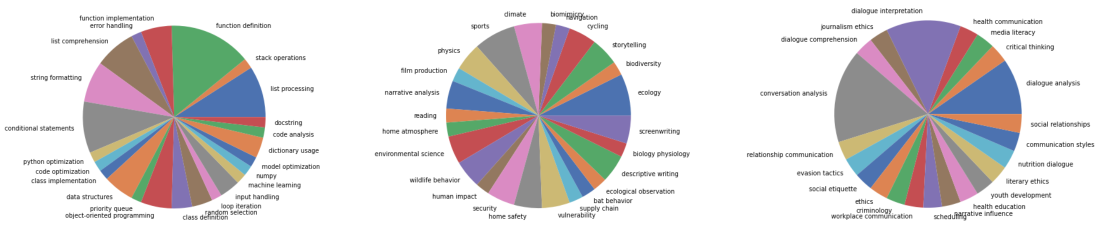
Analysis of Relevance. Since the seed data is task-specific, the synthetic data should also be task-relevant if it closely aligns with the seed data. To evaluate this, we randomly sample 10 synthetic data points per task from the Code, C&E, and Impl. tasks in the BIG-Bench benchmark. We use the Jina embedding model (Günther et al., 2023) to encode all data points, and compute the similarity between each synthetic data point and its corresponding seed data. As shown in Figure 6, the synthetic data exhibits strong relevance to the seed data, with an average similarity score above 0.5.

Additionally, we employ Claude Sonnet 3.5 to assess the relevance of synthetic data to the seed data across the three tasks. Claude assigns a relevance score from 0 to 10, with 10 indicating the highest relevance. As shown in Figure 5, the average scores range from 5 to 9, further confirming the task alignment of the synthetic data. The standard deviation arises because the samples contain data points with significant diversity, yet remain



(a) Synthetic data for the Code task (b) Synthetic data for the C&E task (c) Synthetic data for the Impl. task

Figure 6: For exploring the relevance of synthetic data with the seed data, we compute the similarity between the randomly sampled 10 synthetic data and the seed data per task. The tasks include Code, Impl. and C&E.



(a) Synthetic data for the Code task (b) Synthetic data for the C&E task (c) Synthetic data for the Impl. task

Figure 7: We assess the diversity of knowledge by randomly sampling 20 synthetic data points generated by AIDE for the Code, C&E, and Impl. tasks from BIG-Bench.

Benchmarks	Task Name	Diversity of Synthetic Data (AIDE)	Diversity of Gold Data
BIG-Bench	Code	0.59	**0.50**
	C&E	0.21	**0.15**
	Impl.	0.43	**0.40**
	Math	**0.49**	0.50
	Time	**0.70**	0.91
MMLU	Bio.	0.41	**0.29**
	CS	0.66	**0.24**
	Phi.	0.49	**0.30**
	EE	0.60	**0.18**
	Market.	0.44	**0.25**
ARC-Challenge	-	0.43	**0.18**
GSM8K	-	0.43	**0.21**
TruthfulQA	-	0.67	**0.20**

Table 6: Quantitative comparison of diversity between synthetic data from AIDE for different tasks and gold data from different tasks. We **highlight lower Self-BLEU scores**, which implies higher diversity.

relevant to the corresponding task.

Analysis of Diversity. AIDE expands attributes through using topics to retrieve personas from Persona Hub, which diversifies the data synthesis. To verify the diversity of synthetic data, we randomly sample 20 synthetic data per task from the Code, C&E, and Impl. tasks. Using the prompt shown in Figure19, we employ Claude Sonnet 3.5 to assess the diversity of the synthetic data based on relevant knowledge. As illustrated in Figure 7a, the sampled synthetic data for the Code task covers a variety of programming topics and operations. In the C&E and Impl. tasks, we observe that the synthetic data spans a wide range of knowledge domains, as shown in Figures 7b and 7c.

Additionally, following prior work (Ye et al., 2022a), we compute Self-BLEU (Zhu et al., 2018) to quantitatively assess the diversity of both synthetic and gold data. The results in Table 6 show that the synthetic data generated by AIDE achieves Self-BLEU scores comparable to those of gold data across most tasks, demonstrating its effectiveness in producing diverse synthetic data.

5 Conclusion

Existing data synthesis methods struggle to generate synthetic data that is both task-relevant and diverse for fine-tuning or require large seed datasets. In this paper, we introduce AIDE, a novel framework that enables task-relevant, diverse, and high-quality data expansion from few seed examples. It features multi-hop synthesis guided by attributes and personas, along with a residual connection to mitigate irrelevance at deeper hops. Our experiments show that fine-tuning Mistral-7B and Llama models with AIDE outperforms the models fine-tuned with gold data and SOTA synthesis methods.

References

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *Preprint*, arXiv:2406.20094.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

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

Saumya Gandhi, Ritu Gala, Vijay Viswanathan, Tongshuang Wu, and Graham Neubig. 2024. Better synthetic data by retrieving and transforming existing datasets. *Preprint*, arXiv:2404.14361.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *Preprint*, arXiv:2310.19923.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large language models on graphs: A comprehensive survey. *arXiv preprint arXiv:2312.02783*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, pages 46534–46594. Curran Associates, Inc.

Igor Melnyk, Pierre Dognin, and Payel Das. 2022. Knowledge graph generation from text. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1610–1622, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, and et al. Agarwal. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12.

Feifan Song, Bowen Yu, Hao Lang, Haiyang Yu, Fei Huang, Houfeng Wang, and Yongbin Li. 2024. Scaling data diversity for fine-tuning language models in human alignment. *Preprint*, arXiv:2403.11124.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Vijay Viswanathan, Chenyang Zhao, Amanda Bertsch, Tongshuang Wu, and Graham Neubig. 2023. Prompt2Model: Generating deployable models from natural language instructions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 413–421, Singapore. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zifeng Wang, Chun-Liang Li, Vincent Perot, Long Le, Jin Miao, Zizhao Zhang, Chen-Yu Lee, and Tomas Pfister. 2024. CodecLM: Aligning language models with tailored synthetic data. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Mexico City, Mexico. Association for Computational Linguistics.

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

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2024. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyan Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, Zhiyang Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Jiajia Huang, Xiao-Yang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. 2024. Finben: A holistic financial benchmark for large language models. *Preprint*, arXiv:2402.12659.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *Preprint*, arXiv:2306.05443.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. 2022. ZeroPrompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4235–4252, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022a. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2022b. ProGen: Progressive zero-shot dataset generation via in-context feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3671–3683, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Graham Neubig, and Tongshuang Wu. 2024a. Self-guide: Better task-specific instruction following via self-synthetic finetuning. In *First Conference on Language Modeling*.

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

Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Minghao Li, Fei Huang, Nevin L. Zhang, and Yongbin Li. 2024c. Tree-instruct: A preliminary study of the intrinsic relationship between complexity and alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia. ELRA and ICCL.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Detailed Related Work

Data Synthesis for Instruction Tuning in Open Domains. OpenAI has utilized human annotators to develop diverse instruction-response datasets for training InstructGPT (Ouyang et al., 2022). Similarly, Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023) explore open-domain instruction tuning using the Llama model. Evol-Instruct (Xu et al., 2024) offers fine control over instruction complexity, while Tree-Instruct (Zhao et al., 2024c) underscores the significance of complexity in LLM alignment. CodecLM (Wang et al., 2024) adapts instructions for various tasks. However, these methods lack domain specificity, often introducing irrelevant data. For instance, mixing medical and coding data can negatively impact the fine-tuning process for medical question-answering tasks.

Data Synthesis for Instruction Tuning in Task-specific Domains. Recent research has focused on generating diverse and relevant datasets through data synthesis. For example, ZeroGen (Ye et al., 2022a) synthesizes data from task-specific prompts, though challenges arise in domains like multiple-choice, where the label set can be infinite. Methods such as DataTune (Gandhi et al., 2024) and Prompt2Model (Viswanathan et al., 2023) transform existing datasets based on task descriptions, but they rely on large pre-existing collections. Approaches like Self-Guide (Zhao et al., 2024a) and ProGen (Ye et al., 2022b), which use limited examples for guiding synthesis, lack sufficient diversity in the generated data.

B Multi-Hop Synthesis

The Figure 8 shows an example of the multi-hop synthesis, which the seed data $X_i^{(0)}$ is used to synthesize its 1-hop neighbors $X_1^{(1)}$ and $X_2^{(1)}$ during the 1-hop synthesis. Similarly, each 1-hop neighbor can be applied to generate 2-hop neighbors of $X_i^{(0)}$. For each input data $X_i^{(0)}$ where $1 \leq i \leq n$, we recursively synthesis data using the same pattern until reaching the depth of K.

C An Example of Unfolded Multi-Hop Synthesis

Figure 9 illustrates an example of unfolded multi-hop synthesis. In this example, we set $K = 2$. $X_i^{(0)}$ is one of the seed data point and $X^{(1)} = \{X_1^{(1)}, X_2^{(1)}, ..., X_{m_1}^{(1)}\}$ represents synthetic data from 1-hop synthesis while $X^{(2)} = \{X_1^{(2)}, X_2^{(2)}, ..., X_{m_2}^{(2)}\}$ represents synthetic data from 2-hop synthesis. r is the relation between a topic t and knowledge attribute a. The predefined operation Op is the abbreviation of operation. Green area includes a path of synthesis showing the relevance between two data points. Orange area shows a path to synthesize data with diversity and relevance. We zoom in one of the branches related to $X_3^{(1)}$ in 2-hop synthesis. Table 1 demonstrates an example of the synthesis.

D Residual Connection

We introduce residual connections between a seed data point and its neighbors. Specifically, for any depth d where $1 < d \leq K$, we establish connections when $d \leq L$ where L is the depth of residual connection within the range $(1, K]$. For example, in Figure 9, when $K = 2$, setting $L = 2$ allows connections between the seed data and all neighbors at hop depth 2, ensuring seed information is available for generating the neighbors.

Experiments in Figure 4 demonstrate that when the hop depth K is large, applying residual connections with a greater depth L enhances the relevance of the synthetic data, leading to improved performance in the fine-tuned model. However, as hop depth K increases, removing low-relevance neighbors instead of using residual connections to retain them can lead to a reduction in the amount of synthetic data.

E Detailed Experimental Setup

Data Synthesis Setup. We configure the SOTA data synthesis methods using their default settings. Since BIG-Bench lacks a training set, we sample 10 task-related seed data points per task from Hugging Face datasets to generate synthetic data. For

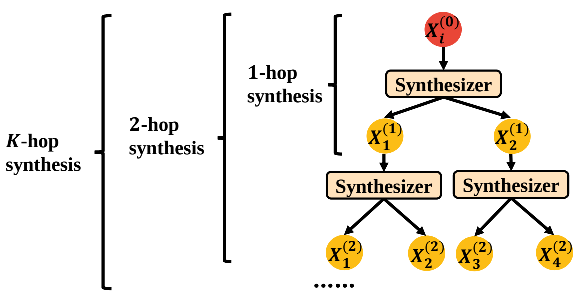


Figure 8: Multi-hop synthesis with the depth of hop K use a seed data point to synthesize new data points. The data points with yellow color represent synthetic data while we use red color to denote a seed data point.

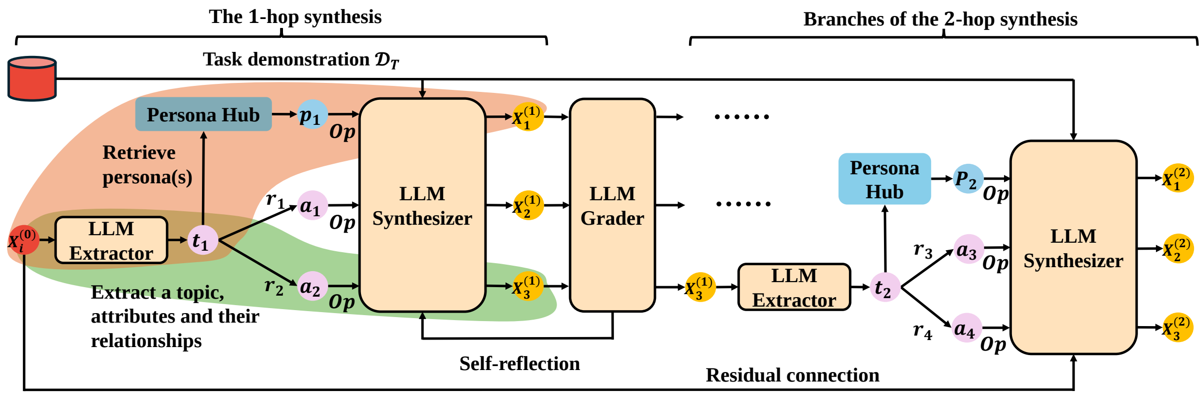


Figure 9: An example of unfolded multi-hop synthesis when $K = 2$.

the remaining benchmarks, we similarly sample 10 seed data points per task from their respective training sets to produce synthetic data. We set the depth of hop $K = 2$ in the multi-hop synthesis. We employ Claude Sonnet 3.5 as the LLM generator, the LLM synthesizer, the LLM grader and the LLM annotator in AIDE. We require the LLM to generate $\mathcal{A}^{(K)}$ of a data point $X_i^{(K)}$, which consists of 1 topic and 3 most related attributes. Each triplet in $\mathcal{A}^{(K)}$ followed by 3 operations: concretizing, adding constraint and adding reasoning. With a topic, we retrieve top-5 related personas to diversify attributes.

Fine-tuning Setup. We applied the LoRA (Hu et al., 2022) to fine-tune Mistral-7B. We randomly split 10% of the synthetic data as validation set while the rest of synthetic data as training set. The process was carried out over 10 epochs with batch size equal to 10. We select learning rate $5e-5$ with LoRA's α parameter as 16 and choose the run with the lowest validation loss at any point. We used the AdamW optimizer (Loshchilov and Hutter, 2019) and set LoRA $r = 8$. We conduct our training on a server with 8 NVIDIA A100 GPUs.

Self-Reflection for Synthetic Data To ensure the correctness, relevance, and diversity of synthetic data, we apply existing self-reflection techniques (Madaan et al., 2023; Pan et al., 2024) after synthesis (Figure 1). A LLM grades synthetic data $X_i^{(K)}$ on these aspects, providing a score (from 1 to 10) and feedback. Data exceeding a score threshold (i.e., threshold equal to 5) is added to the dataset; otherwise, it undergoes limited re-synthesis iterations. A LLM annotator then labels the data, with self-reflection ensuring labeling correctness. Related prompts are shown in Appendix N.

Benchmarks	Task Name	Depth of K	Amount of seed Data	Quantity of Synthetic Data
BIG-Bench	Code	2	10	3.0K
	C&E	2	10	3.2K
	Impl.	2	10	3.1K
	Math	2	10	3.1K
	Time	2	10	3.2K
MMLU	Bio.	2	10	3.4K
	CS	2	10	3.2K
	Phi.	2	10	3.4K
	EE	2	10	3.0K
	Market.	2	10	3.3K
ARC-Challenge	-	2	10	3.3K
GSM8K	-	2	10	3.2K
TruthfulQA	-	2	10	3.1K
FinBen	CFA	2	10	893
MedQA	-	2	10	2.2K

Table 7: Statistics of synthetic data. Note that we adapt the self-reflection mechanism to enhance data quality, which also filters out some synthetic data.

F Statistics of Synthetic Data

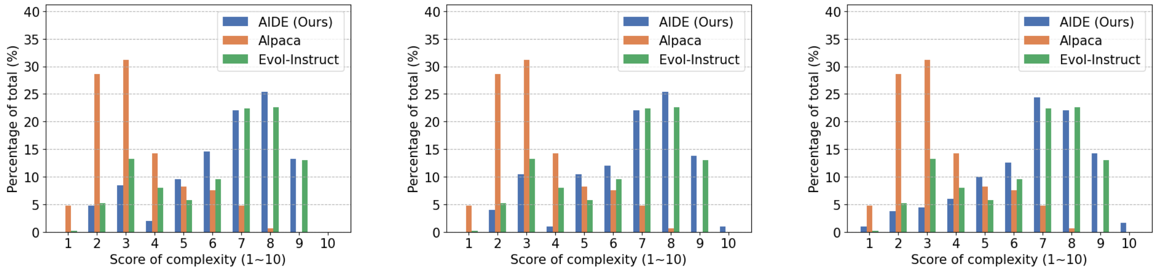
In Table 7, we demonstrate the amount of seed data used and the quantity of data synthesized in AIDE. Specifically, using $K = 2$ and 10 seed data points for each task, AIDE generates approximately 3K new data points in about 20 hours when adapting the self-reflection mechanism to improve the quality of new data.

G Detailed Analysis of Relevance, Diversity and Complexity (RQ3)

We conduct experiments to assess whether the synthetic data generated by AIDE preserves its complexity.

G.1 Analysis of Complexity

Similar to Evol-Instruct (Xu et al., 2024) using 5 predefined operations to expand the complexity of synthetic data, AIDE utilizes 3 predefined opera-



(a) Synthetic data for the Code task (b) Synthetic data for the C&E task (c) Synthetic data for the Impl. task

Figure 10: The complexity of randomly sampling 500 synthetic data from AIDE based on different domains, including code, cause and effect and implicatures. We also compare the complexity of randomly sampling 500 synthetic data from the state-of-the-art data synthesis methods including Alpaca and Evol-Instruct.

tions including reasoning, constraint and concrete following triplets from $\mathcal{A}$ to expand the complexity during data synthesis. For verifying the complexity of synthetic data from AIDE, we randomly sample 500 synthetic data from different synthetic methods including Alpaca, Evol-Instructs and our AIDE. Then we apply Claude Sonnet 3.5 to evaluate the complexity of synthetic data using the same prompt as that from Evol-Instruct. We plot the distribution of score of complexity ranging from 1 to 10, shown on Figure 10. We find that most of synthetic data from AIDE and Evol-Instruct obtain the score of complexity higher than 5, when comparing with that from Alpaca. It is worth mentioning that AIDE only uses 3 predefined operations less than the operations applied in Evol-Instruct while having the synthetic data with comparable complexity.

G.2 Visualization

We follow the approach in (Zhao et al., 2024b) and analyze the coverage of synthetic data from AIDE in the embedding space. Specifically, we use the jina-embeddings-v2-base-code (Günther et al., 2023) to embed data points about coding while employ jina-embeddings-v2-base-en to encode other text data. With the embeddings, we utilize t-SNE (van der Maaten and Hinton, 2008) to project embeddings into a two-dimensional space. We adopt the real data from the code line description task and the C&E task as baselines to demonstrate the coverage of synthetic data from AIDE.

In Figure 11a, we observe that the embedding clusters of synthetic data via AIDE and the embeddings of all real data from the Code task appear to be largely disjoint. Figure 11b demonstrates that the synthetic data has a larger range which covers all real data from the C&E task. This supports a conclusion that AIDE with few seed data related to specific tasks systematically cover different distributions of the target task space, and therefore fine-tuning Mistral-7B with synthetic data from AIDE leads to a positive effect on the improvement of performance of Mistral-7B in specific tasks.

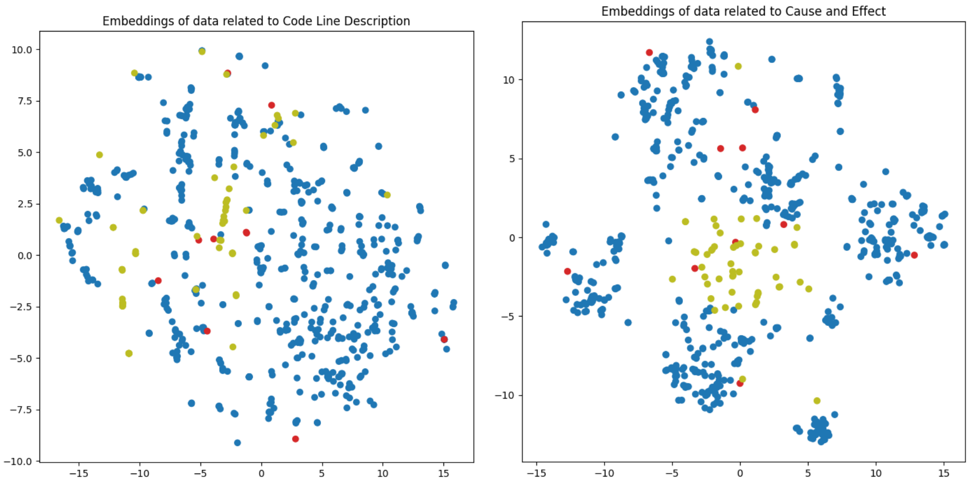
H Benchmark Statistics

The details of the benchmarks we employ in the paper are included below:

- **BIG-Bench** (bench authors, 2023) includes over 200 tasks that are currently challenging for language models, encompassing a wide range of categories. We selected the code line description task, cause and effect task, implicatures task, elementary math task and temporal sequence task, totally 5 tasks, which involve coding understanding, causal reasoning, logical reasoning. The selected tasks without training sets include 60, 153, 492, 7.688k and 1k data points in their test sets, respectively.

- **MMLU** (Hendrycks et al., 2021) is designed to evaluate the broad capabilities of language models across 57 tasks. We select 5 tasks from the benchmark, including high school biology, college computer science, philosophy, electrical engineering and marketing, which respectively contain 310, 100, 311, 145 and 234 data point in the test sets.

- **ARC** (Clark et al., 2018) is a set of grade-school science questions, which are designed to test a model's ability to perform complex reasoning. We select ARC-Challenge with the more difficult questions that are particularly challenging for AI models because they often require multiple steps of reasoning, inference,



(a) Synthetic data (Code task)

(b) Synthetic data (C&E task)

Figure 11: We observe that randomly sampling 600 synthetic data generated by AIDE using the seed data covers the all real test data from two tasks in the regions of embedding space, after projecting to two dimensions via t-SNE.

and external knowledge beyond the text provided in the question. We apply 1.17k testing data points in this task to test LLMs.

- **GSM8K** (Cobbe et al., 2021) is a dataset of 8.5K high quality linguistically diverse grade school math word problems. The dataset was created to support the task of question answering on basic mathematical problems that require multi-step reasoning. We select the main subset which has 7.47k training data points and 1.32k testing data points.

- **TruthfulQA** (Lin et al., 2022) is a benchmark to measure whether a language model is truthful in generating answers to questions. We select the multiple choice sets which contains 817 questions for testing.

- **MedQA** (Jin et al., 2020) is a comprehensive resource designed to enhance medical question-answering systems. It comprises 10,178 multiple-choice questions sourced from medical exams across the United States, Mainland China, and Taiwan. Each question is accompanied by several answer options, with the correct answer clearly indicated. We select 1,956 data points for the training set and 217 for the validation set. Additionally, we sample 10 seed data points to synthesize 2,173 data points through AIDE.

- **FinBen** (Xie et al., 2024) is part of the PIXIU project (Xie et al., 2023), an open-source initiative aimed at developing, fine-tuning, and evaluating large language models (LLMs) in the financial domain. PIXIU encompasses various components including FinBen, a financial language benchmark. The CFA task consists of 1.03k data points, which we divide as follows: 100 data points for the test set, 804 as gold training data, 89 for the validation set, and 10 as seed data points to synthesize 893 additional data points through AIDE.

Task Name	Abbreviation	# Test data
Code Line Descriptions	Code	60
Cause and Effect	C&E	153
Implicatures	Impl.	492
Elementary Math	Math	7,688
Temporal Sequence	Time	1,000
High School Biology	Bio.	300
College Computer Science	CS	100
Philosophy	Phi.	311
Electrical Engineering	EE	145
Marketing	Market.	234
Flare-cfa	CFA	100
ARC-Challenge	-	1,170
GSM8k	-	1,320
TruthfulQA	-	817
MedQA	-	100

Table 8: Data statistic of selected tasks from BIG-Bench, MMLU, ARC-Challenge, GSM8K and Truthful QA.

I Prompt for Extracting a Topic and Knowledge Attributes

We utilize Claude Sonnet 3.5 as the LLM extractor in AIDE, as shown in Figure 1. In Figure 12, we demonstrate a prompt used in the LLM extractor to extract a topic and knowledge attributes.

J Prompt for Synthesizing Data Points with a Triplet and an Operation

We apply Claude Sonnet 3.5 as the LLM synthesizer in AIDE, as illustrated in Figure 1. Figure 13 provides an example of a prompt used by the LLM synthesizer to generate a new data point, incorporating a triplet and a constraint operation.

K Prompt for Synthesizing Data Points with a Topic and Personas

We use Claude Sonnet 3.5 as the LLM synthesizer to generate new data points based on a persona and a constraint operation. Figure 14 demonstrates a prompt provided to the LLM synthesizer, incorporating both a persona and a constraint operation.

L An Example of 10-hop Synthesis without Residual Connection

Figure 15 presents an example of 10-hop synthesis without applying the residual connection. In multi-hop synthesis, when the hop depth K becomes large (e.g., $K = 10$), the synthetic data tends to include more irrelevant information.

M An Example of 10-hop Synthesis with Residual Connection

We introduce the residual connection mechanism in AIDE , as detailed in Section 3.3 and Figure 9. Figure 16 illustrates an example of 10-hop synthesis incorporating the residual connection.

N Prompt for Self-Reflection

During the self-reflection, when multi-hop synthesis synthesizes data through knowledge attributes for maintaining relevance, we apply a LLM as grader to check the relevance of the synthetic data and obtain a relevance score. Similarly, while we generate synthetic data through multi-hop synthesis using persona to expand diversity, a LLM grader checks the diversity of the synthetic data and return a diversity score. We show the prompt about checking relevance and diversity in Figure 17. With a self-reflection prompt in Figure 18, we collect the score of diversity and relevance as the feedback to process the synthetic data.

O Ethical Considerations

While AIDE is an effective framework for generating diverse, task-relevant data, it's important to consider the ethical implications. With only a few seed data points, AIDE leverages LLMs to extract, synthesize, grade, and annotate instruction-response pairs. However, like human annotators, LLMs can occasionally generate unethical, toxic, or misleading content. Although we use self-reflection techniques during synthesis, it's essential to adopt proven methods for detoxifying and reducing bias in LLM outputs. Stricter inspection and filtering rules should also be applied. Given AIDE's flexibility, future advances in bias mitigation and fairness can be integrated as additional modules.

P Limitations

We recognize AIDE 's limitations in the following two areas, which can serve as inspiration for future research opportunities in the field of data synthesis.

Ethical Consideration. Since our method AIDE relies on an LLM to serve as the extractor, synthesizer, grader, and annotator, it may inherit biases and fairness issues from the underlying LLM. However, AIDE stands to benefit from improved LLMs that incorporate advanced techniques for reducing bias and enhancing fairness.

Cognitive Process. While AIDE helps base models improve their performance in the Math task, the zero-shot performance of the fine-tuned base models remain around 20%. In the future, a potential future direction is to integrate Chain-of-Thought techniques into AIDE, such that AIDE can provide better synthetic data to enhance reasoning steps of the base models though fine-tuning.

Prompt for extracting a topic and knowledge attributes of a data point

I want you to act as an instruction analyzer.
Given an instruction, you should recognize its topic and knowledge attributes. You need to list at most 2 knowledge attributes while each knowledge attributes should be transferable and concise with one word or two words. Your should only output the topic within <Topic></Topic> XML tags and output knowledge attributes within <Attributes></Attributes> XML tags.
Follow the examples below to analyze <The Given Instruction>
<Example>
<The Given Instruction> As a sports commentator, describe the winning play in the final seconds of a championship game. </The Given Instruction>
<Topic> creative writing </Topic>
<Attributes> role-play, sports </Attributes>
</Example>
... Some examples ...
<The Given Instruction> {Here is instruction.} </The Given Instruction>

Figure 12: Prompt for extracting a topic and knowledge attributes.

Prompt for synthesis with a triplet and a constraint operation

I want you act as a Prompt Writer. Your objective is to rewrite a given prompt into a more complex instruction to make those famous AI systems (e.g., chatgpt and GPT4) a bit harder to handle. But the rewritten prompt must be reasonable and must be understood and responded by humans. You SHOULD generate the rewritten prompt within <Rewritten Prompt></Rewritten Prompt> XML tags through complicating <The Given Prompt>, such that <Rewritten Prompt> meets the following <EXPECTATIONS>
<EXPECTATION 1> The <Rewritten Prompt> SHOULD BE SIMILAR TO {a seed data point (a residual connection)}.
</EXPECTATION 1>
<EXPECTATION 2> The <Rewritten Prompt> can be obtained by adding simple constraints into content in <The Given Prompt>.
</EXPECTATION 2>
<EXPECTATION 3> The <Rewritten Prompt> is related to {topic} using {knowledge attribute}.
</EXPECTATION 3>
<EXPECTATION 4> Make the <Rewritten Prompt> become as SHORT as possible.
</EXPECTATION 4>
<EXPECTATION 5> <The Given Prompt>, <Rewritten Prompt>, 'given prompt' and 'rewritten prompt' are not allowed to appear in <Rewritten Prompt>.
</EXPECTATION 5>

Follow the below examples to generate <Rewritten Prompt> by {adding constraints} into <The Given Prompt>.

... Some examples ...

<The Given Prompt>{Here is instruction.}</The Given Prompt>

Figure 13: Prompt for synthesis with a triplet and an operation

Prompt for synthesis with a persona and a constraint operation

A persona is the aspect of someone's character. You can use the given character to generate a <Created Prompt>. Your goal is to use <The Given Persona> and an operation to create a <Created Prompt> different from <The Given Prompt>.You SHOULD generate the <Created Prompt> through the following actions:

<Action 1> <Created Prompt> should related to the topic {a topic}. </Action 1>
<Action 2> You need to use <The Given Prompt> and the character from <The Given Persona> to generate <Created Prompt>. </Action 2>
<Action 3> <Created Prompt> should be different from <The Given Prompt>. </Action 3>
<Action 4> The <Created Prompt> can be obtained by adding simple constraints into the generated content. </Action 4>
<Action 5> You should try your best not to make the <Created Prompt> become verbose. </Action 5>
Follow the examples below to generate <Created Prompt> and output the <Created Prompt> within <Created Prompt></Created Prompt> XML tags.

... Some examples ...

<The Given Prompt> {Here is instruction.} </The Given Prompt>
<The Given Persona> {Here is a persona.}</The Given Persona>

Figure 14: Prompt for synthesis with persona and a constraint operation

An example of 10-hop synthesis without the residual connection

Input: Create an example of an economic and a political issue.
1-**hop neighbor**: Design a scenario that incorporates both an economic and a political quandary, and introduce a critical factor that significantly influences the outcome.
......
10-**hop neighbor**: Construct a multifaceted narrative that intricately weaves together various economic, political, and sociocultural elements, entwined with an unexpected turn of events that dramatically shifts the trajectory, demanding a methodical strategy to maneuver through the complex array of repercussions, emphasizing an in-depth exploration of ethical dilemmas, alongside an additional dimension probing into the subconscious drivers behind the choices made by each character, all while taking into account the impact of technological advancements and how they shape the development of the storyline.

Figure 15: An example of 10-hop synthesis without the residual connection. When the depth of hop K is large in multi-hop synthesis (i.e., $K = 10$), more irrelevant information can be introduced in the synthetic data.

An example of 10-hop synthesis with the residual connection

Input: Create an example of an economic and a political issue.
1-**hop neighbor**: Develop a multifaceted scenario encompassing interconnected economic and political challenges, each influencing the other in a complex and nuanced manner.
. . . . . .
10-**hop neighbor**: Craft an engaging narrative interlacing complex economic and political dilemmas, highlighting their symbiotic nature and profound impact on each other, necessitating a nuanced comprehension of their intricate interdependencies for adept navigation.

Figure 16: An example of 10-hop synthesis with the residual connection shown in Figure 9.

Prompt in self-reflection for evaluating the relevance/diversity score of the synthetic data

I want you to act as a domain expert to rate the relevance of <The Given Prompt> and <The Original Prompt>.

You should give an overall score on a scale of 1 to 10, where a higher score indicates the <The Given Prompt> is more relevant to/different from <The Original Prompt>.
You must just give <Score> without any other reasons within the <Score></Score> xml tags.
Follow the examples below to analyze and rate relevance of <The Given Instruction> and <The Original Prompt> in <Score>.

... N Examples ...

Your output should follow the format of examples, which means preserve the same format and show the score within <Score></Score> xml tags.
<The Original Prompt> {Here is the original instruction.} </The Original Prompt>
<The Given Prompt> {Here is the given prompt.} </The Given Prompt>

Figure 17: Prompt in the self-reflection can be used to evaluate the relevance score or diversity score of the synthetic data

Prompt for self-reflection to improve the synthetic data

I want you to act as a professional data generator.

The <Score> from grader shows that the <The Given Prompt> is not relevant to <Pre-prompt> (or the <The Given Prompt> is highly similar to <Pre-prompt>).
You are asked to rewrite <The Given Prompt> as the <Improved Prompt> using the <Pre-prompt>.
Generate <Improved Prompt> that improves the <Score> of relevance (or <Score> of diversity) by making <Improved Prompt> relevant to <Pre-prompt> (or by making <Improved Prompt> different from <Pre-prompt>).
Must only generate <Improved Prompt> within the <Improved Prompt></Improved Prompt> XML tags.

... N Examples ...

<Pre-prompt> {Here is the pre-prompt.} </Pre-prompt>
<The Given Prompt> {Here is the given prompt.} </The Given Prompt>
<Score> {Here is score.} </Score>

Figure 18: Prompt for self-reflection, which can be used to improve the relevance or diversity.

Prompt for a LLM judging the diversity of the synthetic data

You are a helpful AI assistant for evaluating and rating the difficulty and complexity of the following question.

Given an instruction, you should recognize its related knowledge without any explanation.
List several most related knowledge, the knowledge should be transferable, so that LLM can leverage them to answer similar questions.
Each knowledge should be concise with one word or two words.
Follow the examples below to analyze <The Given Instruction>.
<Example>
<The Given Instruction> As a sports commentator, describe the winning play in the final seconds of a championship game. </The Given Instruction>
<Knowledge> sports </Knowledge>
</Example>

... N Examples ...

You must just give the knowledge within the <Knowledge></Knowledge> XML tags without any other reasons.
<The Given Instruction> {Here is the given instruction} </The Given Instruction>

Figure 19: A LLM uses the prompt to judge the diversity of the synthetic data from the perspective of knowledge.

Prompt for a LLM judging the relevance of the synthetic data

You are a helpful AI assistant for evaluating and rating the difficulty and complexity of the following question.

We would like you to evaluate and rate the relevance of <Instruction1> and <Instruction2> .
You should give an overall score on a scale of 1 to 10, where a higher score indicates higher relevance between two instructions. You must just give a score without any other reasons.
Put the score within the <Score></Score> XML tags.

... N Examples ...

<Instruction1> {Here is the Instruction1} </Instruction1>
<Instruction2> {Here is the Instruction2} </Instruction2>

Figure 20: A LLM uses the prompt to judge the relevance of the synthetic data from the perspective of knowledge.