GoT: Effective Graph-of-Thought Reasoning in Language Models

Yao Yao[1,2], **Zuchao Li**[3,*] **and Hai Zhao**[1,2,*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]National Engineering Research Center for Multimedia Software,
School of Computer Science, Wuhan University, Wuhan, 430072, P. R. China
`yaoyao27@sjtu.edu.cn`, `zcli-charlie@whu.edu.cn`,
`zhaohai@cs.sjtu.edu.cn`

Abstract

With the widespread use of language models (LMs) in NLP tasks, researchers have discovered the potential of Chain-of-thought (CoT) to assist LMs in accomplishing complex reasoning tasks by generating intermediate steps. However, human thought processes are often non-linear, rather than simply sequential chains of thoughts. Therefore, we propose Graph-of-Thought (GoT) reasoning, which models human thought processes not only as a chain but also as a graph. By representing thought units as nodes and connections between them as edges, our approach captures the non-sequential nature of human thinking and allows for a more realistic modeling of thought processes. GoT adopts a two-stage framework with an additional GoT encoder for thought graph representation and fuses the graph representation with the original input representation through a gated fusion mechanism. We evaluate GoT's performance on a text-only reasoning task (AQUA-RAT) and a multimodal reasoning task (ScienceQA). Our model achieves significant improvement over the strong CoT baseline on the AQUA-RAT test set and boosts accuracy from 85.19% to 87.59% using the T5-base model over the state-of-the-art Multimodal-CoT (Zhang et al., 2023) on the ScienceQA test set. Our code is publicly available at https://github.com/Zoeyyao27/Graph-of-Thought

1 Introduction

In the field of human cognition, it has long been recognized that the human thought process is far more complex and non-linear than could be captured by a simple, sequential chain of thoughts (Barsalou, 1999). Human thinking is often characterized by its ability to make sudden leaps and connections between seemingly unrelated ideas, which can lead to novel insights and solutions. This non-linear, jumping thought process is a hallmark of human creativity, reasoning, and problem-solving abilities. However, it also poses a significant challenge for cognitive modeling and understanding.

Recently, Large Language Models (LLMs) have been advancing at an unprecedented pace. With the emergence of breakthroughs such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), and GPT-4 (OpenAI, 2023), the field of natural language processing has entered a new era of possibilities. Recent studies (Wei et al., 2022a; Wang et al., 2022; Zhang et al., 2022) have shown that the reasoning ability of LLMs can be unlocked by Chain-of-Thought (CoT) prompting. CoT prompting involves a series of intermediate natural language rationales that lead to the final answer. In addition, Zhang et al. (2023) have introduced Multimodal-CoT, which combines both language and visual modalities to help surpass the limitations of textual information. More detailed related works can be found in Appendix A.

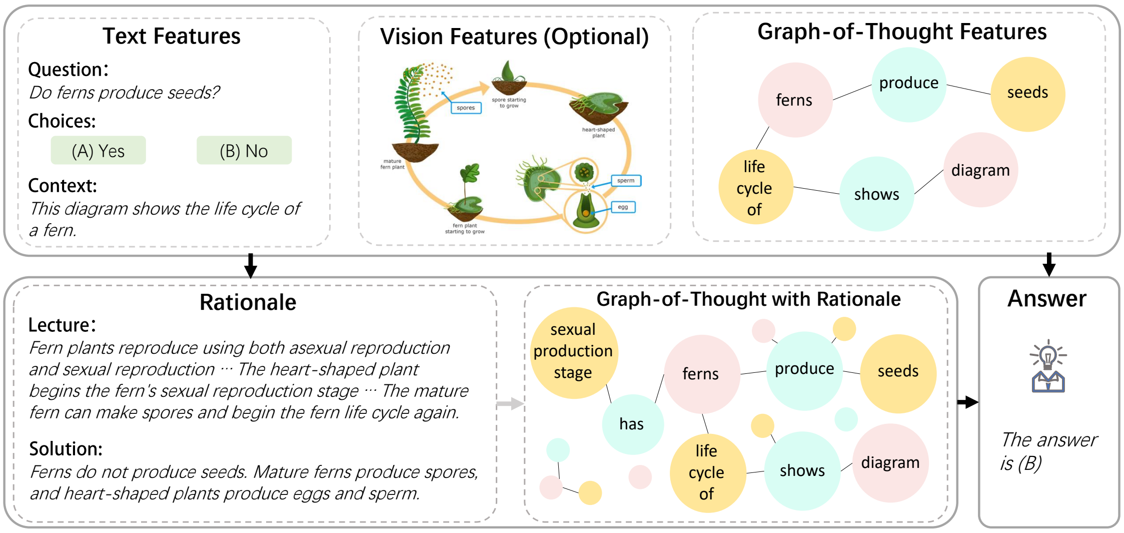
Previous works on Chain-of-Thought (CoT) prompting, which have been limited to textual and visual information, often represented the human reasoning process as sequential thought chains. This approach overlooks the modeling of humans' jumping thought process and neglects to incorporate the complex structural information of reasoning thoughts into the model. Concurrent work Tree-of-thoughts (ToT) (Yao et al., 2023) divides thoughts into thought units and models them as a tree-like search process.

Nevertheless, human cognition transcends this tree structure, exhibiting intricate graph-like formations. Our perspective diverges further as we believe that the human intellect is capable of crafting elaborate thought graphs founded upon linear

---

[*] Corresponding author. This research was supported by the National Natural Science Foundation of China (No. 62306216), the Natural Science Foundation of Hubei Province of China (No. 2023AFB816), the Fundamental Research Funds for the Central Universities (No. 2042023kf0133), the Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2901–2921
June 16-21, 2024 ©2024 Association for Computational Linguistics



Question: *Do ferns produce seeds?*

Context: *This diagram shows the life cycle of a fern.*

Lecture: *Fern plants reproduce using both asexual reproduction and sexual reproduction ··· The heart-shaped plant begins the fern's sexual reproduction stage ··· The mature fern can make spores and begin the fern life cycle again.*

Solution: *Ferns do not produce seeds. Mature ferns produce spores, and heart-shaped plants produce eggs and sperm.*

The answer is (B)

Figure 1: An example of GoT reasoning. Vision features are optional and are only required in multimodal reasoning.

thoughts. Therefore, we aim to enable the concurrent assimilation of linear and nonlinear cognitive processes, surpassing the mere generation of segmented thought units. To address the above limitation, different from ToT, we propose the Graph-of-Thought (GoT), a novel approach to modeling human thought processes not only as a chain but also as a graph. Our method is based on the assumption that the human mind works by connecting and recombining ideas in a non-sequential, graph fashion, rather than following a strict sequential chain. By representing thought units as nodes and connections between thoughts as edges, GoT captures the rich, non-sequential nature of human thinking and allows for a more realistic and logical modeling of reasoning processes.

An example of GoT reasoning is shown in Figure 1. Inspired by Multimodal-CoT (Zhang et al., 2023), we have adopted a two-stage reasoning framework. It first generates rationales and then generates the final answer based on the predicted rationales. In addition to text features, graph features of GoT are integrated during the rationale generation and answer inference. Specifically, GoT is first constructed with an Extract-Cluster-Coreference (ECC) process, which simulates the deductive process in human reasoning. We have used T5 (Raffel et al., 2020a) pre-trained language model as our backbone model. GoT is encoded with a graph attention network and then fused with the original representation via a gated fusion network.

Furthermore, we have also presented a multimodal GoT, which integrates not only text features and GoT features but also visual features. For our experiments, we have used both FLAN-Alpaca [1] (T5)-base and FLAN-Alpaca (T5)-large as our backbone models.

We implement GoT as a two-stage framework and fine-tuning language models and integrating text, thought graph, and vision features for a more realistic and accurate reasoning process. GoT demonstrates exceptional performance on both text-only AQUA-RAT (Ling et al., 2017) and multimodal ScienceQA (Lu et al., 2022) benchmarks, surpassing the accuracy of online system ChatGPT (OpenAI, 2023) by 9.28%, strong baseline Multimodal-CoT (Zhang et al., 2023) by 2.40%, and even exceeding human performance, establishing a new state-of-the-art on ScienceQA test set with far fewer parameters.

2 Graph-of-Thought

The overview of our proposed GoT can be seen in Figure 2. Inspired by Multimodal-CoT (Zhang et al., 2023), GoT also adopts a two-stage framework. (1) Rationale generation stage: In the first stage, the model generates rationales based on the input text (including question, context, and choices) the vision features, and the generated thought graph corresponding to the input text. For multi-modal tasks (Zhang et al., 2023; Zhang and Zhang, 2023; Huang et al., 2023; Peng et al., 2023), it is a common practice to use different encoders to process inputs from different modalities and a straightfor-

---

[1] https://github.com/declare-lab/flan-alpaca. FLAN-Alpaca is developed by fine-tuning T5 model on the Flan collection

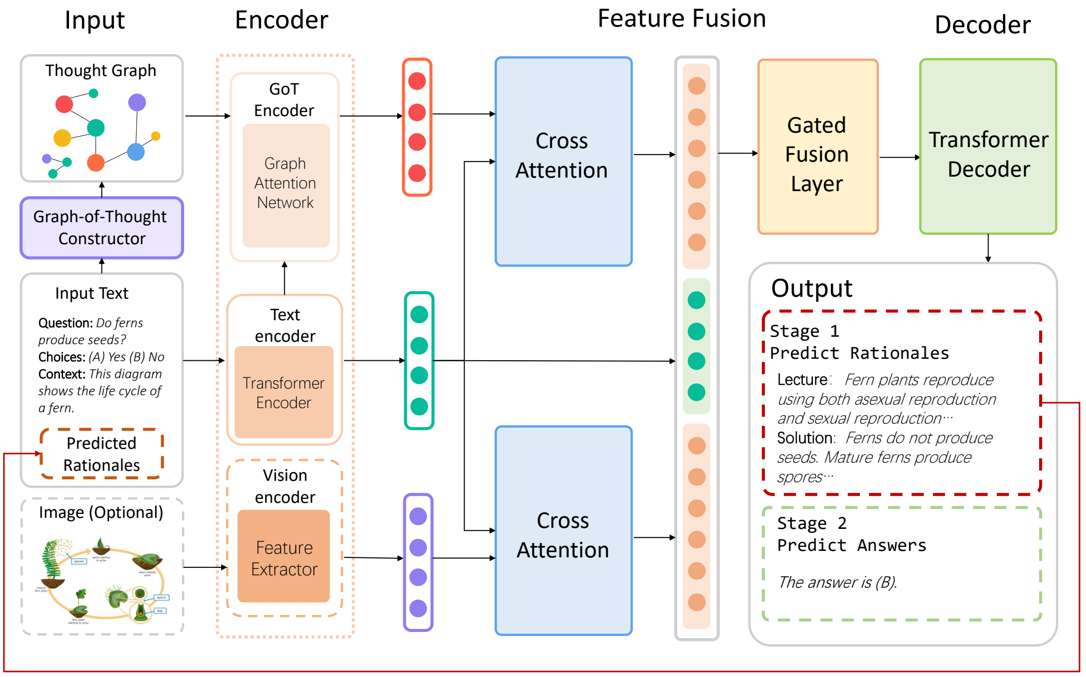


Figure 2: Graph-of-Thought framework overview

ward and versatile approach is to employ encoder-decoder models. Therefore, GoT employs independent encoders to encode input data for each modality. We use a Transformer encoder to encode input text, a vision encoder to encode an image, and a graph attention network to encode the thought graph. The encoded features are further passed into cross-attention to align text tokens with image patches and graph nodes, respectively. We then use a gated fusion layer to fuse these three features further and pass them into the Transformer decoder to predict the target rationales. (2) Answer generation stage: The second stage aims at generating the final answer and is largely similar to the first stage. The main difference is that the input text is concatenated with the predicted rationales from the first stage. It is worth noting that the above process describes a general multimodal reasoning framework. However, for text-only reasoning tasks, there are no image features, so the image encoding and vision feature fusion processes mentioned above can be omitted. In the following section, we will provide a detailed exposition of the two key steps of our GoT reasoning framework: GoT construction and GoT encoding and feature fusion.

2.1 GoT Construction

GoT employs thought graphs to simulate human deductive reasoning, thereby modeling humans' ability for leaps of thought. Our aim is to reflect the most fundamental deduction process by constructing a thought graph. If we have evidence that $x \rightarrow y$ and $y \rightarrow z$, then it follows that $x \rightarrow z$. In Figure 3, the deduction reasoning can be formulated as follows: $\textit{Earthquake} \xrightarrow{\textit{comes from}} \{\textit{earth, quake}\}$, $\{\textit{earth, quake}\} \xrightarrow{\textit{means}} \{\textit{ground, shake}\}$. It is easy to reason that $\textit{Earthquake} \longrightarrow \{\textit{ground, shake}\}$.

We propose a novel Extract-Clustering-Coreference (ECC) process to construct thought graphs. ECC first extracts deductive triplets $T = \{t^i = (t^i_x, t^i_y, t^i_z)\}$ as the discrete raw graph, where t^i_x, t^i_y, and t^i_z are thought units of the i-th triplet, and there exists an edge e^i_{xy} between t^i_x and t^i_y, and an edge e^i_{yz} between t^i_y and t^i_z. Then, ECC clusters the nodes that refer to the same mentions to conduct coreference resolution. Specifically, we replace every graph node that belongs to a coreference cluster with the most representative mention in the cluster. By adopting this technique, our model is better equipped with denser thought graphs and the ability for deductive reasoning. The detailed algorithm is illustrated in Algorithm 1.

In GoT construction, during the rationale generation stage, the input text consists of concatenated question, context, and choices. In multimodal GoT, image caption (Lu et al., 2022) is appended to the input text for GoT to incorporate image in-

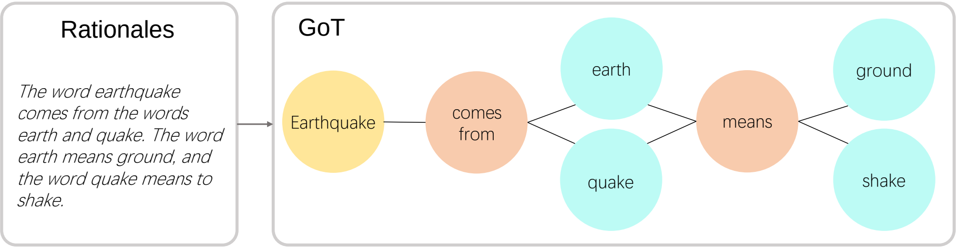

Rationales
The word earthquake comes from the words earth and quake. The word earth means ground, and the word quake means to shake.
GoT
Earthquake
comes from
earth
quake
means
ground
shake

Figure 3: Graph-of-Thought deduction example

formation. During the answer inference stage, the predicted rationales from the rationale generation stage are further concatenated with the input text for corresponding GoT construction.

In our implementation of ECC process, inspired by (Chen and Yang, 2021), we utilize open information extraction (OpenIE) systems [2] (Angeli et al., 2015) to extract subject-verb-object triplets as thought unit nodes. We apply coreference resolution to the extracted nodes using the Stanford CoreNLP system (Manning et al., 2014). The constructed thought graph is denoted as $\mathcal{G}(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ represents the nodes extracted by OpenIE and $\mathcal{E}$ represents the adjacency matrix. Rows and columns correspond to the nodes in the graph, and if there is an edge between two nodes, the corresponding matrix element is 1; otherwise, it is 0.

Algorithm 1 ECC process

```
Input: Input text $S$
Output: Thought graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$
  Extract deductive triplet set $T$ from $S$
  $T = \{t^0, t^1, ..., t^n\}, t^i = (t^i_x, t^i_y, t^i_z)$
  for every triplet $t^i \in T$ do
    $\mathcal{N}_r \leftarrow \mathcal{N}_r \cup \{t^i_x, t^i_y, t^i_z\}$
    $\mathcal{E}_r \leftarrow \mathcal{E}_r \cup \{e^i_{xy}, e^i_{yz}\}$
  end for
  extract coreference clusters $\mathcal{C}$ for $\mathcal{N}_r$
  for every node $n_i \in \mathcal{N}_r$ do
    if $n_i \in \forall c_j \in \mathcal{C}$ then
      $n^*_j \leftarrow$ most representative mention in $c_j$
      $\mathcal{N} \leftarrow \mathcal{N} \cup \{n^*_j\}$
    end if
  end for
  Reconnect $\mathcal{N}$ based on $\mathcal{E}_r$ to construct $\mathcal{E}$
  return $\mathcal{N}$ , $\mathcal{E}$
```

2.2 GoT Encoding and Integration

GoT reasoning utilizes separate encoders to encode input data for each modality. The thought graph is encoded using a graph attention network, while the input text is encoded using a Transformer encoder. In multimodal GoT reasoning, the image is encoded using an additional vision encoder.

2.2.1 Base Encoder

Text Encoder For text representation, we use the Transformer encoder (e.g. T5 (Raffel et al., 2020a)) to encode the input text. Given input sentence $S = \{w_0, ..., w_l\}$, we extract the hidden states from the last layer of the Transformer encoder to obtain the text representation H^T:

$$H^T = \{h_0, h_1, ..., h_l\} = \mathbf{Encoder}_{\text{text}}(S) \quad (1)$$

where h_i is the hidden representation of token i and l represents the length of the text input.

Vision Encoder (Optional) For multimodal reasoning with vision modality, following (Zhang et al., 2023), we extract patch-level features of image I using readily available vision extraction model as vision encoder $\mathbf{Encoder}_{vision}$ and then employ a trainable projection matrix $\mathbf{W_I}$ to project the extracted features into the vision representation H^I which have the same shape with H^T.

$$H^I = \mathbf{W_I}\mathbf{Encoder}_{\text{vision}}(I) \quad (2)$$

2.2.2 GoT Encoder

Node Embedding We first use special tokens `<s>` and `</s>` to highlight every thought graph node. Specifically, for node set with j nodes $\mathcal{N} = \{n_0, ...n_j\}$, we construct the node input as p and then feed the p into the same text encoder and utilize the output representation of the special token `<s>` as the initial node representation. Formally,

$$p = [\texttt{<s>}, n_0, \texttt{</s>}, ..., \texttt{<s>}, n_j, \texttt{</s>}] \quad (3)$$

---

[2] https://github.com/philipperemy/Stanford-OpenIE-Python

$$[h_0^s, h_0^n, h_0^e, ..., h_j^s, h_j^n, h_j^e] = \mathbf{Encoder}_{\text{text}}(p) \tag{4}$$

where the h_i^s and $h_i^e \in \mathbb{R}^D$ are the representation of `<s>` and `</s>` for node n_i respectively, D is the dimension of node embedding, and the $h_i^n = \{h_{i,1}^n, ..., h_{i,m}^n\}$ is the representations of node n_i with m tokens. we use the h_i^s to represent the node representation of n_i.

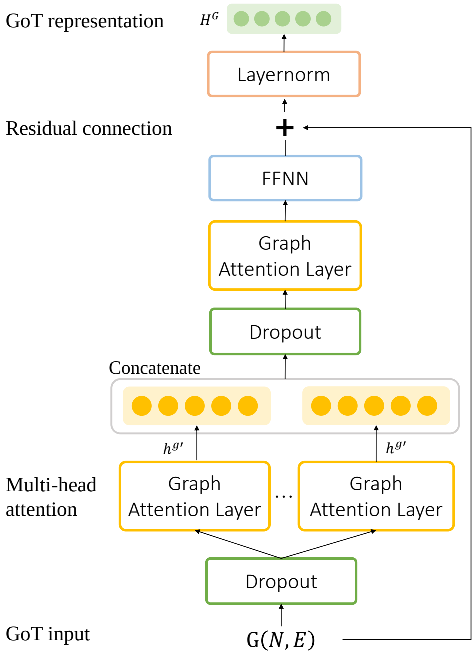


Figure 4: Architecture of GoT encoder

GAT Encoder We employ a graph attention network (GAT) (Velickovic et al., 2018; Chen and Yang, 2021) to encode the thought graph. For every node n_i in graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$, the **graph attention layer** is designed as:

$$a_{ij} = \mathbf{Attention}\left(\left[\mathbf{W}h_i^s || \mathbf{W}h_j^s\right]\right) \tag{5}$$

$$q_{ij} = \mathbf{LeakyReLU}\left(a_{ij}\right) \tag{6}$$

$$\alpha_{ij} = \mathbf{Softmax}(q_{ij}) = \frac{\exp\left(q_{ij}\right)}{\sum_{k\in\mathcal{K}_i}\exp\left(q_{ik}\right)} \tag{7}$$

$$h_i^{g'} = \mathbf{GELU}\left(\sum_{j\in\mathcal{K}_i}\alpha_{ij}\mathbf{W}h_j^s\right) \tag{8}$$

where $||$ denotes concatenate operation, the $\mathbf{W}$ is a trainable weight and the set $\mathcal{K}_i$ contains the node n_i's neighbours in thought graph $\mathcal{G}$. Our graph attention layer first employed a shared attention mechanism $\mathbf{Attention}(.) : \mathbb{R}^{D'} \times \mathbb{R}^{D'} \rightarrow \mathbb{R}$ to compute the attention weights, where D' is the attention layer output dimension. The attention weights a_{ij} measures the importance of node n_i's features to n_j's features. By only calculating the attention weights between nodes who are neighbours, our graph attention layer demonstrates the ability to perceive structural information of graphs. In our implementation, we adopt a single-layer feed-forward neural network (FFNN) as the attention mechanism which is both simple and straightforward.

Figure 4 shows the architecture of our GoT encoder. Our GoT encoder employs a multi-head graph attention layer, following (Velickovic et al., 2018), we concatenate the output of each graph attention layer and further pass it to a output graph attention layer with the same architecture:

$$h_i^{g'} = ||_{k=1}^{K}\mathbf{GELU}\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k h_j^s\right) \tag{9}$$

$$h_i^{g''} = \mathbf{GELU}\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}h_j^{g'}\right) \tag{10}$$

where K is the number of attention heads, $||$ is the concatenate operation, and n is the number of nodes in thought graph. We then use a single-layer feed-forward neural network (FFNN) to obtain the final thought graph embedding H^G:

$$h^{g''} = [h_0^{g''}, ..., h_n^{g''}]; \qquad H^G = \mathbf{FFNN}(h^{g''}) \tag{11}$$

2.3 Feature Fusion

After obtaining the encoded features, we use a single head attention to align the text representation H^T with image representation H^I and thought graph representation H^G, respectively. The image attention output $\mathbf{H^I}$ and thought graph attention output $\mathbf{H^G}$ are calculated by:

$$\mathbf{H^I} = \mathbf{Softmax}\left(\frac{H^T H^{I\top}}{\sqrt{d}}\right)H^I \tag{12}$$

$$\mathbf{H^G} = \mathbf{Softmax}\left(\frac{H^T H^{G\top}}{\sqrt{d}}\right)H^G \tag{13}$$

where Q is H^T and d is the dimension of H^T. We take both K_I and V_I as H^I and K_G and V_G as H^G. Please note that image representation is optional and is only required for multimodal dataset.

Next, a gated fusion mechanism (Wu et al., 2021; Zhang et al., 2023; Li et al., 2022; Zhang et al., 2020) is applied to combine the attention outputs $\mathbf{H^I}$ and $\mathbf{H^G}$ with the text representation H^T. The feature fusion output H can be calculated by:

$$\lambda = \begin{cases} \text{Sigmoid}\left(W_T H^T + W_G \mathbf{H^G}\right) \\ \qquad\qquad\qquad\qquad\qquad \text{text-only} \\ \text{Sigmoid}\left(W_T H^T + W_I \mathbf{H^I} + W_G \mathbf{H^G}\right) \\ \qquad\qquad\qquad\qquad\qquad \text{multimodal} \end{cases}$$

$$H = \begin{cases} (1-\lambda) \cdot H^T + \lambda \cdot \mathbf{H^G} \\ \qquad\qquad\qquad\qquad\qquad \text{text-only} \\ (1-\lambda) \cdot H^T + \lambda \cdot \mathbf{H^I} + \lambda \cdot \mathbf{H^G} \\ \qquad\qquad\qquad\qquad\qquad \text{multimodal} \end{cases}$$

where W_T,W_I and W_G are all trainable weights. We then input the fused feature output H into the decoder to predict the rationales or the final answer.

3 Experiments

Dataset We evaluate our model on the text-only AQUA-RAT (Ling et al., 2017) and multimodal ScienceQA benchmark (Lu et al., 2022). The detailed dataset information and statistics are shown in Appendix B.

Model Setup In our experiments, we used T5 (Raffel et al., 2020a) as our basic model architecture, including both T5-base and T5-large model sizes. Specifically, to ensure a fair comparison, we initialized our model with the finetuned T5 checkpoint FLAN-Alpaca [3] and used ViT-large encoder (Dosovitskiy et al., 2021) for the vision encoder, following (Zhang et al., 2023). We finetuned the models for 100 epochs with a learning rate of 5e-5. The detailed training parameters are available in Appendix C. We trained our models on four NVIDIA A800 80G GPUs.

4 Results and Discussion

4.1 Main Results

Baselines For AQUA-RAT, our baselines include: (1) Zero-Shot and Zero-Shot-CoT LLMs (Kojima et al., 2022); (2) Few-Shot and Manual-CoT LLMs (Wei et al., 2022b) and Auto-CoT (Zhang et al., 2022) (The above baselines all use the text-davinci-002 version of GPT-3 with 175B parameters); (3) Fintuned LLMs: Calcformer-T5-L (Kadlčík et al., 2023) which finetunes calculator-using T5-Large model on the Calc-X collection. To have a fair comparison we also fine-tuned FLAN-Alpaca$_{\text{base}}$ and FLAN-Alpaca$_{\text{large}}$ on AQUA-RAT with traditional two-stage CoT.

For ScienceQA, following (Zhang et al., 2023; Lu et al., 2022), our adopted baselines include: (1) Vision question answering (VQA) baseline models (Yu et al., 2019; Anderson et al., 2018; Kim et al., 2018; Gao et al., 2019; Kim et al., 2021; Lu et al., 2021; Li et al., 2019, 2020); (2) Text-to-text LLMs (Raffel et al., 2020b; Chen et al., 2020) and (3) Text-to-text LLMs with CoT prompting (Lu et al., 2022; Zhang et al., 2023). Both UnifiedQA (Lu et al., 2022) and GPT-3.5 (Lu et al., 2022) use generated image captions to incorporate vision semantics. Whereas, Mutimodal-CoT (Zhang et al., 2023) injects generated image features into traditional CoT reasoning.

MODELS	TRAINING	SIZE	ACC(%)
Zero-Shot (Kojima et al., 2022)	zero-shot	175B	22.40
Zero-Shot-CoT (Kojima et al., 2022)	zero-shot	175B	33.50
Few-Shot (Wei et al., 2022b)	few-shot	175B	24.80
Manual-CoT (Wei et al., 2022b)	few-shot	175B	35.80
Auto-CoT (Zhang et al., 2022)	few-shot	175B	36.50
Calcformer-T5-L (Kadlčík et al., 2023)	train-set	770M	27.20
FLAN-Alpaca$_{\text{base}}$	train-set	223M	30.09 ± 1.12
GoT-T5$_{\text{base}}$	train-set	223M	**32.09** ± 1.62
FLAN-Alpaca$_{\text{large}}$	train-set	738M	33.73 ± 1.14
GoT-T5$_{\text{large}}$	train-set	738M	**34.48** ± 1.11

Table 1: Main test accuracy results (ACC%) of AQUA-RAT. Size=backbone model size.

Results The rationales generation results can be seen in Table 8 in Appendix D. The overall results are reported in Table 1 and Table 2.

In the AQUA-RAT dataset, our GoT$_{\text{base}}$ model attains a 0.78 enhancement in ROUGE-L scores for rationale generation during the initial stage, outperforming the FLAN-Alpaca$_{\text{base}}$ model, which does not integrate GoT. For the answer generation phase, the GoT$_{\text{base}}$ exhibits a substantial accuracy increase of 2.00%, while the GoT$_{\text{large}}$ model records a 0.75% enhancement. Compared to the 175B parameter zero-shot and few-shot LLMs, our GoT-large, employing just a 738M backbone model, achieves results remarkably close to those of Manual-CoT (Wei et al., 2022b).

For ScienceQA dataset, in rationale generation stage, we can see from Table 8 that our model achieves a ROUGE-L of 94.39 and outperforms the Mutimodal-CoT$_{\text{base}}$ by 1.15. For the final an-

---

[3] https://huggingface.co/declare-lab/flan-alpaca-base

MODEL	TRAINING	SIZE	NAT	SOC	LAN	TXT	IMG	NO	G1-6	G7-12	AVG
Human	-	-	90.23	84.97	87.48	89.60	87.50	88.10	91.59	82.42	88.40
Vision question answering baselines											
MCAN (Yu et al., 2019)	train-set	95M	56.08	46.23	58.09	59.43	51.17	55.40	51.65	59.72	54.54
Top-Down (Anderson et al., 2018)	train-set	70M	59.50	54.33	61.82	62.90	54.88	59.79	57.27	62.16	59.02
BAN (Kim et al., 2018)	train-set	112M	60.88	46.57	66.64	62.61	52.60	65.51	56.83	63.94	59.37
DFAF (Gao et al., 2019)	train-set	74M	64.03	48.82	63.55	65.88	54.49	64.11	57.12	67.17	60.72
ViLT (Kim et al., 2021)	train-set	113M	60.48	63.89	60.27	63.20	61.38	57.00	60.72	61.90	61.14
Patch-TRM (Lu et al., 2021)	train-set	90M	65.19	46.79	65.55	66.96	55.28	64.95	58.04	67.50	61.42
VisualBERT (Li et al., 2019, 2020)	train-set	111M	59.33	69.18	61.18	62.71	62.17	58.54	62.96	59.92	61.87
Text-to-text LLMs											
UnifiedQA$_{\text{base}}$ (Raffel et al., 2020b)	zero-shot	223M	68.16	69.18	74.91	63.78	61.38	77.84	72.98	65.00	70.12
GPT-3.5 (Chen et al., 2020)	zero-shot	175B	74.64	69.74	76.00	74.44	67.28	77.42	76.80	68.89	73.97
Text-to-text LLMs with CoT											
UnifiedQA$_{\text{base}}$ (CoT) (Lu et al., 2022)	zero-shot	223M	71.00	76.04	78.91	66.42	66.53	81.81	77.06	68.82	74.11
GPT-3.5 (CoT) (Lu et al., 2022)	2-shot	175B	75.44	70.87	78.09	74.68	67.43	79.93	78.23	69.68	75.17
ChatGPT (CoT) (Lu et al., 2023)	few-shot	-	78.82	70.98	83.18	77.37	67.92	86.13	80.72	74.03	78.31
GPT-4 (CoT) (Lu et al., 2023)	few-shot	-	85.48	72.44	90.27	82.65	71.49	92.89	86.66	79.04	83.99
Mutimodal-CoT$_{\text{base}}$ (Zhang et al., 2023)	train-set	223M	84.37	88.30	84.36	83.72	80.32	86.90	85.83	84.05	85.19
GoT-T5$_{\text{base}}$	train-set	223M	86.25 ± 0.31	93.55 ± 0.06	85.51 ± 0.11	85.89 ± 0.32	86.30 ± 0.28	86.34 ± 0.12	87.79 ± 0.10	87.23 ± 0.40	**87.59** ± 0.20
Mutimodal-CoT$_{\text{large}}$ (Zhang et al., 2023)	train-set	738M	91.03	93.70	86.64	90.13	88.25	89.48	91.12	89.26	90.45
GoT-T5$_{\text{large}}$	train-set	738M	90.88 ± 0.22	93.57 ± 0.38	88.45 ± 0.44	90.26 ± 0.35	88.16 ± 0.25	90.29 ± 0.47	91.19 ± 0.16	90.14 ± 0.23	**90.81** ± 0.12

Table 2: Main test accuracy results (%) of ScienceQA. SIZE=backbone model size. Question classes: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context, G1-6 = grades 1-6, G7-12 = grades 7-12, AVG= average accuracy scores

swer generation stage, our GoT achieves SOTA in all subjects and all grades. The most direct comparison is that our model achieves an accuracy of 87.59% which is 2.40% higher than that of the Mutimodal-CoT$_{\text{base}}$ with the similar number of parameters.

GoT demonstrates a significant advantage over traditional CoT, elevating the accuracy from 30.09% to 32.09% in AQUA-RAT and from 85.19% to 87.59% in ScienceQA task. The results sufficiently suggest that utilizing thought graph features for deductive reasoning is a more effective approach than the existing methods, which only consider text or vision features by simply incorporating image captions or fusing generated image features. In conclusion, our results confirm the effectiveness of utilizing two-dimensional graph-of-thought and demonstrate the potential of incorporating GoT into reasoning for LMs.

4.2 Further Exploration

4.2.1 Ablation Study

AQUA-RAT In order to make sure that introducing thought graphs into GoT reasoning indeed boost the performance, we conduct the following experiments:

(1) **Random Thought Graph** In the Random Thought Graph experiment, we maintain the GoT framework while introducing randomness into the process. We construct a thought graph by randomly selecting nodes and arbitrarily establishing connections between them. This approach is designed to evaluate the extent to which the GoT reasoning mechanism is reliant on the structured organization of thought graphs. (2) **Triplets Concatenation** In the Triplets Concatenation experiment, we take a straightforward approach by appending the extracted triplets directly to the input text. This method aims to assess the impact of omitting the structural information typically provided by thought graphs, offering insight into the significance of this structural element in the reasoning process. (3) **Coreference Injection** In the Coreference Injection experiment, we explore the potential benefits of integrating coreference resolution directly into the language model's reasoning process. We achieve this by incorporating coreference information into the input text, where all instances of coreferent entities are replaced with a consistent phrase, followed by model fine-tuning. This experiment seeks to understand the role of coreference resolution in enhancing the model's deductive capabilities.

Table 3 shows the overall ablation results. From the table, we can see that by randomly construct thought graphs to disrupt the deductive reasoning process, our model suffers a loss of 1.78%, indicating the effectiveness of GoT. The results of Triplets

MODEL	MODEL SIZE	ACC	Δ
GoT-T5$_{\text{base}}$	233M	**32.09**	-
w/ Random Thought Graph		30.31	-1.78
Triplets Concatenation	233M	31.20	-0.89
Coreference Injection	233M	30.32	-1.77

Table 3: Ablation results of GoT on AQUA-RAT dataset.

Concatenation on the AQUA-RAT showed an accuracy of 31.20%. This performance gap of 0.89 clearly demonstrates the significance of the structural information in our approach. For Coreference Injection, the model suffers a loss of 1.77 % accuracy. We believe that these outcomes can be attributed to a couple of factors: (1) Simply replacing coreferent entities may lead to a loss of coherence in sentences, resulting in a reduction of semantic information and consequently having a limited impact on overall accuracy. (2) Open Information Extraction (OpenIE) for coreference resolution is not flawless, and direct replacement of entities might introduce noise that misleads the language model during judgment.

Contrastingly, the construction of a thought graph in the GoT framework does not compromise the original textual information (questions and rationales). Instead, it introduces additional structural assistance for LMs to conduct reasoning effectively. Thus, we contend that GoT's approach is indispensable and beneficial, as it supplements the LM's comprehension without introducing potential noise or loss of coherence in the input text.

ScienceQA To examine the impact of different backbone and vision encoder configurations on the GoT, we employed a distinct set of model settings. More specifically, we adopted the pre-trained T5 checkpoint UnifiedQA (Khashabi et al., 2020) as the backbone model and utilized DETR (Carion et al., 2020) for the vision encoder, with results illustrated in the Table 4. As shown, our GoT outperforms Mutimodal-CoT across various model configurations. A comparison reveals that GoT can achieve greater improvements on smaller models. We believe the main reason is that when the language model is not as robust, or when employing a relatively weaker vision encoder like DETR compared to ViT, GoT can leverage the inherent information within the language to enhance performance significantly. Additionally, to prove that our GoT's performance gain is not simply due to an increase in parameters, we conducted an ablation study. We expanded the parameter count of Multimodal-CoTbase to match our 233M model size by adding two layers of MLP instead of one in the gated fusion module, referred to as Multimodal-CoTbase(enlarged). We also constructed a random thought graph ablation study on the ScienceQA dataset. The results from the ablation studies can be observed in the table 4. From the table, it is evident that our model significantly outperforms the enlarged Multimodal-CoT by an accuracy of 2.04%. These findings convincingly demonstrate the significance of incorporating thought graphs into multimodal reasoning. The performance of GoT with a randomly constructed thought graph was even lower than Mutimodal-CoT, indicating that when the language model and vision encoder are weaker, the model relies more heavily on GoT for reasoning.

Model	ACC	Δ
UnifiedQA+DETR		
Mutimodal-CoT$_{\text{base}}$	77.67	-
Mutimodal-CoT$_{\text{large}}$	81.37	-
GoT$_{\text{base}}$	**81.21**	3.54
GoT$_{\text{large}}$	**82.74**	1.37
Ablation Studies		
Mutimodal-CoT$_{\text{base}}$(enlarged)	79.17	-2.04
GoT$_{\text{base}}$ w/ Random Thought Graph	76.74	-4.47

Table 4: Ablation results of GoT on ScienceQA dataset. For GoT models Δ indicates the performance gains of GoT models over their Multimodal-CoT counterparts. In the ablation studies, Δ represents improvements relative to the GoT$_{\text{base}}$ model

4.2.2 Analysis

Performance on Different Classes In order to investigate the impact of GoT on the overall model performance across different subjects , we calculated the accuracy for different subjects and compared it with that of Mutimodal-CoT. We also compare the performance of two models on different question classes.The radar Figure 5 shows the overall results for our base model. With respect to various subjects and question classes, our model demonstrates superior performance over the Mutimodal-CoT$_{\text{base}}$ and attains a more consistent and enhanced outcome. Our model presents outstanding advantages especially in the field of social science, with an accuracy improvement of 5.25%. For different question classes, our model demonstrates the largest improvement on questions involving images. Our hypothesis is that by constructing a thought graph and integrating the three features of text, image, and thought graph, we can better align the textual and visual information for the model, thus maximizing the utilization of visual information and obtaining more accurate answers.

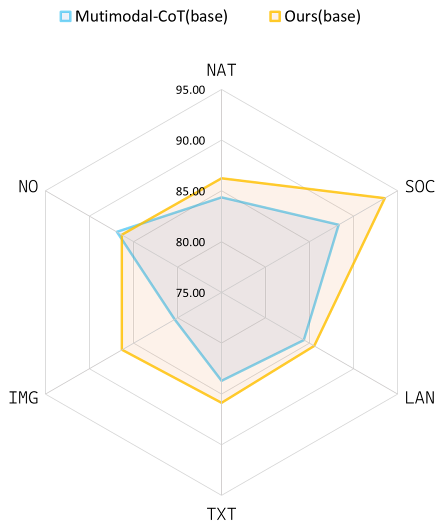


Figure 5: Performance on different question classes

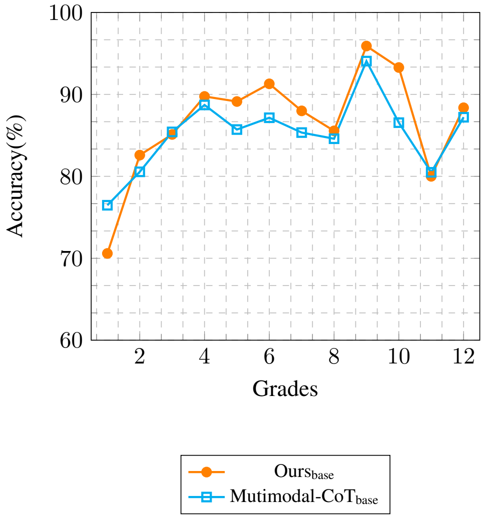


Figure 6: Performance on different grades

Performance on Different Grades It can be seen from the Table 2 that Mutimodal-CoT experience a decrease in accuracy of 1.78 as the grade level of the given question increases while GoT only has minor decrease of 0.56. We believe the main reason is that by incorporating GoT, models acquires the ability for deductive reasoning and can better comprehend the relationships between different entities and thus better understand the meaning of the problems. Through this method, for higher-grade problems with greater complexity, the model can construct a thought graph to help itself generate a more complete logical chain for deduction, thereby generating more accurate answers. More detailed model performance on different grades can be found in Figure 6. We can see that in the lower grade, two models achieves a similar performance. As the grade level increases and the difficulty of the questions becomes more challenging, the gap between our model and the Mutimodal-CoT model gradually widens. Due to the small number of questions ($\leq$ 130) available for each grade in grade 1 and grades 11-12, there is greater fluctuation in the accuracy of both models. Nevertheless, it is evident from the table that our model exhibits stronger and more stable advantages over Mutimodal-CoT in each grade.

Case Study and Limitation In order to gain a deeper understanding of the performance of GoT, we conduct case studies which can be found in the Appendix E. We also visualize the attention weights a_{ij} in GoT encoder to demonstrate how GoT performs deductive reasoning to generate more accurate answers in Appendix F. For the limitation of this work, compared to CoT, GoT may result in additional computational costs and slightly slower training times. Detailed limitation analysis can be found in Appendix G.

5 Conclusion

We introduce a novel Graph-of-Thought (GoT) reasoning approach, which is an innovative method for modeling the non-sequential nature of human thinking for LMs. GoT enhances LMs with deductive reasoning abilities, providing a more realistic representation of thought processes. Our experiments showcases the superiority of GoT on the text-only reasoning dataset AQUA-RAT, achieving a similar result compared to GPT-3 model while utilizing significantly fewer parameters. Furthermore, GoT establishes a new state-of-the-art on the multimodal reasoning benchmark, ScienceQA with fewer parameters. This performance surpasses strong ChatGPT and GPT-4 systems, as well as human performance, demonstrating the efficacy of GoT. Through comprehensive case studies and ablation studies, we provide substantial evidence of the effectiveness of GoT in reasoning tasks. If you want it, you GoT it!

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. Computer Vision Foundation / IEEE Computer Society.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Lawrence W Barsalou. 1999. Perceptual symbol systems. *Behavioral and brain sciences*, 22(4):577–660.

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*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1380–1391. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven C. H. Hoi, Xiaogang Wang, and Hongsheng Li. 2019. Dynamic fusion with intra- and intermodality attention flow for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6639–6648. Computer Vision Foundation / IEEE.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. 2023. Language is not all you need: Aligning perception with language models. *CoRR*, abs/2302.14045.

Marek Kadlčík, Michal Štefánik, Ondrej Sotolar, and Vlastimil Martinek. 2023. Calc-X and calcformers: Empowering arithmetical chain-of-thought through interaction with symbolic systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12101–12108, Singapore. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. In *Findings of the*

Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, volume EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 1571–1581.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *CoRR*, abs/2205.11916.

Bei Li, Chuanhao Lv, Zefan Zhou, Tao Zhou, Tong Xiao, Anxiang Ma, and JingBo Zhu. 2022. On vision features in multimodal machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6327–6337, Dublin, Ireland. Association for Computational Linguistics.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2020. What does BERT with vision look at? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5265–5275. Association for Computational Linguistics.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *CoRR*, abs/2304.09842.

Pan Lu, Liang Qiu, Jiaqi Chen, Tanglin Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. 2021. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *CoRR*, abs/2306.14824.

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

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

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *CoRR*, abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022a. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

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

Zhiyong Wu, Lingpeng Kong, Wei Bi, Xiang Li, and Ben Kao. 2021. Good for misconceived reasons: An empirical revisiting on the need for visual context in multimodal machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6153–6166, Online. 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. *arXiv preprint arXiv:2305.10601*.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6281–6290. Computer Vision Foundation / IEEE.

Zhuosheng Zhang, Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Zuchao Li, and Hai Zhao. 2020. Neural machine translation with universal visual representation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhuosheng Zhang and Aston Zhang. 2023. You only look at screens: Multimodal chain-of-action agents. *CoRR*, abs/2309.11436.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *CoRR*, abs/2210.03493.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *CoRR*, abs/2302.00923.

Appendix

A Related Works

In chain-of-thought reasoning, one idea leads to the next in a logical sequence and builds on previous knowledge. Each idea is supported by evidence or reasoning, and the conclusions drawn from the chain are logical and sound. Most CoT methods can be divided into two categories based on how to generate the final answer: (1) prompting for CoT, including zero-shot CoT and few-shot CoT; and (2) fine-tuning for CoT.

Zero-shot CoT Prompting As large language models continue to advance rapidly, many researchers are beginning to explore CoT reasoning for LLMs. The zero-shot CoT method proposed by Kojima et al. (2022) consists of two stages: (1) adding a "*Let's think step by step*" prompt to generate CoT, and (2) concatenating the generated CoT and adding the phrase "*So the answer is*" to obtain the final answer. Tree-of-Thought (ToT) (Yao et al., 2023) enables deliberate decision-making through exploration of coherent text units. ToT divides thoughts into thought units and models them as a tree-like search process. Although both GoT and ToT aim to capture human non-linear thoughts, GoT is distinct from ToT in terms of both methodology and objectives. We believe that human thinking involves both linear and non-linear aspects. Thus, we build upon the linear CoT framework by incorporating non-linear structures to simultaneously capture both linear and non-linear human reasoning. Tree-of-thoughts focuses on modeling non-linear thoughts explicitly, whereas our approach leverages non-linear structures to assist the Chain-of-Thought reasoning.

Few-shot CoT Prompting Few-shot CoT reasoning for LLMs, however, utilizes multiple input-output pairs to prompt the LLMs to output CoT and obtain the final answer. Due to its ability to provide better performance compared to Zero-shot CoT, Few-shot CoT has gained more attention in research, particularly through effective demonstrations. Few-shot CoT prompting was first formally explored by Wei et al. (2022a) and is a form of discrete prompt learning that involves context learning in large models. Compared to traditional in-context learning, which prompts LLMs with a list of input-output demonstration pairs along with a test input to allow the model to predict output, Few-shot CoT prompting outputs additional logical reasoning procedures apart from the target output. Wang et al. (2022) proposed a follow-up method to (Wei et al., 2022a). The main improvement is that the model uses the majority vote for the answers, which was found to significantly improve the performance of the CoT. However, these few-shot CoT models depend on hand-crafted demonstrations. To solve this problem, Zhang et al. (2022) proposed Auto-CoT, which maintains the diversity of sampled questions and generates reasoning chains to automatically construct demonstrations. Specifically, Auto-CoT consists of two main stages: (1) Problem clustering: divide the given dataset of problems into several clusters; (2) Demonstration sampling: select a representative problem from each cluster and use a simple heuristic method to generate its reasoning chain. Furthermore, Lu et al. (2023) also explores few-shot CoT reasoning for recently popular LLMs ChatGPT and GPT-4.

CoT Fine-tuning In Zhang et al. (2023), it was proposed to fine-tune smaller language models instead of prompting them in LLMs. And this approach enabled the CoT to go beyond textual information and incorporate visual (image) modalities using a gated fusion mechanism into a two-stage CoT. The results demonstrated that CoT fine-tuning with fewer parameters has potential. Therefore, in this work, we focus on fine-tuning for CoT to reduce the number of required model parameters and help LLMs better comprehend different modalities. However, previous CoT research has been limited to different modalities, such as textual and vision information, without considering the deduction reasoning process. Therefore, in this work, we move beyond modeling the reasoning process solely as a thought chain and elevate it to a thought graph. We provide a more comprehensive and nuanced representation, enabling LLMs to perceive the deduction reasoning process accurately, resulting in more precise answer generation.

B Dataset

AQUA-RAT dataset consists of about 100,000 algebraic word problems with natural language rationales. For AQUA-RAT, the model is trained to reasoning through the steps to generate the final answer. ScienceQA benchmark is the pioneering large-scale dataset for multimodal science questions, equipped with comprehensive annotations for answers, including detailed lectures and explanations. The dataset contains 21k questions covering three subjects: natural science, language science, and social science. Each question is presented with a context in the form of natural language or an optional image. The model is trained to elucidate the reasoning process in natural language while choosing the answer from a set of options.

Splits	#Problems
Train	97467
Dev	254
Test	254

Table 5: AQUA-RAT dataset statistics (# denotes numbers)

Statistic	Number
Splits	
#Train	12,726
#Dev	4,241
#Test	4,241
#Total	21,208
Attribute	
#Subjects	3
#Topic	26
#Category	127
#Skill	379

Table 6: ScienceQA dataset statistics (# denotes numbers)

C Training Parameters

Parameters	Value
Epochs	100
Batch size for T5-base (per device)	10
Batch size for T5-large (per device)	8
Learning rate	5e-5
Weight decay	0.01
Max input length	512
Max number of nodes	150

Table 7: Training parameters for GoT

D Rationale Generation Results

The rationale genration results can be found in Table 8. We can observe from Table 8 that the impact of GoT on rationale generation is limited. We attribute this limitation to the fact that the input text for thought graph construction only includes questions and choices. Consequently, the thought graph constructed from such limited information can only facilitate constrained deductive reasoning. However, in the answer generation stage, when provided with rationales, the model needs to possess stronger deductive reasoning capabilities to understand the relationship between rationales, questions, and choices.

E Case Study

To facilitate a more illustrative comparison between GoT and the CoT, we have selected several representative examples. Figure 7 illustrates the examples from AQUA-RAT dataset. Figure 8 to Figure 11 illustrates examples from ScienceQA dataset. From Figure 8 and Figure 9, we can see that GoT can better understand the rationales and generate more accurate result. In Figure 10, we can see that when provided with wrong rationale, our model is more robust to the noise and can focus on more important key information. (We highlight the noisy wrong rationale in red and correct key rationale in green). Figure 11 presents a language problem which have less context and requires a certain amount of common sense knowledge. Hence, the impact of constructing a mind map on enhancing the model is not significant. Therefore, both GoT and CoT predict wrong answers.

F Representation Visualization

In order to demonstrate the deductive reasoning process of GoT more intuitively, we visualized the attention weights of the GoT encoder. The visualization results can be found in Figure 12. We took Figure 10 as an example. In Figure 10, even given a wrong rationale, GoT still manages to generate the right answer. We select 14 representative thought nodes and found that "blue","color", and "common" have the greatest weights which indicates that GoT guides the model to focus on more important words and conduct correct deductive reasoning. For the disruptive node "a hard object," our model can effectively discriminate against it and assign a lower attention weight to prevent the model from selecting incorrect answers, as traditional CoT models often do due to erroneous rationales.

G Limitation

Compared to Mutimodal-CoT (Zhang et al., 2023), incorporating GoT may result in additional computational costs and slightly slower training times.

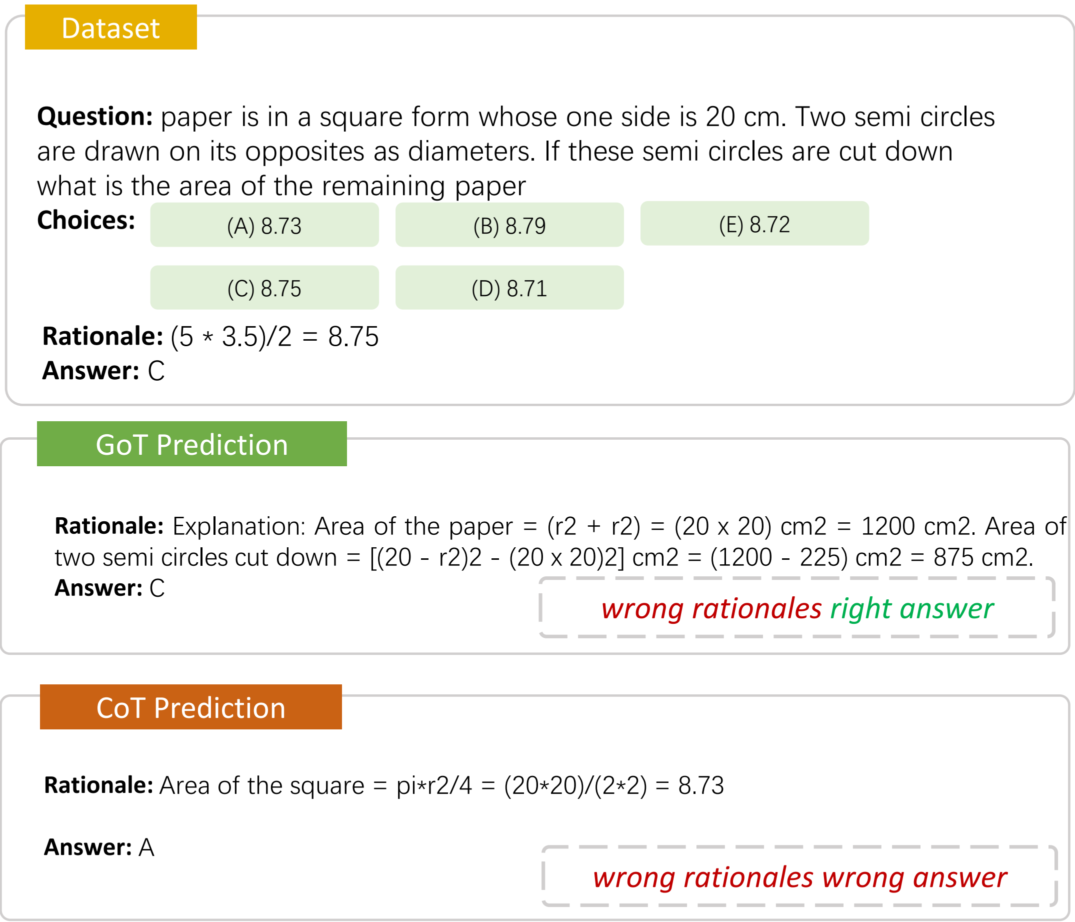
MODELS	BLEU1	BLEU4	ROUGE	SIMILARITY
AQUA-RAT				
FLAN-Alpaca$_{\text{base}}$	19.78	3.49	28.40	68.61
FLAN-Alpaca$_{\text{large}}$	22.45	5.40	29.55	70.34
GoT-T5$_{\text{base}}$	22.05	5.02	29.18	69.09
GoT-T5$_{\text{large}}$	**24.47**	**6.68**	**29.86**	**71.58**
ScienceQA				
Mutimodal-CoT$^{*}_{\text{base}}$ (Zhang et al., 2023)	91.04	86.81	93.24	96.34
GoT-T5$_{\text{base}}$	92.50	88.79	94.39	96.74
GoT-T5$_{\text{large}}$	**93.49**	**90.09**	**95.17**	**97.33**

Table 8: Rationale generation results (%). (*: we re-run the Mutimodal-CoT$_{\text{base}}$ to report the full rationale scores. We use sentence-transformers (https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2) to obtain sentence embeddings and calculate the cosine similarity for SIMILARITY)

The training parameters and inference times of the different models are presented in Table 9, which reveals that our model requires a 0.2% increase in parameters compared to Mutimodal-CoT.

	#Parameters	Inference time (eval samples/per second)
Mutimodal-CoT$_{\text{base}}$	227M	16.33
Ours	233M	13.38

Table 9: The number of training parameters and inference time of different models (# denotes numbers)



Dataset

Question: paper is in a square form whose one side is 20 cm. Two semi circles are drawn on its opposites as diameters. If these semi circles are cut down what is the area of the remaining paper

Choices: (A) 8.73 (B) 8.79 (E) 8.72 (C) 8.75 (D) 8.71

Rationale: (5 * 3.5)/2 = 8.75

Answer: C

GoT Prediction

Rationale: Explanation: Area of the paper = (r2 + r2) = (20 x 20) cm2 = 1200 cm2. Area of two semi circles cut down = [(20 - r2)2 - (20 x 20)2] cm2 = (1200 - 225) cm2 = 875 cm2.

Answer: C

wrong rationales right answer

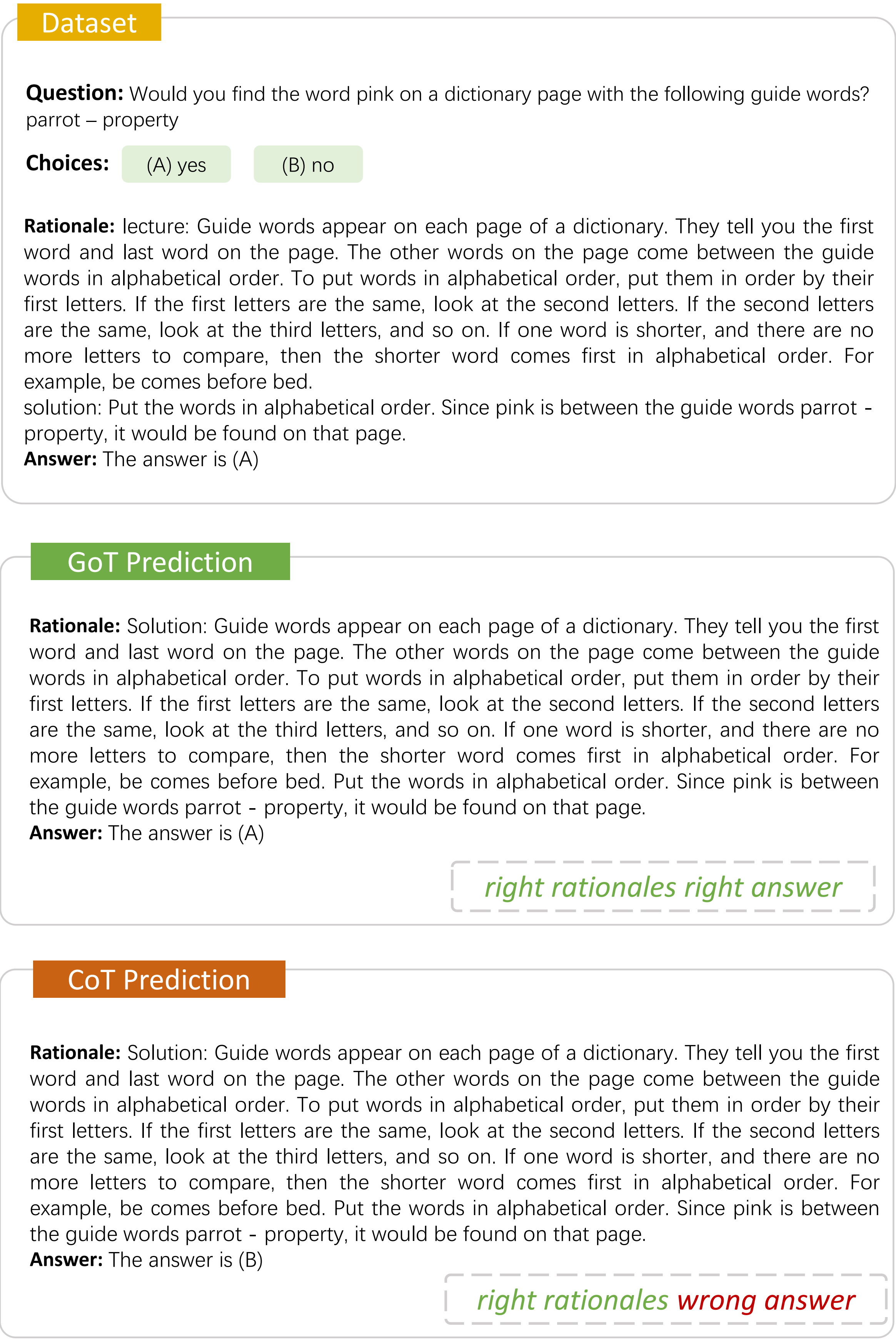
CoT Prediction

Rationale: Area of the square = pi*r2/4 = (20*20)/(2*2) = 8.73

Answer: A

wrong rationales wrong answer

Figure 7: Examples of AQUA-RAT



Dataset

Question: Would you find the word pink on a dictionary page with the following guide words?
parrot – property

Choices: (A) yes (B) no

Rationale: lecture: Guide words appear on each page of a dictionary. They tell you the first word and last word on the page. The other words on the page come between the guide words in alphabetical order. To put words in alphabetical order, put them in order by their first letters. If the first letters are the same, look at the second letters. If the second letters are the same, look at the third letters, and so on. If one word is shorter, and there are no more letters to compare, then the shorter word comes first in alphabetical order. For example, be comes before bed.
solution: Put the words in alphabetical order. Since pink is between the guide words parrot - property, it would be found on that page.
Answer: The answer is (A)

GoT Prediction

Rationale: Solution: Guide words appear on each page of a dictionary. They tell you the first word and last word on the page. The other words on the page come between the guide words in alphabetical order. To put words in alphabetical order, put them in order by their first letters. If the first letters are the same, look at the second letters. If the second letters are the same, look at the third letters, and so on. If one word is shorter, and there are no more letters to compare, then the shorter word comes first in alphabetical order. For example, be comes before bed. Put the words in alphabetical order. Since pink is between the guide words parrot - property, it would be found on that page.
Answer: The answer is (A)

right rationales right answer

CoT Prediction

Rationale: Solution: Guide words appear on each page of a dictionary. They tell you the first word and last word on the page. The other words on the page come between the guide words in alphabetical order. To put words in alphabetical order, put them in order by their first letters. If the first letters are the same, look at the second letters. If the second letters are the same, look at the third letters, and so on. If one word is shorter, and there are no more letters to compare, then the shorter word comes first in alphabetical order. For example, be comes before bed. Put the words in alphabetical order. Since pink is between the guide words parrot - property, it would be found on that page.
Answer: The answer is (B)

right rationales wrong answer

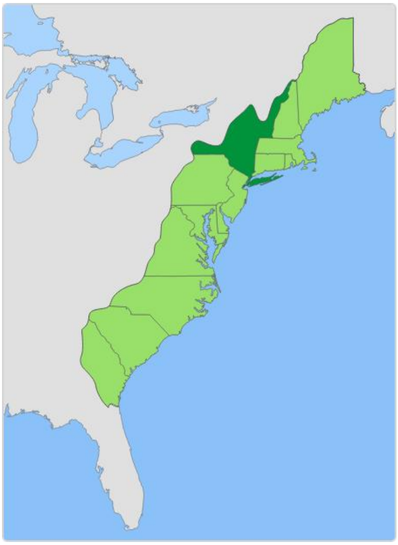
Figure 8: Examples of ScienceQA

Dataset

Question: What is the name of the colony shown?

Choices: (A) New Hampshire (B) Vermont (C) New York (D) Delaware

Rationale: The colony is New York. During the colonial era, New Hampshire and New York both claimed the territory that would later become the state of Vermont. Vermont was never its own colony.
Answer: The answer is (C)



GoT Prediction

Rationale: Solution: The colony is New York. During the colonial era, New Hampshire and New York both claimed the territory that would later become the state of Vermont. Vermont was never its own colony.
Answer: The answer is (C)

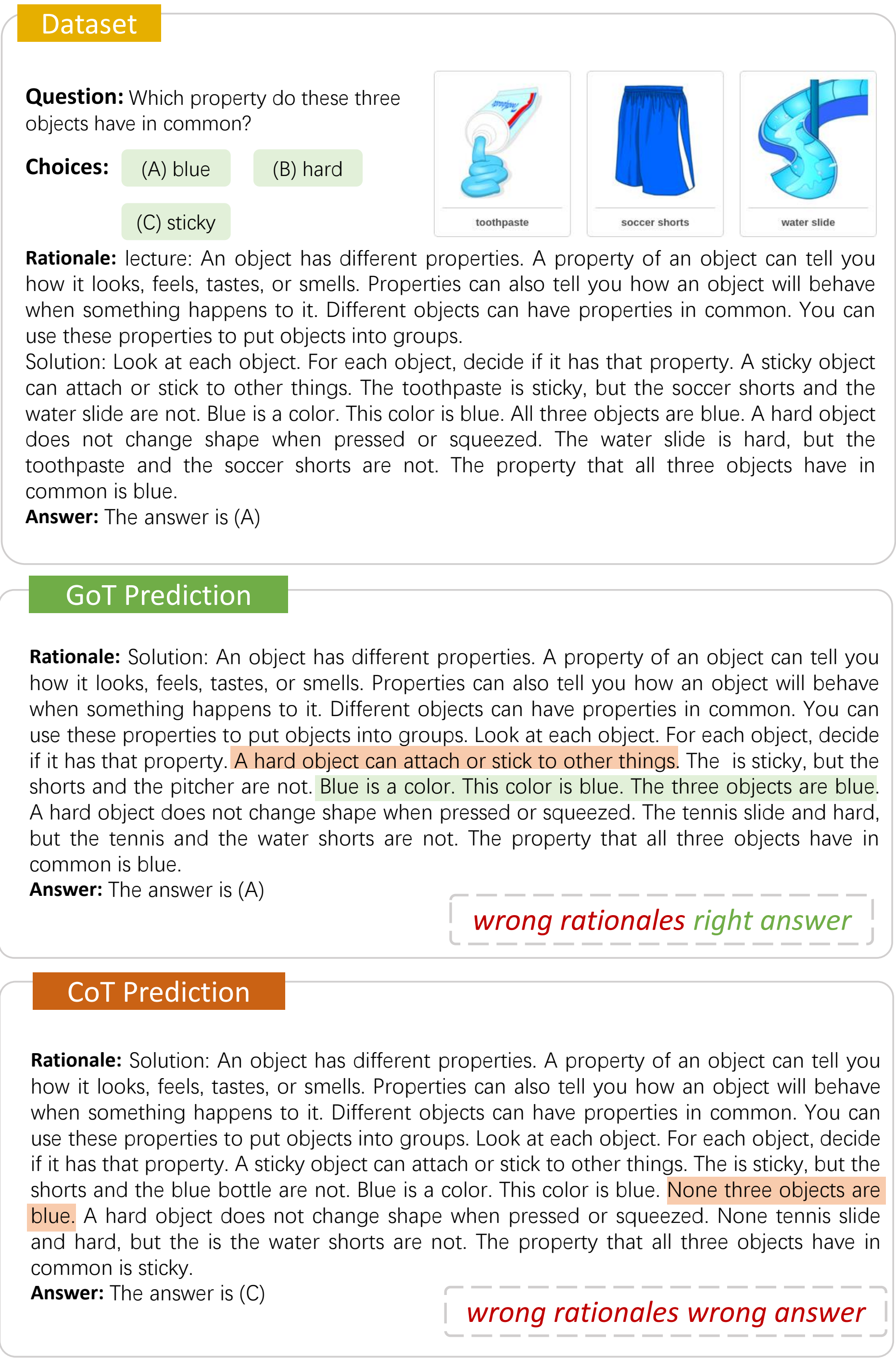
right rationales right answer

CoT Prediction

Rationale: Solution: The colony is Delaware York. During the colonial era, New Hampshire and New York both claimed the territory that would later become the state of Vermont. Vermont was never its own colony.
Answer: The answer is (D)

wrong rationales wrong answer

Figure 9: Examples of ScienceQA



Dataset

Question: Which property do these three objects have in common?

Choices: (A) blue (B) hard (C) sticky

toothpaste
soccer shorts
water slide

Rationale: lecture: An object has different properties. A property of an object can tell you how it looks, feels, tastes, or smells. Properties can also tell you how an object will behave when something happens to it. Different objects can have properties in common. You can use these properties to put objects into groups.
Solution: Look at each object. For each object, decide if it has that property. A sticky object can attach or stick to other things. The toothpaste is sticky, but the soccer shorts and the water slide are not. Blue is a color. This color is blue. All three objects are blue. A hard object does not change shape when pressed or squeezed. The water slide is hard, but the toothpaste and the soccer shorts are not. The property that all three objects have in common is blue.
Answer: The answer is (A)

GoT Prediction

Rationale: Solution: An object has different properties. A property of an object can tell you how it looks, feels, tastes, or smells. Properties can also tell you how an object will behave when something happens to it. Different objects can have properties in common. You can use these properties to put objects into groups. Look at each object. For each object, decide if it has that property. A hard object can attach or stick to other things. The is sticky, but the shorts and the pitcher are not. Blue is a color. This color is blue. The three objects are blue. A hard object does not change shape when pressed or squeezed. The tennis slide and hard, but the tennis and the water shorts are not. The property that all three objects have in common is blue.
Answer: The answer is (A)

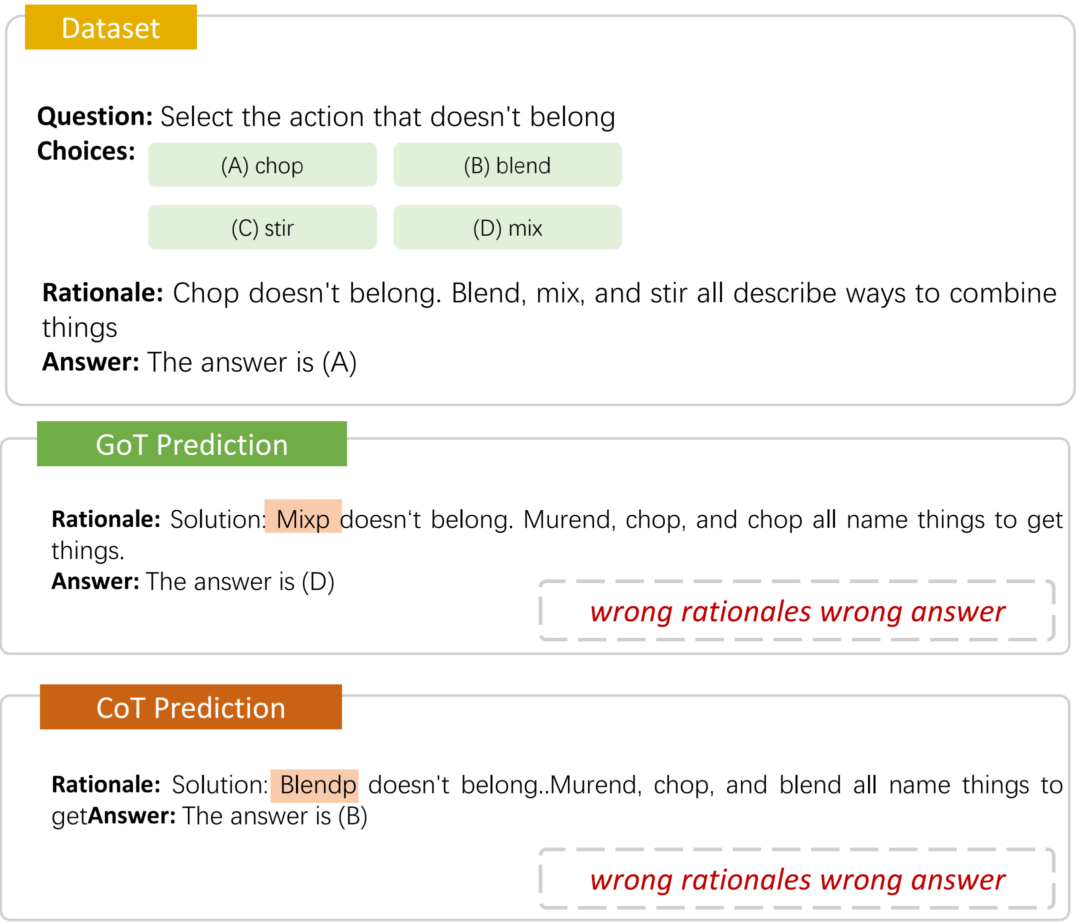
wrong rationales right answer

CoT Prediction

Rationale: Solution: An object has different properties. A property of an object can tell you how it looks, feels, tastes, or smells. Properties can also tell you how an object will behave when something happens to it. Different objects can have properties in common. You can use these properties to put objects into groups. Look at each object. For each object, decide if it has that property. A sticky object can attach or stick to other things. The is sticky, but the shorts and the blue bottle are not. Blue is a color. This color is blue. None three objects are blue. A hard object does not change shape when pressed or squeezed. None tennis slide and hard, but the is the water shorts are not. The property that all three objects have in common is sticky.
Answer: The answer is (C)

wrong rationales wrong answer

Figure 10: Examples of ScienceQA



Dataset

Question: Select the action that doesn't belong

Choices:

(A) chop

(B) blend

(C) stir

(D) mix

Rationale: Chop doesn't belong. Blend, mix, and stir all describe ways to combine things

Answer: The answer is (A)

GoT Prediction

Rationale: Solution: Mixp doesn‘t belong. Murend, chop, and chop all name things to get things.

Answer: The answer is (D)

wrong rationales wrong answer

CoT Prediction

Rationale: Solution: Blendp doesn't belong..Murend, chop, and blend all name things to get**Answer:** The answer is (B)

wrong rationales wrong answer

Figure 11: Examples of ScienceQA

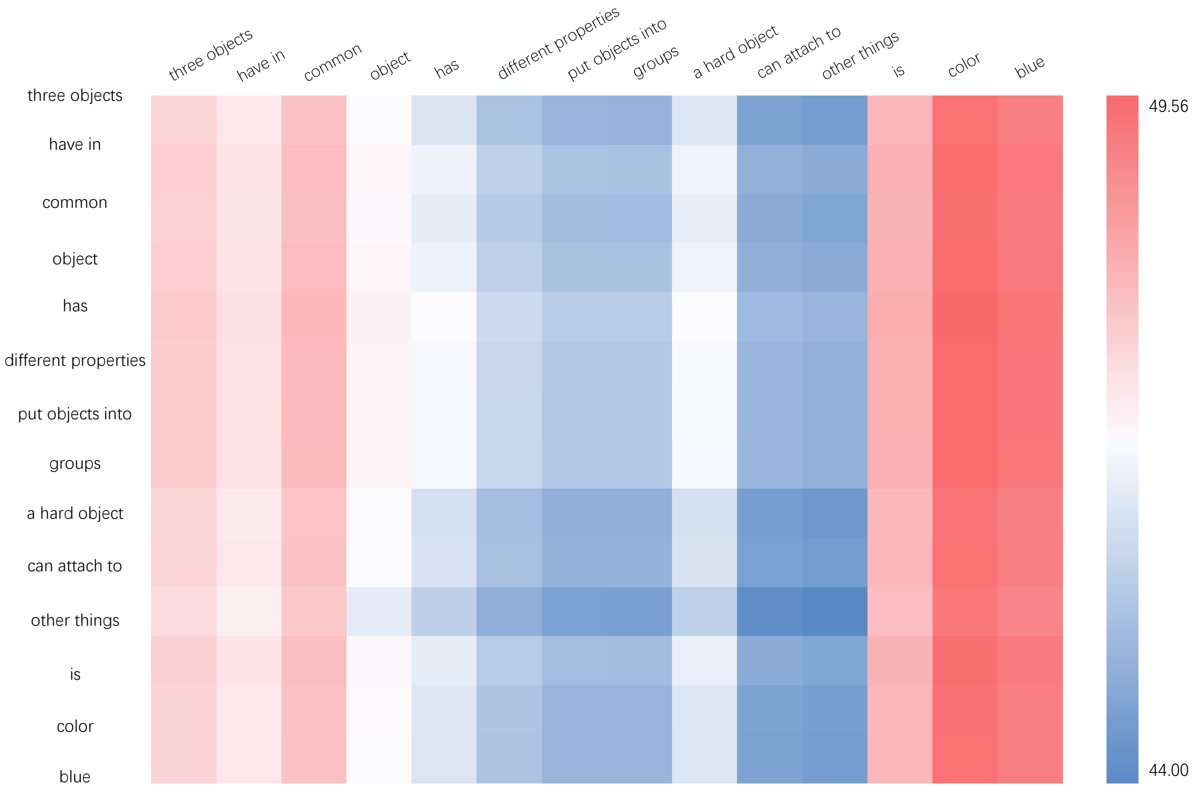


Figure 12: Representation visualization