Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section C in the supplementary materials
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code and data are released at `https://github.com/shirley-wu/daco`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section 5 in the main content, and Section D in the supplementary materials
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Did not perform experiments of multiple random seeds due to resource constraints
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Section D in the supplementary materials
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Section 3 in the main content
   (b) Did you mention the license of the assets? [Yes] Spider [1] releases their dataset using Apache-2.0 license
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Dataset is released at `https://github.com/shirley-wu/daco`
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The annotators are trained and we have obtained their consent
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data is public data and does not contain personally identifiable information or offensive content
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] Annotations are done through internal annotators and instructions are potentially confidential information
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] Annotations are done through internal annotators and costs are potentially confidential information

A Appendix

Website is at `https://shirley-wu.github.io/daco/index.html`. Data and code are released at `https://github.com/shirley-wu/daco`. Croissant metadata record is at `https://github.com/shirley-wu/daco/blob/main/data/croissant.json`. We license our resources under Apache-2.0 license.

We thereby state that we bear all responsibility in case of violation of rights, etc., and confirmation of the data license.

B Dataset Documentation

Below are dataset documentation following the framework from datasheets for datasets:

Motivation:

- *For what purpose was the dataset created?* - For the novel task of data analysis as explained in the main content.
- *Who created the dataset and on behalf of which entity?* - This dataset is created during a collaboration of ByteDance AI Lab and University of California, Los Angeles.
- *Who funded the creation of the dataset?* - ByteDance

Composition:

- *What do the instances that comprise the dataset represent?* - Each instance contains a database of tabular data, a question, a reasoning process including code snippets, and a final answer. Everything is in text format, except the database is stored as `pd.DataFrame`
- *How many instances are there in total?* - As detailed in Table 1 in the main content.
- *Does the dataset contain all possible instances or is it a sample of instances from a larger set?* - The dataset is not a sample from a larger set.
- *What data does each instance consist of?* - Raw data.
- *Is there a label or target associated with each instance?* - Yes, as explained in Section 3 in the main content.
- *Is any information missing from individual instances?* - No
- *Are relationships between individual instances made explicit?* - N/A
- *Are there recommended data splits?* - Yes. We split the dataset randomly, and encourage people to follow this split for reproductivity. We also curate human annotations only for the test set.
- **Are there any errors, sources of noise, or redundancies in the dataset?** - Yes. The input questions and answer annotations are generated by ChatGPT, which will inevitably contain errors. We try to manage the affect by manually filtering the questions, and by curating a test set of human refined answer annotations.
- *Is the dataset self-contained, or does it link to or otherwise rely on external resources?* - Self-contained.
- *Does the dataset contain data that might be considered confidential?* - No.
- *Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?* - No.
- *Does the dataset identify any subpopulations?* - No.
- *Is it possible to identify individuals, either directly or indirectly from the dataset?* - No.
- *Does the dataset contain data that might be considered sensitive in any way?* - No.

Collection process: as described in Section 3 in the main content.

Preprocessing/cleaning/labeling: we release the raw text data and do not perform any preprocessing/cleaning/labeling of the texts.

Uses:

- *Has the dataset been used for any tasks already?* - The data analysis task, as in the main content
- *Is there a repository that links to any or all papers or systems that use the dataset?* - `https://github.com/shirley-wu/daco`
- *What (other) tasks could the dataset be used for?* - As in the main content
- *Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?* - N/A
- *Are there tasks for which the dataset should not be used?* - N/A

Distribution:

- *Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?* - Yes
- *How will the dataset will be distributed?* - `https://github.com/shirley-wu/daco`
- *When will the dataset be distributed?* - Already released
- *Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?* - Yes, Apache-2.0 license
- *Have any third parties imposed IP-based or other restrictions on the data associated with the instances?* - No
- *Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?* - No

Maintanance:

- *Who will be supporting/hosting/maintaining the dataset?* - The dataset is not planned to be a dynamic dataset, but the authors will keep maintaining the github repo
- *How can the owner/curator/manager of the dataset be contacted?* - Github or email `xueqing.wu@cs.ucla.edu`
- *Is there an erratum?* - No
- *Will the dataset be updated?* - No unless to correct errors
- *Will older versions of the dataset continue to be supported/hosted/maintained?* - N/A
- *If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?* - Yes, please feel free to do that as long as citing our work and following the license

C Limitations

While we put forth to construct the first of its kind dataset, DACO, for the comprehensive data analysis task, the challenging nature of the data analysis process (which often requires certain domain expertise) itself presents two major limitations of this work: **(1) It is expensive to build expert-annotated dataset.** In our work, the large-scale annotations are automatically generated by GPT-4, and their quality cannot always be well guaranteed. Although we curate a test set refined by humans, those answers are initially generated by GPT-4, which may introduce biases to human annotators during the refinement process for the final answer. Annotations curated by experts from scratch might have higher quality, but they are indeed quite costly and can create other sort of alignment problems. **(2) It is nontrivial to evaluate model generations.** Evaluating the quality of the analyses is by itself challenging and requires data science expertise. For automatic evaluation, we use ChatGPT to rank the helpfulness of model generations, which can partially but not perfectly align with human preference. Additionally, ChatGPT cannot always robustly evaluate the correctness of model generations. We use an off-the-shelf NLI model to evaluate the entailment probability between human-refined ground truths and the model generations, which can partially reflect the correctness of model generations. However, the entailment probability prediction can sometime propagate errors which lead to false positives or negatives. We make efforts to alleviate such an issue by additionally collecting human evaluations, which are supposed to better reflect the answer quality, despite that humans can occasionally exhibit subjective evaluation patterns. Notice that our annotators do not fully check the correctness of the generated answers, where we task them to focus more on the helpfulness metrics defined in this work.

D Implementation Details

For **zero-shot API-based systems** including ChatGPT and GPT-4, we evaluate two settings, directly reading the table content, and using code generation. For the former setting, we linearize the table content into text representation as model input. Due to token limit, we feed the first 20 rows as input, which covers the full content of 93% tables. For the code generation setting, we employ the pipeline described in Figure 2(b) in the main content. When the generated code causes a syntax or runtime error, we re-sample the model until the generated code can be executed. We allow up to 5 resamplings for each turn. We use the `gpt-3.5-turbo-16k-0613` API for ChatGPT and `gpt-4-32k` API for GPT-4. We limit the number of total coding turns maximally at 9. For annotation generation where GPT-4 self-correction is allowed, we limit the number of self-correction within 2 for each turn and 4 for the whole session.

For **finetuned models** including SFT, RLHF and fine-grained RLHF, we use CodeGeeX2-6B [2] as the base model. We first train the SFT model using GPT-4 annotations, and then train our RLHF models on top of the SFT model. When training R_{a+c} and R_r, we initialize the model from the SFT model. When training our fine-grained RLHF model, we initialize the value model V from R_{a+c}, and initialize the policy model π from the SFT model. In inference, we use nucleus decoding with p = 0.9 and temperature = 1.0. Similarly, we allow up to 5 resamplings when the generated code causes an error. The SFT model is trained with 8 A100 GPU for about 4 hours. The RLHF models are trained with 8 A100 GPU for about 18 hours. Detailed hyper-parameters are in Table 1. The only hyper-parameter we tune is λ for fine-grained RLHF. We experiment with 0.8, 0.9 and 1.0 and discover that 1.0 works the best.

	SFT	RL
learning rate	1e-5	2e-6
gradient accumulation	4	4
total steps	600	200
λ	-	1.0
γ	-	1.0

Table 1: **Hyperparameters.**

E Qualitative Examples

We show final answers generated by SFT and RLHF in Figure 1. RLHF better focuses on user query, while SFT tends to display generic statistics that are less relevant to user query.

We show examples of code generations in Figure 2. We also report their reward scores from contribution RM and regularization RM.

As a game developer, I want to balance the stats of different Pokemon to ensure fair gameplay.

SFT

Findings
1. There are 8 attributes (height, weight, hp, attack, defense, sp atk, sp def, speed) in the dataset that have strong positive or negative correlations, indicating that certain attributes are related.
2. The pairs of attributes with strong positive or negative correlations are as follows:
   - height and weight (correlation: 0.65)
   - hp and attack (correlation: 0.45)
   - hp and defense (correlation: 0.44)
   - hp and speed (correlation: 0.34)
3. The pairs of attributes with strong negative correlation include:
   - sp atk and defense (correlation: -0.6)
   - sp def and speed (correlation: -0.5)

Suggestions
1. For balance purposes, you could consider assigning a higher weight to taller Pokemon and a higher percentage of the total weight for Pokemon, as height and weight are strongly correlated with each other.
2. To ensure that the gameplay remains diverse and fair, you should also assign appropriate attack, defense, sp atk, sp def, and speed values to the Pokemon, taking into consideration their correlations with the other attributes.
3. Since the Pokemon attributes are strongly correlated, consider implementing a more complex system that takes advantage of this relationship, such as a network or genetic algorithm-based approach to balance the attributes of diverse Pokemon species.

RLHF

Findings
1. The majority of Pokemon are Water (38.13%), followed by Normal (24.72%) and Grass (13.82%).
2. Fire, Psychic, and Steel types are relatively less frequent in the dataset.
3. Types with high positive correlations with main stats (HP, attack, defense, and speed) include Water, Normal, and Grass types, while types with high negative correlations include Fire, Psychic, and Steel types.

Suggestions
1. Consider balancing the game around the most common types (Water, Normal, and Grass) to make the gameplay more diverse and interesting for different types of strategies.
2. Since types with high positive correlations have higher HPs, attacks, and speeds, it may be beneficial to focus on the Pokemon with those stats to balance the game accordingly.
3. Monitor the performance of high-positive correlation types (Water, Normal, and Grass) to ensure their representation in the game doesn't lead to an imbalance in terms of stats or gameplay strategy.
4. Monitor the performance of high-negative correlation types (Fire, Psychic, and Steel) to ensure their representation in the game isn't detrimental due to a potential imbalance in game mechanics.

Figure 1: **Case study.**

F GPT Prompts

Here we show the prompts we use for ChatGPT and GPT-4. Prompt for query generation is in Table 2. Prompt for helpfulness annotation collection is Table 3. Prompt for helpfulness evaluation is Table 4.

References

[1] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[2] Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Zihan Wang, Lei Shen, Andi Wang, Yang Li, Teng Su, Zhilin Yang, and Jie Tang. Codegeex: A pre-trained model for code generation with multilingual evaluations on humaneval-x. *CoRR*, abs/2303.17568, 2023.

I will examine the relationships from the correlation matrix, including the most attended concerts and the conductor's experience, as well as the orchestras with the highest official ratings and their conductors.

```
# Get the top 3 most attended concerts
top_attended = merged_df.nlargest(3, "Attendance")

# Get conductor names and age with most attended concerts
top_conductors = top_attended[["Name", "Age", "Year_of_Work", "Attendance"]].sort_values("Attendance",
ascending=False)

print("Top 3 Conductors with Most Attended Concerts:")
print(top_conductors)
print()

# Get top 3 orchestras with the highest official ratings
top_rated_orchestras = merged_df.nlargest(3, "Official_ratings_(millions)")

# Get conductor names and orchestras with the highest official ratings
top_rated_conductors = top_rated_orchestras[["Orchestra", "Name", "Age", "Year_of_Work",
"Official_ratings_(millions)"]].sort_values("Official_ratings_(millions)", ascending=False)

print("Top 3 Orchestras with Highest Official Ratings and Their Conductors:")
print(top_rated_conductors)
print()
```

Contribution RM: 5.05 Regularization RM: 5.15 Two RMs combined: 5.10

(a) A good case that receives high scores from both contribution RM and regularization RM.

Next, I will join the tables 'Order_Deliveries', 'Actual_Orders', 'Products', and 'Customers' to create a single DataFrame with the necessary information to analyze the most popular products across delivery routes.

```
# Join tables
orders_deliveries = db["Order_Deliveries"]
actual_orders = db["Actual_Orders"]
products = db["Products"]
customers = db["Customers"]

# Merge tables
merged_df = pd.merge(orders_deliveries, actual_orders, on="actual_order_id")
merged_df = pd.merge(merged_df, products, left_on="actual_order_id", right_on="product_id")
merged_df = pd.merge(merged_df, customers, left_on="driver_employee_id", right_on="customer_id")

# Display a sample of 5 rows
print(merged_df.head())
```

Contribution RM: -2.63 Regularization RM: 6.59 Two RMs combined: 1.98

(b) A bad case that receives low score from contribution RM and high score from regularization RM.

Based on the analysis, provide a conclusion summarizing the findings and suggesting possible reasons for the high resettlement rates in the top 5 countries.

```
print("Conclusion:")
print("The average resettlement rate in the top 5 countries is 593.8%, which is quite high compared to the
average resettlement rate of 553.8% in the entire dataset.")
print("This is due to the high resettlement rates in the top 5 countries, which may be due to the following
reasons:")
print("1. The top 5 countries have a higher population, which may indicate a higher demand for migration.")
print("2. The top 5 countries have specialized in Industry or Commerce, which might require a high number of
workers to be migrated.")
print("3. The top 5 countries have a significant history of migration, which may indicate a diverse population
and a need for migration.")
```

Contribution RM: 8.35 Regularization RM: -5.87 Two RMs combined: 1.24

(c) A reward hacking case that receives high score from contribution RM and low score from regularization RM.

Figure 2: **Qualitative examples** of code generations, and their scores assigned by reward models.

I have a database of [database title]. I am a stakeholder and I am analyzing the database to make a decision. Who am I and what decision might it be? List 10 possibilities in a numbered list.

Each point should introduce who I am and briefly explain my intention in this format: As a/the [who I am], I want to [explain my intention]

Examples:

Based on the extracurricular activities database:
1. As the dean of student affairs, I want to decide on extracurricular activities to promote or cut
2. As the department head, I want to decide on faculty advisor assignments
3. As the school administrator, I want to review and revise faculty activity engagement

Based on a diabetes database:
1. As a healthcare policy maker, I want to decide on healthcare resource allocation
2. As a NIH official, I want to decide on medical research funding
3. As a health insurance actuary, I want to improve health insurance pricing strategy
4. As a health provider, I want to decide on patient care and treatment

Based on an allergy database:
1. As a catering manager, I want to plan meal options
2. As the school principal, I want to plan allergy awareness programs
3. As an administrator in the Student Affairs or Housing department, I want to decide on housing assignments
4. As the school administrator, I want to improve campus emergency preparedness
5. As the school principal, I want to develop policies for allergy accommodations

Based on a Home Equity Line of Credit (HELOC) product database, you can:
1. As the credit risk manager, I want to modify the credit underwriting policy

The database is as follows:

Database `[title]` has [x] tables. Table names are: [aaa], [bbb], [ccc]

Table `[caption]` has [x] rows and [y] columns. Column are:
`[column name]`, example values: [value 1], [value 2], [value 3], [value 4], [value 5]
...

Table 2: Prompt for **query collection**.

I have a database of [database title]. As a [stakeholder role], I want to [describe intention].

Given below two findings/conclusions, which one is more helpful to my analysis?
* [answer bullet point 1]
* [answer bullet point 2]

Your response should be in the following format:
* Reasoning: <explain your reasoning here>
* Answer: <repeat the more helpful finding here>

Table 3: Prompt for **helpfulness annotation collection**.

I have a database of [database title]. As a [stakeholder role], I want to [describe intention].

I have hired two data analysts to perform the analysis, and they gave me two different reports (listed below). Each report consists of two lists, one for findings and one for suggestions. Which one is more helpful to my analysis? When evaluating helpfulness, you should consider the following three rubrics in decreasing priority: (1) relevance to my analysis goal; (2) insightfulness; and (3) diversity of perspectives, especially for suggestions.

Your response should be in the following format. Note: <answer> should be either Report-1 or Report-2
* Answer: <answer>
* Reasoning: <explain your reasoning here>

The reports are as follows:

Report-1

[report 1]

Report-2

[report 2]

Table 4: Prompt for **helpfulness evaluation**.