CoT-Planner: Chain-of-Thoughts as the Content Planner for Few-shot Table-to-Text Generation Reduces the Hallucinations from LLMs

Anonymous ACL submission

Abstract

Few-shot table-to-text generation seeks to generate natural language descriptions for the given table in low-resource scenarios. Previous works mostly utilized Pre-trained Language Models (PLMs) even Large Language Models (LLMs) to generate fluent descriptions of the tables. However, they are prone to hallucinations that do not conform to the table. In this work, we propose CoT-Planner, a simple but efficient Chain-of-Thoughts-based approach that can be used to reduce the generation of hallucinations in the few-shot table-to-text generation. We first use a large language model (such as ChatGPT) to automatically generate ten intermediate content plans in the form of a Chain-of-Thoughts (CoT) for each table and corresponding description pair. Then, we refined the most accurate content plan for each sample and used the table and text pairs with the added content plan (CoT-Plan) as demonstrations for In-Context Learning (ICL). Both automatic and human evaluations on the numericNLG dataset show our method can effectively alleviate hallucinations, thereby improving factual consistency in few-shot table-to-text generation. The code and data will be released upon acceptance.

1 Introduction

Table-to-text generation (Table2Text) is an important branch of Natural Language Generation (NLG), aiming at generating textual natural language descriptions that can fluently and precisely describe the given table. Table2Text has a wide variety of application scenarios, such as weather forecasting report (Liang et al., 2009), sport news generation (Wiseman et al., 2017), medical report generation (Nishino et al., 2020) and open-domain table-based question answering (Chen et al., 2020a, 2021; Jiang et al., 2022).

In recent years, supervised natural language generation models have shown the ability to generate natural language text at an astounding degree of fluency and coherence, due to the advent of pre-trained language models (PLMs) such as GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020). However, table-to-text generation faces the dilemma of lack of labeled data. In our daily lives, numerous statistical tables are produced, yet they lack nearly any corresponding descriptions in natural language. To address this concern, researchers are exploring alternative methods in the few-shot settings (Luo et al., 2022). Fortunately, large language models (LLMs; Zhao et al., 2023) that contain hundreds of billions (or more) of parameters, such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), Galactica (Taylor et al., 2022), and LLaMA (Touvron et al., 2023a), can solve few-shot tasks through in-context learning (ICL; Dong et al., 2023) which incorporates input-output demonstrations into the prompt. More recently, ChatGPT[1] and GPT-4 (OpenAI, 2023) benefit from instruction fine-tuning and perform well on new tasks even in the few-shot scenario.

Nevertheless, when presented with complex reasoning tasks, this simple ICL method usually performs poorly. As shown in Figure 1, in the table-to-text generation task involving numerical reasoning (Suadaa et al., 2021), the simple ICL method is prone to generating descriptions with various hallucinations: (1) the fact exists, but the numerical value is incorrect; (2) the fact does not exist, and the numerical value is also incorrect; (3) the numerical value is correct, but the fact does not exist.

Wei et al. (2022) find that chain-of-thought prompting—a technique that involves incorporating step-by-step reasoning processes into the demonstration—can enhance the reasoning ability of LLMs. Inspired by this discovery, in this paper, we propose **CoT-Planner**, a simple but efficient Chain-of-Thoughts-based few-shot table-to-text generation approach to reduce the hallucina-

[1] https://openai.com/blog/chatgpt/

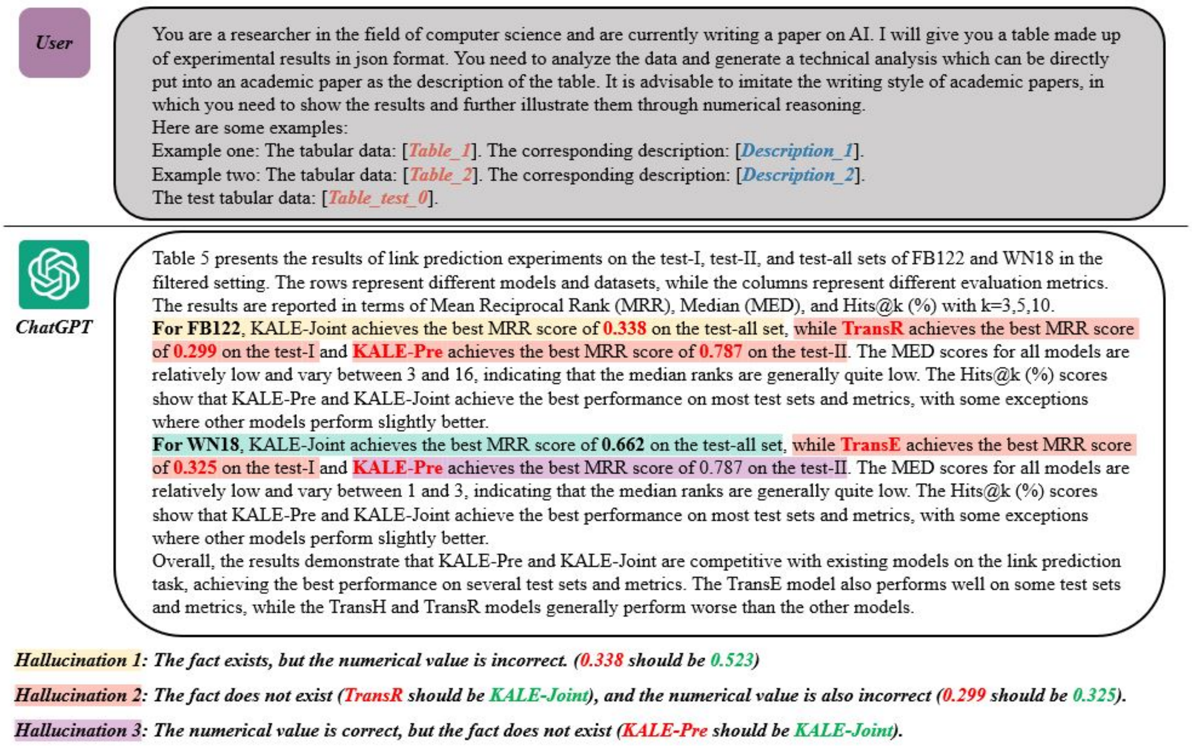

Figure 1: Hallucinations of ChatGPT in the table-to-text generation focusing on numerical reasoning.

tions from LLMs. Specifically, we first utilize LLMs to automatically generate the intermediate content plan in the form of the Chain-of-Thoughts and then introduce the content plan with the original corresponding input and output as the example of In-Context Learning for the few-shot table-to-text generation. Compared with traditional two-stage methods (Puduppully et al., 2019; Moryossef et al., 2019a,b; Su et al., 2021b; Luo et al., 2022), our method does not require fine-tuning of the two-stage model with content planning data, which is particularly suitable for low-resource scenarios. Furthermore, descriptions generated under the guidance of an intermediate CoT-Plan are more trustworthy and interpretable than descriptions produced using the typical ICL method. To evaluate the effectiveness of our approach, we conduct extensive experiments on a wide range of Large Language Models, such as ChatGPT, LLaMA-2(Touvron et al., 2023b), Alpaca(Taori et al., 2023), and Vicuna(Zheng et al., 2023). Our results reveal that LLMs can achieve remarkable performance with only 1 or 2 CoT-Plan demonstrations in the table-to-text generation task. Our human evaluation indicates that the CoT-Planner can effectively reduces the hallucinations generated by various LLMs in few-shot table-to-text generation.

2 Related Work

2.1 Few-shot Table-to-Text Generation.

Ma et al. (2019) firstly studied table-to-text generation under the low-resource constraint, and separated the generation process into two stages: key fact prediction and surface realization. Pre-trained language models (PLMs; Chen et al., 2020b) such as GPT-2, T5, and BART have performed well in various few-shot natural language generation (NLG) tasks in recent years (Li et al., 2021). However, adapting pre-trained language models to the table-to-text generation task requires serialization for structured data, resulting in the loss of its structured information. To preserve the table's structural information and improve the text's fidelity, Gong et al. (2020) exploited multi-task learning with two auxiliary tasks: table structure reconstruct from GPT-2's representation and the content matching based on the optimal transport distance. Su et al. (2021a) proposed the Prototype-to-Generate (P2G) framework, which utilized the retrieved prototypes to help the model bridge the structural gap between tables and texts. And Ke et al. (2022) introduced self-training to explicitly capture the relationship between structured data and texts. To generate a coherent and faithful sentence with high coverage of

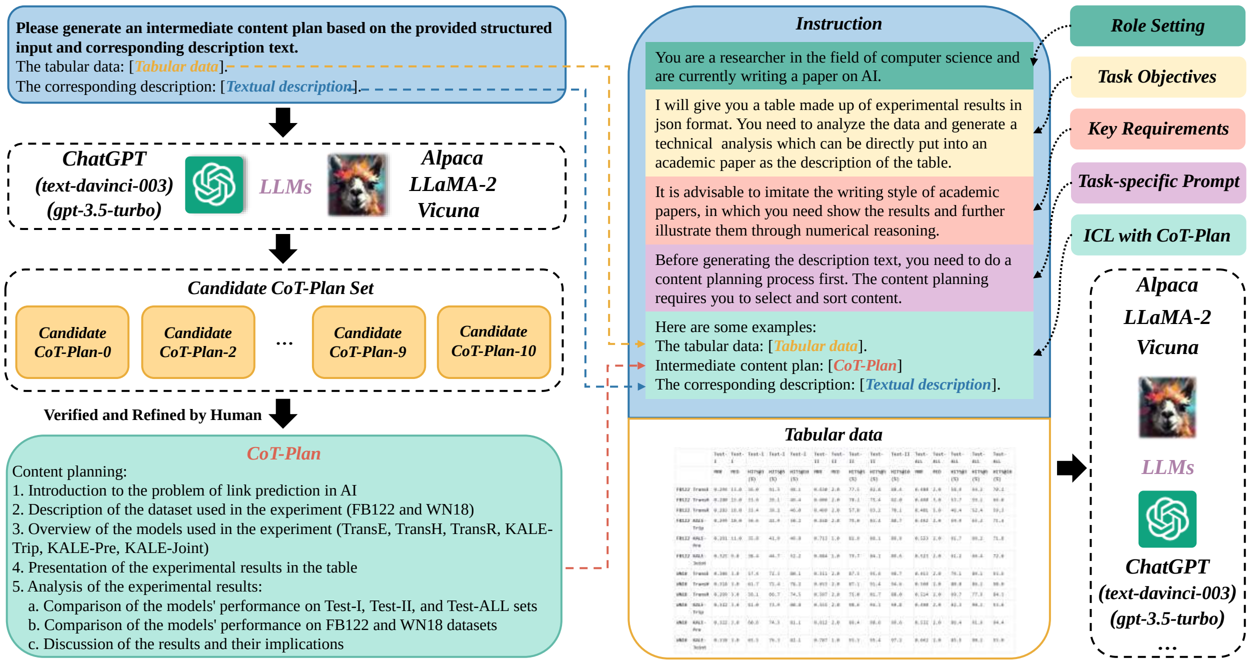

Figure 2: The overview of the proposed CoT-Planner approach. Left: Semi-automatic CoT-Plan; Right: In-Context Learning with CoT-Plan.

table slots, Zhao et al. (2021) proposed a table slot attention mechanism to empower the model generalization ability in inference and designed a memory unit to monitor the visits of each table slot. Li et al. (2023) introduced a unified representation for knowledge graphs, tables, and meaning representations, which led to significant improvements in transfer learning scenarios across structured forms in the few-shot settings. Inspired by prompt tuning that was first proposed by GPT-3, Luo et al. (2022) prepended a task-specific prefix for the PLMs to make the table structure better fit the pre-trained input. Jiang et al. (2023) developed an Iterative Reading-then-Reasoning (IRR) approach to support large language models (LLMs) in reading and reasoning on the structured data with the help of external interfaces. Different from the above studies, we focus on how to reduce the hallucinations from LLMs in few-shot table-to-text generation.

2.2 Chain-of-Thoughts Reasoning with LLMs.

While LLMs have shown remarkably effective in a range of NLP tasks, their capacity for reasoning is often seen as a drawback. Even worse, this capability cannot be gained simply by increasing the size of the model. It has recently been found that LLMs can do intricate reasoning over text when they are given the Chain-of-Thoughts prompting(Wei et al., 2022). CoT prompting allows the model to learn more precisely about the reasoning process and the complexities of the queries. And Wang et al. (2023) propose to use self-consistency with CoT to further improve performance. Besides, the Chain-of-Symbol (CoS; Hu et al., 2023) represents the complex environments with condensed symbolic chain representations during planning in symbolic reasoning. The original chain structure naturally limits the scope of exploration. Tree of Thoughts (ToT; Yao et al., 2023), a variant of CoT, allows LLMs to perform deliberate decision-making by considering multiple different reasoning paths and self-evaluating choices to decide the next course of action. Skeleton-of-Thought (SoT; Ning et al., 2023) is another variant of ToT, which decomposes a problem into subproblems that can be processed in parallel. Furthermore, Graph of Thoughts (GoT; Besta et al., 2023; Lei et al., 2023) additionally introduces aggregation and refinement operations compared to the ToT. However, current research does not delve into the ability of Chain-of-Thoughts prompting with LLMs to perform numerical reasoning on tables (Chen, 2023). In this paper, we are specifically interested in understanding LLMs' capability to reason over numerical tables with CoT-Planner, especially in data-to-text generation tasks.

3 CoT-Planner

In this section, we present the proposed CoT-Planner approach for the few-shot table-to-text generation task. Figure 2 depicts the overall architecture of our approach. As shown in the figure, the CoT-Planner framework consists of two subtasks: (1) Semi-automatic CoT-Plan and (2) In-Context Learning with CoT-Plan. We begin by showing in Section 3.1 how to semi-automatically generate the CoT-Plan (the content plan in the form of the Chain-of-Thoughts) in zero-shot scenarios. Next, in Section 3.2, we demonstrate the process of In-Context Learning with CoT-Plan for the few-shot table-to-text generation task.

3.1 Semi-automatic CoT-Plan.

Semi-automatic CoT-Plan integrates the advantages of both manual and automatic construction methods (Chu et al., 2023). Specifically, it first generates the corresponding CoT-Plan for each table-description pair directly using a large language model such as ChatGPT, as illustrated in Figure 2 (left). Inspired by zero-shot-CoT (Kojima et al., 2022), we implemented zero-shot content planning using just one simple prompt with the table-description pair. To ensure that the generated CoT-Plan is more reliable, we repeated the above operation ten times, thus forming a set of 10 candidate CoT-Plan for each example. The candidate CoT-Plan set is then verified and refined by human experts: (1) verifying the candidate CoT-Plan by comparing the factual consistency between each candidate CoT-Plan and the corresponding table; (2) refining the verified candidate CoT-Plan by removing redundant content and supplementing sentences with insufficient explanations. Each training example finally forms a high-quality CoT-Plan for subsequent In-Context Learning. The semi-automatic CoT-Plan reduces the workload of manual writing while introducing manual quality inspection to ensure the quality of CoT-Plan and enhance the reasoning ability and stability of LLMs.

3.2 In-Context Learning with CoT-Plan.

As shown in Figure 2 (right), for the Table2Text task, the input to the LLMs consists of 6 parts:

- **Role Setting (RS)**: You are a researcher in the field of computer science and are currently writing a paper on AI.
- **Task Objectives (TO)**: I will give you a table made up of experimental results in json format. You need to analyze the data and generate a technical analysis which can be directly put into an academic paper as the description of the table.
- **Key Requirements (KR)**: It is advisable to imitate the writing style of academic papers, in which you need to show the results and further illustrate them through numerical reasoning.
- **Task-specific Prompt (TSP)**: Before generating the description text, you need to do a content planning process first. This process requires you to select and sort content.
- **ICL with CoT-Plan**. Conventional ICL only incorporates input-output demonstrations into prompts. However, in our proposed method, the high-quality CoT-Plan generated by the first subtask is also integrated into the input-output demonstrations. Therefore, each demonstration has three components: input X (tabular data), CoT-Plan C_{Plan}, and output Y (textual description).
- **Tabular data**. This part is a test input for the few-shot table-to-text generation task. For complex tables with multiple rows and columns, the input data will be serialized into a long sequence. This helps to ensure that the large language model can effectively process and understand all of the information presented in the table, and generate accurate and coherent descriptions.

The basic instruction I_{RS} defines the role we want the LLM to play. The basic instruction I_{TO} defines the specific objectives we want the LLM to achieve for table-to-text generation tasks. The basic instruction I_{KR} further requires the large language model to follow a specified writing style and focus on numerical reasoning. Suppose there is a probabilistic language model p_{LM}.

In the conventional ICL scenario, the main objective is to maximize the likelihood of textual description $Y = (y_1, y_2, \cdots, y_{|Y|})$ given the input tabular data X and prompt T_{ICL}, as shown in Equ(1, 2).

$$p(Y|T_{ICL}, X) = \prod_{i=1}^{|Y|} p_{LM}(y_i|T_{ICL}, X, y_{<i}) \quad (1)$$

$$T_{ICL} = \{I_{RS}, I_{TO}, I_{KR}, (t_1, d_1), \cdots, (t_n, d_n)\} \quad (2)$$

where t_n and d_n represent the tabular data of the n-th sample in the demonstrations, respectively. And $|Y|$ represents the number of tokens of the textual description Y.

In the CoT-Planner scenario, where the prompt T_{Plan} contains the task-specific prompt I_{TSP} and the demonstrations contain the content planning process C_{Plan}, we need to maximize the likelihood of textual description Y and rationale $R = (r_1, r_2, \cdots, r_{|R|})$, as shown in Equ(3, 4, 5, 6, 7).

$$p(Y|T_{Plan}, X) = p(Y|T_{Plan}, X, R) \cdot p(R|T_{Plan}, X) \quad (3)$$

$$p(R|T_{Plan}, X) = \prod_{i=1}^{|R|} p_{LM}(r_i|T_{Plan}, X, r_{<i}) \quad (4)$$

$$p(Y|T_{Plan}, X, R) = \prod_{j=1}^{|Y|} p_{LM}(y_j|T_{Plan}, X, R, y_{<j}) \quad (5)$$

$$T_{Plan} = \{I_{Plan}, (t_1, c_1, d_1), \cdots, (t_n, c_n, d_n)\} \quad (6)$$

$$I_{Plan} = \{I_{RS}, I_{TO}, I_{KR}, I_{TSP}\} \quad (7)$$

where c_n represents the CoT-Plan (C_{Plan}) of the n-th sample in the demonstrations, and $|R|$ represents the number of tokens of the rationale R.

4 Experimental Results

4.1 Experimental Settings.

Here, we introduce the dataset, evaluation metrics, and baselines used in our experiment.

4.1.1 Dataset.

NumericNLG Dataset The numericNLG dataset was released by Suadaa et al. (2021). The split settings for training, validation, and testing were 1084:136:135 for the numericNLG dataset. Most of the table content in this dataset is numerical because it shows the experimental results from the scientific papers. We use this dataset to evaluate the accuracy and factual consistency of the descriptions generated for tables with numerical content. Specifically, <table_id> serves as the table's identifier, and <caption> is the table's brief headline for each numericNLG table. Additionally, there are various views of a cell for each table cell, including <metric>, <header>, and <value> for each row and column. The difficulty of this dataset lies in the need for numerical reasoning.

4.1.2 Automatic Evaluation Metrics.

We evaluate the generated description text from the following three aspects:

(1) We first assessed the informativeness of the generated texts using BLEU(Papineni et al., 2002), METEOR(Lavie and Agarwal, 2007), and ROUGE-L(Lin, 2004).

(2) We second computed the BERTScore(Zhang et al., 2020) to evaluate the semantic similarity between the generated texts and the ground-truth table descriptions using contextualized token embeddings of pre-trained BERT(Devlin et al., 2019).

(3) The unfaithful generation usually contains hallucinated content that can not be aligned to any input structured data, especially in table-to-text generation. Thus, considering both the reference text and table content, we also use the PARENT (Dhingra et al., 2019) metric to evaluate the faithfulness of the generated text to the input table.

4.1.3 Baselines.

In these experiments, we mainly take into account the following baseline models.

(1) Non-pre-trained Models

Template-based Generator. Following previous methods Suadaa et al. (2021), we also use a domain-specific template-based generator to generate two types of sentences in table descriptions: table referring sentences and data description sentences.

Pointer-Generator. Pointer-Generator (See et al., 2017) is a seq2seq model with the attention and copy mechanism. This model handles the out-of-vocabulary problem in data-to-text generation by combining copying from source text and generating from a vocabulary. We take table serialization as input for the pointer-generator model.

(2) Pre-trained Language Models (PLMs)

Fine-tuned GPT-2. GPT-2 (Radford et al., 2019) is a pre-trained language model with a decoder-only transformer architecture. In the fine-tuning stage, we concatenate the serialized table T_S and corresponding description text Y to train the language modeling of the pre-trained model. In the inference phase, we used only the serialized table T_S as the input to generate description text Y starting after the last token of the T_S.

TableGPT. To simultaneously improve text fidelity and leverage structural information, TableGPT (Gong et al., 2020) utilizes a multi-task learning paradigm that consists of two auxiliary tasks: one task aligns the tables and the information

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	BERTS	PARENT
Template-based Generator	10.28	5.52	2.83	1.14	11.31	11.49	86.88	17.15
Pointer-Generator	5.10	2.71	1.16	0.56	7.82	15.21	76.38	1.40
Fine-tuned GPT-2	16.13	9.02	4.68	2.20	10.14	17.48	85.12	6.56
TableGPT	18.69	8.21	3.31	1.51	11.06	16.90	-	-
TASD	21.81	11.03	4.92	2.15	11.87	20.40	-	-
Text-davinci-003	21.53	10.62	5.21	2.52	22.23	20.56	84.70	**17.21**
- with TSP	21.58	10.51	5.16	2.51	21.62	20.31	84.48	16.74
- with 1-shot ICL	23.89	11.94	**5.93**	**2.94**	22.76	**22.09**	85.71	15.29
- with TSP+1-shot CoT-Plan	**24.15**	**11.97**	5.90	2.79	**23.60**	21.45	**85.72**	13.67
GPT-3.5-turbo-16k	15.45	7.46	3.41	1.36	22.90	15.85	83.16	13.46
- with TSP	15.78	7.62	3.63	1.40	23.10	16.28	83.51	12.26
- with 1-shot ICL	15.79	7.58	3.60	1.47	23.11	15.89	83.56	**13.59**
- with TSP+1-shot CoT-Plan	**17.64**	**8.30**	**3.94**	**1.57**	**23.16**	**17.15**	**84.11**	13.05
LLaMA 2	13.73	4.31	1.31	0.37	15.15	13.01	82.96	4.67
- with TSP	12.84	4.11	1.25	0.44	15.24	12.28	82.68	5.07
- with 1-shot ICL	15.39	5.22	1.66	0.48	17.62	13.06	82.82	5.11
- with TSP+1-shot CoT-Plan	**17.76**	**6.44**	**2.15**	**0.52**	**19.52**	**14.62**	**84.12**	**5.47**
Alpaca-2	14.93	6.62	3.12	**1.28**	**22.69**	15.30	82.82	**13.46**
- with TSP	14.42	6.31	2.84	1.22	22.1	14.91	82.59	12.33
- with 1-shot ICL	14.59	5.53	1.81	0.59	19.30	13.14	82.13	6.85
- with TSP+1-shot CoT-Plan	**18.32**	**7.82**	**3.25**	1.23	20.70	**16.89**	**83.93**	8.26
Vicuna	7.76	3.62	1.63	0.72	15.8	12.32	80.78	7.73
- with TSP	7.80	3.53	1.51	0.73	15.37	12.19	80.56	6.59
- with 1-shot ICL	20.55	10.58	5.70	2.85	21.35	20.42	84.56	10.15
- with TSP+1-shot CoT-Plan	**21.20**	**11.13**	**6.13**	**3.12**	**21.60**	**21.23**	**84.89**	**12.47**

Table 1: Performance comparisons of the automatic evaluation on the numericNLG dataset. BERTS denotes BERTScore.

in the generated text, while the other reconstructs the table structure from representations of GPT-2.

TASD. TASD (Chen et al., 2022) first adopted a three-layered multi-head attention network to realize the table-structure-aware text generation model with the help of the pre-trained language model. Furthermore, a multi-pass decoder framework is adopted to enhance the capability of polishing generated text for table descriptions.

(3) Large Language Models (LLMs)

This family of models contains tens or hundreds of billions of parameters. In this paper, we also add a baseline method that directly uses various LLMs (e.g. ChatGPT, LLaMA 2, Alpaca-2, and Vicuna) to accomplish the table-to-text generation task in a zero-shot manner. We use the same basic instructions (role setting, task objective, and key requirements) in our approach to implement this baseline method, to ensure that the only distinction between our approach and this baseline method is the use of a task-specific prompt (TSP) and some examples of In-Context Learning (with CoT-Plan).

4.1.4 Implementation Details.

Concerning ChatGPT, we tested two models, Text-davinci-003 and GPT-3.5-turbo-16k, respectively, for inference on the numericNLG dataset. Their parameters are all 175B, but the former has a context window of 4k, while the latter has a context window of 16k. We used a temperature of 0.5 without any frequency penalty and top-k truncation. About LLaMA 2, we mainly used the Llama2-13B-4k version with the top-1 setting. For Alpaca-2, we mainly tested the Chinese-Alpaca-2-13B-16k(Cui et al., 2023) model on the numericNLG dataset. For Vicuna, we mainly used the Vicuna-v1.5-13B-16k model (top-k = 10, top-p = 0.5, temperature = 0.2) to generate descriptions of tabular data.

4.2 Main Results and Analysis.

Table 1 presents the automatic evaluation results comparisons between CoT-Planner and other baselines on the numericNLG dataset. First, with the

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	BERTS	PARENT
GPT-3.5-turbo-16k	15.45	7.46	3.41	1.36	22.90	15.85	83.16	13.46
- with TSP	15.78	7.62	3.63	1.40	23.10	16.28	83.51	12.26
- with 1-shot ICL	15.79	7.58	3.60	1.47	23.11	15.89	83.56	13.59
- with 1-shot CoT-Plan	14.08	6.66	3.00	1.19	22.72	14.92	83.22	11.72
- with TSP+1-shot CoT-Plan	**17.64**	**8.30**	**3.94**	1.57	23.16	**17.15**	**84.11**	13.05
- with 2-shot ICL	16.62	7.95	3.77	1.44	23.5	16.65	83.79	13.53
- with TSP+2-shot ICL	16.26	7.75	3.61	1.50	23.23	16.63	83.77	12.76
- with TSP+2-shot CoT-Plan	17.43	8.16	3.87	**1.63**	**23.26**	17.11	83.97	**14.14**
Alpaca-2	14.93	6.62	3.12	**1.28**	22.69	15.30	82.82	**13.46**
- with TSP	14.42	6.31	2.84	1.22	22.1	14.91	82.59	12.33
- with 1-shot ICL	14.59	5.53	1.81	0.59	19.30	13.14	82.13	6.85
- with 1-shot CoT-Plan	17.8	7.73	3.25	1.26	21.05	16.89	**84.04**	9.18
- with TSP+1-shot CoT-Plan	**18.32**	**7.82**	**3.25**	1.23	20.70	**16.89**	83.93	8.26
- with 2-shot ICL	14.12	6.35	2.86	1.09	**22.84**	12.86	82.85	6.03
- with TSP+2-shot ICL	12.75	5.50	2.41	0.90	20.89	12.37	81.60	6.74
- with TSP+2-shot CoT-Plan	12.53	4.80	1.38	0.32	16.47	13.18	82.18	4.04
Vicuna	7.76	3.62	1.63	0.72	15.8	12.32	80.78	7.73
- with TSP	7.80	3.53	1.51	0.73	15.37	12.19	80.56	6.59
- with 1-shot ICL	20.55	10.58	5.70	2.85	21.35	20.42	84.56	10.15
- with 1-shot CoT-Plan	19.94	10.56	5.83	2.94	21.25	20.97	84.86	10.38
- with TSP+1-shot CoT-Plan	**21.20**	**11.13**	**6.13**	**3.12**	**21.60**	21.23	84.89	**12.47**
- with 2-shot ICL	13.73	6.75	3.53	1.70	20.05	16.01	80.64	8.34
- with TSP+2-shot ICL	13.77	6.87	3.50	1.66	19.9	16.13	80.64	8.90
- with TSP+2-shot CoT-Plan	20.91	10.82	5.67	2.62	20.27	**22.36**	**85.43**	11.93

Table 2: Ablation experiments on the numericNLG dataset. BERTS denotes BERTScore.

basic instruction (role setting, task objectives, and key requirements) as the prompt, LLMs have the capability to directly generate fluent descriptions of the numerical tables, achieving comparable performance as full-data supervised-tuning methods, in a zero-shot setting without using any example. Second, our proposed method can significantly improve the performance of LLMs, especially GPT-3.5-turbo-16k, LLaMA 2, and Vicuna. It indicates the effectiveness of CoT-Planner in helping LLMs reasoning over numerical tables. However, the performance of Alpaca-2 with 1-shot ICL is worse than that of the zero-shot baseline method, indicating that Alpaca-2 has trouble comprehending examples of the data-to-text generation task. In PARENT, hallucinations make it difficult to measure the true faithfulness of the generated text to the input table based on their scores. Therefore, table1 shows that this metric exhibits different trends in different LLMs. Overall, LLMs with CoT-Planner are more effective than ordinary ICL methods, achieving new state-of-the-art performance on the numericNLG dataset in the few-shot scenario.

4.3 Ablation Study.

Moreover, to verify the effectiveness of different modules, we compare CoT-Planner with its variants on three models with the 16k context window since the 4k context window can only contain at most 1-shot example. Table 2 shows our ablation experimental results. We then analyze the following three questions:

(1) Is only TSP effective? As can be seen in Table 2, compared to the baseline method in a zero-shot setting, the method that only added TSP did not significantly improve the text generated by the LLMs and even deteriorated the performance of Vicuna and Alpaca-2. Moreover, the lack of examples of content planning in ICL makes it difficult for LLMs to comprehend TSP accurately, which leads to the generation of erroneous descriptions.

(2) Is only CoT-Plan effective?

Table 2 shows that the method with only 1-shot CoT-Plan is slightly inferior to the method with both TSP and 1-shot CoT-Plan added simultaneously. In conclusion, we can declare that the best option is to combine the CoT-Plan with TSP. The two complement each other in terms of definition and instance, which helps the LLMs better understand specific tasks.

(3) More examples are better?

From Table 2, we can see that the 2-shot CoT-Plan is generally less effective than the 1-shot CoT-Plan on LLMs with the 16k context window, especially on Alpaca-2 and Vicuna. Due to the average length of the CoT-Plan examples exceeding 3340 words, the understanding ability of the LLMs for contextual examples exceeding 2-shot has significantly decreased. To further explore this issue, we compared the results of GLM-4-9B-128k from 1-shot to 5-shot. As can be seen in Table 4, as the number of examples increases, the overall performance of GLM-4-9B shows an obvious increase.

4.4 Human Evaluation on Hallucinations.

To better assess the quality of generated descriptions for tables with numerical content, we conducted human evaluation experiments targeting three types of hallucinations on complex tables. Specifically, we selected 17% of the 59 samples with complex tables (at least 7 rows and 4 columns) in the test set. Then we separately counted the proportion of three types of hallucinations in each sample and used their arithmetic mean as the final result. As shown in Table 3 and Figure 4, our method (CoT-Planner) effectively reduces the hallucinations generated by various large language models, while ordinary ICL methods may even exacerbate the hallucination problem of large language models. From the results of H-1, it can be observed that our method makes the large language models more accurate in numerical reasoning, thereby generating descriptions with fewer numerical hallucinations. In addition, our method achieved the lowest proportion on H-2, indicating that it can at least accurately predict facts or values, especially on the GPT-3.5-turbo-16k model (H-2 = 0.00%).

4.5 Case Study.

In order to understand the effect of our method more intuitively, we select one representative example and present its descriptions generated by different methods with the GPT-3.5-turbo-16k model in Figure 3. Under the zero-shot setting, the model generates a description containing four H-1 hallucinations. The reason for these hallucinations is that the model confuses the results of the baseline method and the proposed method. In the conventional ICL scenario, the description generated by the model not only failed to solve the H-1 hallucination but also produced the more serious H-2 hallucination. However, in the CoT-Planner scenario, the description generated by the model does not contain any hallucinations. This demonstrates that our approach (CoT-Planner) effectively reduces hallucinations generated by LLMs, particularly in numerical reasoning over tables.

Method	H-1	H-2	H-3	Total
Text-davinci-003	13.61	3.58	8.25	25.44
- w/ 1-shot ICL	8.25	3.75	15.65	27.65
- w/ 1-shot CoT-Planner	2.50	2.92	6.17	**11.59**
GPT-3.5-turbo-16k	9.69	0.63	2.76	13.08
- w/ 1-shot ICL	6.25	3.28	5.59	15.12
- w/ 1-shot CoT-Planner	4.45	0.00	5.11	**9.56**
LLaMA 2	4.00	38.19	6.86	49.05
- w/ 1-shot ICL	9.57	45.00	1.25	55.82
- w/ 1-shot CoT-Planner	5.75	25.07	0.00	**30.82**
Alpaca-2	4.17	15.72	6.58	26.47
- w/ 1-shot ICL	3.76	15.98	16.68	36.42
- w/ 1-shot CoT-Planner	1.00	4.46	17.97	**23.43**
Vicuna	6.68	23.64	4.43	34.75
- w/ 1-shot ICL	7.00	22.00	5.00	34.00
- w/ 1-shot CoT-Planner	2.50	4.78	13.00	**20.28**

Table 3: Human Evaluation on Hallucinations. H-n denotes the proportion of Hallucination-n type (%). Besides, Total = H-1 + H-2 + H-3. CoT-Planner: TSP + CoT-Plan. The proposed method (LLMs with 1-shot CoT-Planner) achieved the best scores (bold).

5 Conclusion

In this work, we present CoT-Planner, a simple but efficient CoT-based approach that can be used to reduce the generation of hallucinations from LLMs in the few-shot table-to-text generation. In our approach, we first utilize LLMs to automatically generate the intermediate CoT-Plan in the form of a CoT and then introduce the CoT-Plan with the original corresponding input and output as the example of In-Context Learning for the few-shot table-to-text generation. To verify the effectiveness of our approach, we implement our approach on various LLMs. Experimental results on 5 LLMs show that our approach can effectively reduce the hallucinations from LLMs, thereby improving factual consistency in few-shot table-to-text generation. We also provide a thorough case study to highlight the strengths and weaknesses of different approaches to enlighten other researchers in related areas.

Limitations

Our approach has several limitations: (1) the contextual examples chosen are not necessarily the most appropriate and there is still a lot of room for improvement. (2) this method is still costly because it can only achieve good performance based on large language models. Therefore, we need to think about how to give similar reasoning powers to smaller models. (3) although we believe that content planning in the form of a chain structure is more suitable for table-to-text generation tasks, whether content planning in the form of trees or graphs is more effective requires further exploration.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. Graph of thoughts: Solving elaborate problems with large language models. *CoRR*, abs/2308.09687.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Miao Chen, Xinjiang Lu, Tong Xu, Yanyan Li, Jingbo Zhou, Dejing Dou, and Hui Xiong. 2022. Towards table-to-text generation with pretrained language model: A table structure understanding and text deliberating approach. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8199–8210. Association for Computational Linguistics.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1090–1100. Association for Computational Linguistics.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. Open question answering over tables and text. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020a. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7929–7942. Association for Computational Linguistics.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020b. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 183–190. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *CoRR*, abs/2309.15402.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur P. Parikh, Ming-Wei Chang, Dipanjan Das, and William W. Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4884–4895. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1978–1988. International Committee on Computational Linguistics.

Hanxu Hu, Hongyuan Lu, Huajian Zhang, Wai Lam, and Yue Zhang. 2023. Chain-of-symbol prompting elicits planning in large langauge models. *CoRR*, abs/2305.10276.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *CoRR*, abs/2305.09645.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 932–942. Association for Computational Linguistics.

Pei Ke, Haozhe Ji, Zhenyu Yang, Yi Huang, Junlan Feng, Xiaoyan Zhu, and Minlie Huang. 2022. Curriculum-based self-training makes better few-shot learners for data-to-text generation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4178–4184. ijcai.org.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation, WMT@ACL 2007, Prague, Czech Republic, June 23, 2007*, pages 228–231. Association for Computational Linguistics.

Bin Lei, Pei-Hung Lin, Chunhua Liao, and Caiwen Ding. 2023. Boosting logical reasoning in large language models through a new framework: The graph of thought. *CoRR*, abs/2308.08614.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Alexander Hanbo Li, Mingyue Shang, Evangelia Spiliopoulou, Jie Ma, Patrick Ng, Zhiguo Wang, Bonan Min, William Yang Wang, Kathleen R. McKeown, Vittorio Castelli, Dan Roth, and Bing Xiang. 2023. Few-shot data-to-text generation via unified representation and multi-source learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16171–16189. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. 2021. Few-shot knowledge graph-to-text generation with pretrained language models. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1558–1568. Association for Computational Linguistics.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 91–99. The Association for Computer Linguistics.

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

Yutao Luo, Menghua Lu, Gongshen Liu, and Shilin Wang. 2022. Few-shot table-to-text generation with prefix-controlled generator. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 6493–6504. International Committee on Computational Linguistics.

Shuming Ma, Pengcheng Yang, Tianyu Liu, Peng Li, Jie Zhou, and Xu Sun. 2019. Key fact as pivot: A two-stage model for low resource table-to-text generation. In *Proceedings of the 57th Conference of* the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2047–2057. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019a. Improving quality and efficiency in plan-based neural data-to-text generation. In *Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan, October 29 - November 1, 2019*, pages 377–382. Association for Computational Linguistics.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019b. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2267–2277. Association for Computational Linguistics.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *CoRR*, abs/2307.15337.

Toru Nishino, Ryota Ozaki, Yohei Momoki, Tomoki Taniguchi, Ryuji Kano, Norihisa Nakano, Yuki Tagawa, Motoki Taniguchi, Tomoko Ohkuma, and Keigo Nakamura. 2020. Reinforcement learning with imbalanced dataset for data-to-text medical report generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2223–2236. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6908–6915. AAAI Press.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Yixuan Su, Zaiqiao Meng, Simon Baker, and Nigel Collier. 2021a. Few-shot table-to-text generation with prototype memory. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 910–917. Association for Computational Linguistics.

Yixuan Su, David Vandyke, Sihui Wang, Yimai Fang, and Nigel Collier. 2021b. Plan-then-generate: Controlled data-to-text generation via planning. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 895–909. Association for Computational Linguistics.

Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards table-to-text generation with numerical reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1451–1465. Association for Computational Linguistics.

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. `https://github.com/tatsu-lab/stanford_alpaca`.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan

Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2253–2263. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *CoRR*, abs/2305.10601.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Wenting Zhao, Ye Liu, Yao Wan, and Philip S. Yu. 2021. Attend, memorize and generate: Towards faithful table-to-text generation in few shots. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 4106–4117. Association for Computational Linguistics.

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

A Case on numericNLG dataset

The representative example and its descriptions generated by different methods (zero-shot, 1-shot ICL, and 1-shot CoT-Planner) with the GPT-3.5-turbo-16k model are shown in Figure 3.

	Headline Generation			Key Phrase Generation			Classification
	R-1	R-2	R-L	R-1	R-2	R-L	Accuracy
Baseline (Pointer-Generator Network)	25.1	5.3	21.1	30.9	10.6	28.7	62.8
Proposed (MTL + SD + HCL)	**26.9**	**6.1**	**22.4**	**32.8**	**11.2**	**30.5**	64.4
Comparison of Decoder Information Sharing Method							
MTL + SD	26.3	6.0	21.8	32.3	10.4	29.9	63.9
MTL + SD + Cascade Model	26.3	5.6	21.6	31.8	10.6	29.5	64.4
MTL + SD + Cascade Model (Gold)	26.5	5.8	21.9	32.8	10.4	30.3	**64.5**
MTL + SD + Soft-Parameter Sharing	25.8	5.9	21.4	32.1	10.0	29.6	64.0
MTL + SD + Non-Hierarchical Consistency Loss	25.9	6.0	21.4	32.6	10.9	30.2	64.0
MTL + SD + HCL with Normalized Attention Weights	26.2	6.0	21.7	31.9	10.5	29.5	63.9
Comparison of Encoder Information Sharing Method							
HCL (SD and MTL are not applied)	25.8	5.6	21.2	31.0	10.1	28.7	63.1
SD + HCL (MTL is not applied)	25.6	5.6	21.5	31.2	10.2	28.9	62.6

[GPT-3.5-turbo-16k Zero-shot] Description:
Table 6 presents a comparison of different methods for decoder and encoder information sharing on the job advertisement dataset. The metrics used in this comparison are the same as those in Table 3. The proposed method, which adopts Hierarchical Consistency Loss (HCL), achieved the highest scores compared to the other sharing methods. The table consists of 10 rows and 7 columns. The row headers provide information about the different decoder and encoder information sharing methods being compared. The column headers indicate the task (Headline Generation, Key Phrase Generation, Classification) and the corresponding evaluation metric (R-1, R-2, R-L, Accuracy). The proposed method (MTL + SD + HCL) achieved the best performance across all tasks and metrics, as indicated by the bold scores in the table. For the task of Headline Generation, the proposed method achieved an R-1 score of **25.1**, R-2 score of **5.3**, and R-L score of **21.1**. In comparison, the baseline method (Pointer-Generator Network) achieved slightly lower scores of **26.9, 6.1, and 22.4**, respectively. Similarly, for Key Phrase Generation, the proposed method outperformed other sharing methods. It achieved an R-1 score of **30.9**, R-2 score of **10.6**, and R-L score of **28.7**, while the baseline method obtained scores of **32.8, 11.2,** and **30.5**, respectively. In terms of Classification accuracy, the proposed method achieved a score of **62.8**, while the baseline method obtained a slightly lower score of **64.4**. It is worth noting that the proposed method consistently outperformed other sharing methods across all tasks and metrics. This demonstrates the effectiveness of adopting Hierarchical Consistency Loss (HCL) in improving the performance of the model. Overall, the experimental results in Table 6 highlight the superiority of the proposed method (MTL + SD + HCL) in comparison to other decoder and encoder information sharing methods. The achieved scores demonstrate the effectiveness of the proposed approach in various natural language processing tasks, such as Headline Generation, Key Phrase Generation, and Classification.

Hallucination 1: For the task of Headline Generation, the proposed method achieved an R-1 score of 25.1 (26.9), R-2 score of 5.3 (6.1), and R-L score of 21.1 (22.4).

Hallucination 1: In comparison, the baseline method (Pointer-Generator Network) achieved slightly lower scores of 26.9 (25.1), 6.1 (5.3), and 22.4 (21.1), respectively.

Hallucination 1: It achieved an R-1 score of 30.9 (32.8), R-2 score of 10.6 (11.2), and R-L score of 28.7 (30.5), while the baseline method obtained scores of 32.8 (30.9), 11.2 (10.6), and 30.5 (28.7), respectively.

Hallucination 1: In terms of Classification accuracy, the proposed method achieved a score of 62.8 (64.4), while the baseline method obtained a slightly lower score of 64.4 (62.8).

[GPT-3.5-turbo-16k with one-shot ICL] Description:
Table 6 presents a comparison of different decoder and encoder information sharing methods for the job advertisement dataset. The metrics used in the evaluation are the same as those in Table 3. The proposed method, which adopts Hierarchical Consistency Loss (HCL) in addition to Multi-Task Learning (MTL) and Soft-Decoupled (SD) training, achieves the best scores compared to the other sharing methods. In terms of headline generation, the proposed method achieves an R-1 score of **25.1**, **which is slightly lower than the baseline Pointer-Generator Network's score of 26.9.** However, the proposed method outperforms the baseline in R-2 and R-L scores, achieving **6.1** and **22.4**, respectively, compared to the baseline's **5.3** and **21.1**. This indicates that the proposed method generates more accurate and coherent headlines. For key phrase generation, the proposed method consistently outperforms the baseline across all metrics. It achieves an R-1 score of **30.9**, compared to the baseline's **32.8**, and an R-2 score of **10.6**, compared to the baseline's **11.2**. The proposed method also achieves a **higher** R-L score of **28.7**, compared to the baseline's **30.5**. This demonstrates that the proposed method is more effective in generating key phrases. In terms of classification accuracy, the proposed method achieves a score of **62.8**, which is **higher** than the baseline's **64.4**. Although the difference is relatively small, it indicates that the proposed method performs well in classifying job advertisements. Comparing the different decoder information sharing methods, it can be observed that the proposed method consistently achieves better scores than the other methods. The inclusion of HCL, in addition to MTL and SD, contributes to the improved performance. This suggests that the hierarchical consistency loss helps in capturing more meaningful information during training. When comparing the encoder information sharing methods, the proposed method with HCL performs better than **the method with only SD**. This indicates that the inclusion of HCL enhances the encoder's ability to share information effectively. Overall, the results in Table 6 demonstrate that the proposed method, which adopts MTL, SD, and HCL, achieves the best scores across different metrics for headline generation, key phrase generation, and classification tasks. This highlights the effectiveness of the proposed approach in improving the performance of the job advertisement dataset.

Hallucination 1: It achieves an R-1 score of 30.9 (32.8), compared to the baseline's 32.8 (30.9), and an R-2 score of 10.6 (11.2), compared to the baseline's 11.2 (10.6).

Hallucination 1: The proposed method also achieves a higher R-L score of 28.7 (30.5), compared to the baseline's 30.5 (28.7).

Hallucination 1: In terms of classification accuracy, the proposed method achieves a score of 62.8 (64.4), which is higher than the baseline's 64.4 (62.8).

Hallucination 2: In terms of headline generation, the proposed method achieves an R-1 score of 25.1 (26.9), which is slightly lower (higher) than the baseline Pointer-Generator Network's score of 26.9 (25.1).

Hallucination 3: When comparing the encoder information sharing methods, the proposed method with HCL performs better than the method with only SD (not exist).

[GPT-3.5-turbo-16k with one-shot CoT-Planner] Description:
5.3 Overall Results. Table 6 presents a comparison of the decoder and encoder information sharing methods for the job advertisement dataset. The table includes various metrics, including R-1, R-2, R-L for headline generation and key phrase generation, as well as accuracy for classification. The proposed method, adopting MTL (Multi-Task Learning), SD (Soft-Decoupled), and HCL (Hierarchical Consistency Loss), achieved the best scores compared to the other sharing methods. This indicates that the proposed method outperforms other methods in capturing the semantic information and generating accurate results. When comparing the decoder information sharing methods, it can be observed that MTL + SD + HCL consistently outperforms other methods in all tasks. This suggests that combining multi-task learning, soft-decoupling, and hierarchical consistency loss leads to improved performance across different tasks. On the other hand, when comparing the encoder information sharing methods, HCL (without applying MTL and SD) achieves competitive results in most tasks. This implies that the hierarchical consistency loss alone can effectively capture and share information among different tasks. In conclusion, the proposed method, incorporating MTL, SD, and HCL, demonstrates superior performance compared to other sharing methods. The results highlight the importance of leveraging semantic information and utilizing hierarchical consistency in improving the performance of AI models. These findings have implications for the development of more effective and efficient AI systems in various domains.

There is no hallucination in this description.

Figure 3: A case on numericNLG dataset. Hallucination 1: The fact exists, but the numerical value is incorrect. Hallucination 2: The fact does not exist, and the numerical value is also incorrect. Hallucination 3: The numerical value is correct, but the fact does not exist. Top: tabular data of Test_{59}.

Settings	Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	BERTS	SUM
Zero-shot	GLM-4-9B-128k	13.09	6.11	2.84	1.00	22.88	14.38	82.81	143.11
	- w/ 1-shot ICL	16.16	7.14	2.86	1.02	22.56	15.54	83.70	148.98
	- w/ 2-shot ICL	17.52	7.97	3.37	1.18	22.91	16.76	83.89	153.60
Few-shot	- w/ 3-shot ICL	17.46	7.98	3.53	1.40	23.43	16.86	84.02	154.68
ICL	- w/ 4-shot ICL	17.88	8.22	3.58	1.30	**23.74**	17.16	84.19	156.07
	- w/ 5-shot ICL	**18.29**	**8.48**	**3.77**	**1.40**	23.73	**17.19**	**84.23**	**157.09**
	- w/ 1-shot CoT-Planner	16.96	7.99	3.64	1.34	22.93	16.56	83.78	153.20
	- w/ 2-shot CoT-Planner	17.84	8.21	3.60	1.42	23.12	17.10	84.24	155.53
Few-shot	- w/ 3-shot CoT-Planner	17.46	8.10	3.44	1.15	23.11	16.92	84.15	154.33
CoT-Planner	- w/ 4-shot CoT-Planner	17.86	8.40	3.71	1.28	**23.58**	17.22	84.14	156.19
	- w/ 5-shot CoT-Planner	**18.92**	**8.69**	**3.86**	**1.52**	22.99	**17.56**	**84.31**	**157.85**

Table 4: Experimental results on GLM-4-9B-128k model. BERTS denotes BERTScore. SUM denotes summation.

B Hallucinations in Human Evaluation

We have added a visualized figure to more intuitively observe the proportion of hallucinations on different LLMs in human evaluation experiments.

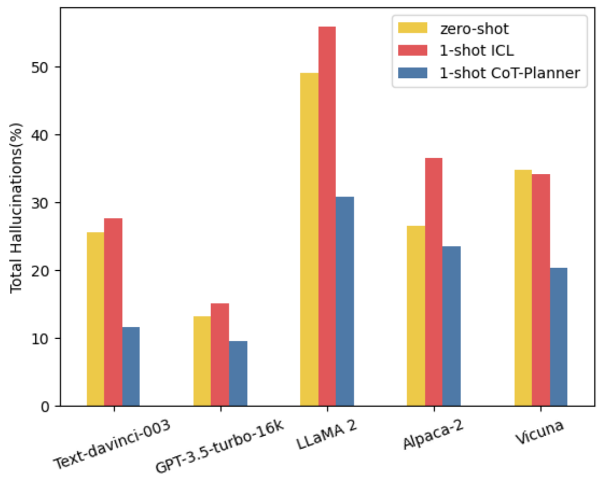

Figure 4: Total hallucinations of different LLMs in zero-shot and few-shot settings.

C Demonstration examples for various few-shot settings

We chose demonstration examples from the training set that satisfy both features: complex tables and accurate descriptions involving numerical reasoning. Specifically, in the 1-shot experiment, we used the 966th (containing 12 rows and 11 columns) sample from the training set as an example; in the 2-shot experiment, we used the 966th and 1009th (containing 11 rows and 4 columns) samples from the training set as examples. Similarly, we selected samples 966th, 1009th, 1040th, 1046th, and 1052nd as demonstration examples for the 5-shot experiment on the GLM-4-9B model.

D GLM-4-9B-128k Results

The results of GLM-4-9B-128k[2] from zero-shot to 5-shot in different settings are shown in Table 4. We use the sum of automatic evaluation metrics other than PARENT to represent the model's overall performance. We can more intuitively see the model's overall performance trend from Figure 5.

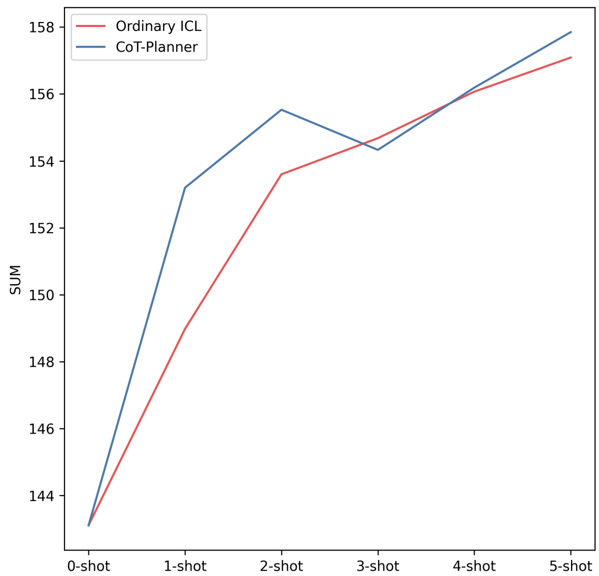

Figure 5: The overall performance (SUM) of GLM-4-9B-128k from zero-shot to 5-shot settings.

[2]https://huggingface.co/THUDM/glm-4-9b