Learning Chain of Counterfactual Thought for Bias-Robust Vision-Language Reasoning (Supplementary Materials)

Yifeng Zhang, Ming Jiang, and Qi Zhao

University of Minnesota, Minneapolis MN 55455, USA
{zhan6987, mjiang}@umn.edu, qzhao@cs.umn.edu

1 Introduction

In our main paper, we have presented Counterfactual Bias-Robust Reasoning (CoBRa) a novel counterfactual visual reasoning dataset with paired examples and rich annotations of reasoning processes, and Chain of Counterfactual Thought (CoCT), a bias-robust chain-of-thought (CoT) method that employs counterfactual thinking for improved generalizability for large vision-language models (LVLMs). Both of them address the knowledge bias hidden in LVLMs and contribute to better generalizability across different scenarios. The supplementary material provides additional experimental results and implementation details of our proposed work:

1. Sec. 2 details the construction process of the CoBRa dataset, offering insights into its design, including knowledge representations, knowledge editing rules, reasoning functions, and the generation of questions, answers, and counterfactual images.
2. Sec. 3 expands on the implementation details of CoCT, particularly focusing on training the TLM for predicting bias-robust reasoning processes.
3. Sec. 4 provides additional experimental analyses, including (1) model performance on other comprehensive LVLM benchmarks, (2) impacts of the number of in-context examples on model performance, and (3) additional qualitative examples demonstrating the CoCT prompts based on in-context examples from our CoBRa dataset.

2 CoBRa Dataset Creation

To supplement the main paper's introduction of CoBRa's automated generation pipeline, this section delves into the core components that enable its functionality. These include: (1) a unique knowledge graph representation for both visual and knowledge features, (2) a set of graph editing rules for manipulating the knowledge graph to create counterfactuals, (3) reasoning functions that sample questions based on the graph, (4) a question engine that generates question-answer pairs, and (5) diffusion models used for inpainting counterfactual images.

2.1 Knowledge Representations

CoBRa utilizes a unique knowledge graph representation for both visual and knowledge features. This approach leverages a well-established knowledge base and explicitly defined symbolic reasoning to create image-question pairs with assured consistency and integrity, enabling precise control over the knowledge distribution within the dataset. The process consists of the following steps: First, we curate a diverse set of base images from the established Wikipedia-based Image Text Dataset (WIT) [15]. Visual knowledge is then extracted from these images and their accompanying captions. This extraction utilizes an off-the-shelf scene graph generation method, namely MSDN [9]. The resulting structure is a graph where each node represents an entity, and the edges depict relationships between them. Additionally, following [16,17], each element (entity or relationship) within the graph is enriched with both visual and linguistic features. Auxiliary labels are attached to provide contextual information, including Part-of-Speech (POS) tags and Structural labels (TAG) parsed from captions generated by an off-the-shelf graph-to-caption model (*i.e.*, ASC2C [1]). These tags serve as constraints used in the knowledge editing to enforce the edited knowledge entities to share the same POS and TAG as the original ones.

This comprehensive knowledge graph representation empowers CoBRa with two key capabilities: It enables the use of predefined rules to automatically modify the knowledge graph, facilitating the creation of counterfactual examples (*e.g.*, altering "dog-on-horse" to "horse-on-dog"). In addition, diverse VQA questions can be generated based on the knowledge graph using pre-defined reasoning functions (*e.g.*, "Is the dog larger than the horse?"). Therefore, by controlling the knowledge representation, CoBRa ensures the generation of counterfactual examples while maintaining the integrity of the visual scene and the reasoning processes underlying the question-answering.

2.2 Knowledge Editing Rules

To balance the potentially biased knowledge distributions, our approach involves generating counterfactual variants of the base images and their corresponding knowledge representations. This deliberate modification, guided by multiple knowledge editing rules, introduces knowledge shifts, providing a robust training and evaluation platform for models to develop and assess their resistance to bias. The following are concrete definitions of the knowledge editing rules.

- **addEntity**: This rule adds a random entity to the scene.
- **addRelationship**: This rule randomly selects two entities and adds a valid relationship between them.
- **addProperty**: This rule randomly selects an entity and adds a valid property to it.
- **removeEntity**: This rule removes a random entity from the scene.
- **removeRelationship**: This rule randomly selects two entities and removes one relationship between them.

- **removeProperty**: This rule randomly selects an entity and removes a valid property of it.
- **replaceEntityClass**: This rule randomly replaces an entity with an alternative one. For example, "dog-on-horse" to "dog-on-wall".
- **replaceEntityProperty**: This rule randomly replaces a property of an entity. For example, from "mouse-hasSize-small" to "mouse-hasSize-large".
- **replaceEntityRelationship**: This rule randomly replaces a relationship between two entities. For example, from "man-drive-car" to "man-pull-car".
- **invertRelationship**: This rule inverts the order of the two entities connected by a relationship. For example, from "dog-on-horse" to "horse-on-dog".
- **swapProperty**: This rule swaps the property of two entities. For example, from "mouse-hasSize-small, elephant-hasSize-large" to "mouse-hasSize-small, element-hasSize-large")

This comprehensive approach not only enhances dataset diversity by incorporating multiple sources but also structures modifications to ensure a meaningful evaluation of model robustness in the face of knowledge shifts and bias.

2.3 Reasoning Functions

To facilitate automatic reasoning over the knowledge graph and the generation of example questions and answers, we leverage symbolic reasoning functions originally introduced in XNM [14]. Differently, we extend the attention mechanism of XNM to operate over visual regions while simultaneously performing logical operations on the symbolic labels (*e.g.*, POS and TAG) associated with entities within the knowledge graph. This approach bridges the gap between implicit visual-linguistic features and explicit conceptual knowledge.

The following definitions detail the reasoning functions used in our approach. Functions 1-5 (*i.e.*, Compare, Identify, Describe, Classify, Count) are closely relevant to question types while Functions 6-11 (*i.e.*, Count, Relate, Filter, Find, And, Or, Not) are more general intermediate steps. Here, A and B represent entities, while a, b are properties of A,B, respectively, r indicates the relationship between two entities, and $\mathcal{G}$ symbolizes the knowledge graph.

1. **Compare:** This function involves comparing two entities based on specified properties. It randomly selects properties a and b from the entities A and B, respectively, and generates a question that prompts a comparison between the selected properties.
2. **Identify:** This function focuses on checking the presence of specified properties of an entity. It randomly selects property a from entity A and generates a question that inquires about the existence of that property within the entity.
3. **Describe:** This function aims to summarize or describe the property or relationships of entities. It randomly selects property a from entity A or a relationship r between two entities, generating a question that solicits a summary or description of that property or relationship.

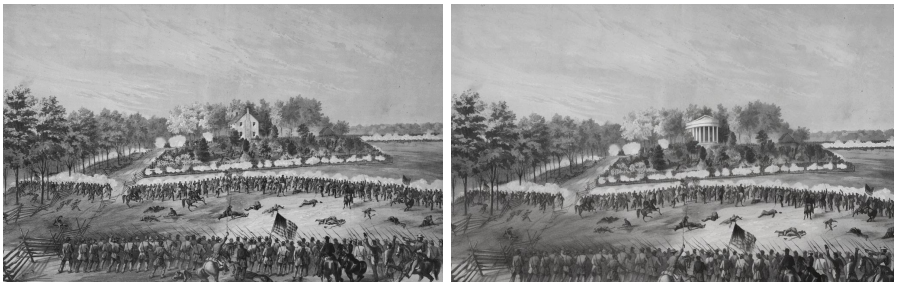

Fig. 1: This example shows a counterfactual scenario. The original image depicts the Battle of Jackson, Mississippi. However, through image inpainting, the building in the center has been altered to resemble the White House in Washington D.C. This change results in a different answer to the question asked: "Did this happen in Mississippi?"

4. **Classify:** This function involves checking for overlaps or similarities of the properties between a source entity and multiple target entities. It randomly selects all the properties $\{a\}_i$ of a source entity A, and properties from target entity candidates (B, C, D, *etc.*), generating a question that classifies the source entity into the most probable target.
5. **Count:** This function pertains to arithmetic inference and sums over the number of attended entities in Graph $\mathcal{G}$. It generates a question that requires arithmetic operations on the entities with specified properties or relationships, fostering reasoning involving numerical relationships.
6. **Relate:** This function introduces relational reasoning by exploring connections or associations between different properties within an entity or across entities.
7. **Filter:** This function involves isolating specific properties or entities based on certain criteria. It encourages reasoning about selective information extraction, contributing to a nuanced understanding of relevant details in the scene.
8. **Find:** This function explores specific entities, properties, or relationships. It enhances reasoning about the localization of particular elements, contributing to a more comprehensive assessment of knowledge reasoning skills.

9-11. **And / Or / Not:** These functions support the logical operations upon the output of existing functions. It facilitates the generation of questions with better compositionality.

By employing these reasoning functions and strategically combining them (as introduced in Sec. 2.4), we enable the generation of diverse questions and answers that require different reasoning skills.

2.4 Question Generation

This section details our approach to generating questions and answers for both original and counterfactual examples. We leverage our systematically designed

reasoning functions and question templates inspired by Zhang *et al.* [22]. The core idea lies in strategically matching these question templates with specific parameters sampled from the knowledge graph. These parameters include entities (*e.g.*, "dog," "house"), their properties (*e.g.*, "color," "size"), and the relationships between them (*e.g.*, "on," "in"). By plugging these parameters into the templates, we generate grammatically correct and semantically meaningful questions that demand different reasoning abilities.

For instance, consider the example in Fig. 1. The question "Did this happen in Mississippi?" appears simple, but it involves a sequence of reasoning steps represented by the following parameterized functions:

1. Find (all): Identify all relevant entities.
2. Relate (locationOf): Establish the relationship between the entity and its location.
3. Identify (Mississippi): Specify the target location.

Matching parameterized reasoning functions with question templates benefits flexible and controllable question generation: The same function (*e.g.*, Find) can be used with different parameters to target various entities within the scene. By selecting specific functions and parameters, we can tailor the generated questions to assess specific aspects of the model's reasoning capabilities. This approach allows us to move beyond simple question templates and create a rich set of questions that effectively probe the reasoning skills of models trained on our CoBRa dataset.

2.5 Answer Generation

Ground-truth answers to the generated questions are derived by directly applying the corresponding symbolic reasoning functions to the knowledge graph. A crucial aspect is the mapping between specific reasoning functions and the answer types they produce. For example, the "Identify" function would typically return an entity as the answer, while the "Compare" function might return a comparison operator ("taller than" or "smaller than") or a relative size descriptor ("large" or "small"). This mapping ensures that the generated answers align with the intended question type. Once the appropriate question type is identified based on the reasoning functions, the functions are applied to the knowledge graph [22] to retrieve the answer directly from the relevant knowledge entities. To guarantee the validity and relevance of the ground-truth answer, we retrieve entities from the knowledge graph that share the same POS tags (*e.g.* "noun", "verb") and structural labels (TAG) as those associated with the entities in the question. This ensures that the answer candidates are semantically compatible with the context of the question.

2.6 Counterfactual Image Generation

This section presents our approach for generating images, specifically counterfactual scenarios, that challenge potential biases in LVLMs. Here, we leverage

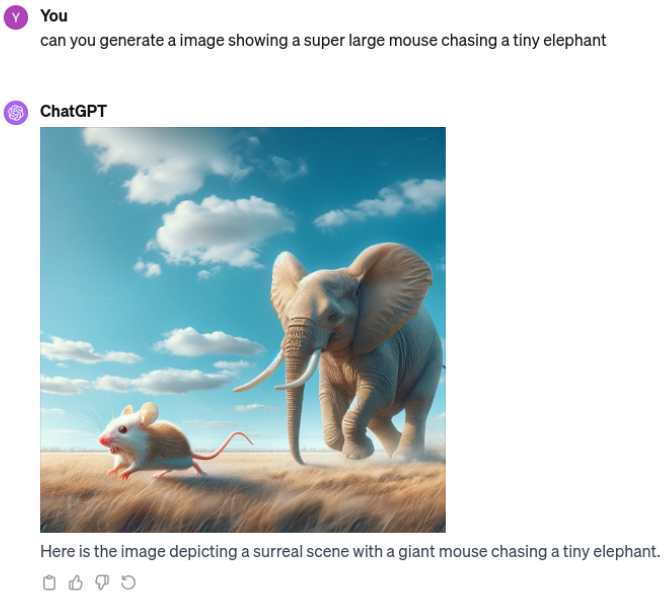

Fig. 2: This example demonstrates the limitations of current image generation models in handling counterfactual scenarios. Even with a detailed description specifying a "super large mouse" and a "tiny elephant," the model generates an image with typical sizes. This highlights the challenge of overcoming inherent biases within these models.

the capabilities of diffusion models while mitigating their limitations in handling unconventional situations.

Our core image generation technique employs Stable Diffusion 2 [13], a state-of-the-art diffusion model that excels at creating realistic images based on textual descriptions. Diffusion models often struggle when generating counterfactual scenes that deviate from their learned prior knowledge. As illustrated in Fig. 2, even detailed prompts specifying entities, properties, and relationships might result in "stereotypical" outputs (*e.g.*, a giant mouse is smaller than a tiny elephant), highlighting the need for improved handling of diverse scenarios.

To achieve image diversification and counterfactual generation, we incorporate an inpainting method instead of directly generating the whole image from the knowledge graph. This involves two key steps: First, in the original image, we localize the object or relationship to replace, using a high-performance object detection model (*i.e.*, YOLOv8 [6]). Based on the detected objects, depth maps are estimated to provide the diffusion model with spatial information about the scene [12]. Next, once the target elements are identified, we retrieve the edited facts from the knowledge graph with the information about the desired modifications (*e.g.*, replacing a small dog with a large one), and compose a detailed prompt that guides the diffusion model in generating the counterfactual sce-

nario. The prompt is composed of multiple phrases depicting the class, property, and relationships to neighboring objects, *e.g.*, `white house, surrounded by trees, grayscale` for Fig. 1.

To further diversify the generated images and explore a broader range of visual possibilities, we optionally incorporate multiple image-to-image style transfer models such as NNST [5]. These models allow us to modify the style of the image, independent of their content. This enables the generation of visually distinct outputs while maintaining the core counterfactual elements specified in the prompt.

By combining diffusion models with inpainting and optional style transfer, our approach tackles the challenges of bias, instability, and limited diversity in image generation from scene descriptions. This comprehensive strategy allows us to create counterfactual scenarios that challenge biases in LVLMs and facilitate the development of reasoning capabilities robust to such biases.

3 Translation Language Model (TLM)

In the main paper, we introduced CoCT, a CoT method leveraging TLMs to generate bias-robust reasoning processes. Here, we delve deeper into the specific architecture and training objectives employed by the TLM within our proposed CoCT.

The core of CoCT's TLM is based on the architecture originally proposed for bilingual translation with limited parallel data [7,8]. This encoder-decoder architecture allows languages from both the source and target domains to be projected into a shared latent space, facilitating reasoning across different modalities. In CoCT, we formulate the explicit prediction of reasoning functions and parameters from the visual-linguistic input as two unsupervised machine translation tasks. Specifically, we treat the visual-linguistic embeddings as a language in the source domain, and the functions/parameters as languages in the target domain. Following conventional TLM training settings, our training adheres to two key objectives [8]:

Denoising Auto-Encoding Loss. In CoCT, both source domain (visual-linguistic input) and target domain (reasoning functions and parameters) sequences are trained to reconstruct themselves from corrupted versions. Assume $\mathcal{S}$ and $\mathcal{T}$ are the sequences from the source and target domain, with a noisy model $C(\cdot)$ that randomly drops or swaps a token, the denoising auto-encoding loss is defined as

$$\mathcal{L}_{ae} = \mathrm{E}_{x\in\mathcal{S}}[-\log P_{s\to s}(x|C(x)] + \mathrm{E}_{y\in\mathcal{T}}[-\log P_{t\to t}(y|C(y)], \tag{1}$$

where $P_{s\to s}$ and $P_{t\to t}$ are the combination of encoder and decoder that operates on the source and target domain, respectively.

Back-Translation Loss. The unsupervised problem is converted into a supervised one through iterative back-translation. A naive model [8] is used to initialize the translation between the source and target domains (*e.g.*, visual-linguistic embedding to reasoning functions/parameters, and vice versa). These models are iteratively optimized, enhancing the overall TLM's ability to translate between the two domains.

Given the inferred translation from the source language $u(x)$ and that from the target language $v(y)$, the back-translation loss is defined as

$$\mathcal{L}_{bt} = \mathrm{E}_{x\in\mathcal{S}}[-\log P_{t\to s}(x|u(x)] + \mathrm{E}_{y\in\mathcal{T}}[-\log P_{s\to t}(y|v(y)], \tag{2}$$

where $P_{s\to t}$ and $P_{t\to s}$ are the TLM that translates sequences between the source and target domain. With both losses, we iteratively train a TLM from a naive encoder-decoder network in an unsupervised manner.

3.1 Combination with Bias-Robustness Loss.

As introduced in the main paper, the final objective function is defined as the combination of the denoising auto-encoding loss $\mathcal{L}_{ae}$, the back-translation loss $\mathcal{L}_{bt}$, and the bias-robust loss $\mathcal{L}_{br}$ (see Section 4.1 in the main paper):

$$\mathcal{L} = \mathcal{L}_{br} + \mathcal{L}_{ae} + \mathcal{L}_{bt}. \tag{3}$$

This combined loss function ensures that CoCT prioritizes generating bias-robust reasoning processes while maintaining the core functionalities of the TLM, transforming it from a naive encoder-decoder network into a system capable of extracting bias-robust reasoning processes directly from the input.

4 Supplemental Results

In the main paper, our experimental analyses have focused on demonstrating the effectiveness of CoCT and its various components in achieving robust reasoning performance across different scenarios. Here, we extend our analyses by presenting additional experimental results, including (1) a performance comparison on comprehensive LVLM benchmarks, (2) an ablation study on the number of in-context examples, and (3) qualitative examples of the CoCT prompts. These supplemental findings solidify the effectiveness of CoCT in promoting bias-robust reasoning within LVLMs.

4.1 Results on Comprehensive LVLM Benchmarks

Mitigating knowledge bias is essential for enhancing LVLMs' reasoning capabilities. As an extension to the evaluations in the main paper, we assess CoCT's performance on additional datasets: MM-Vet [18], MME [4], MMMU [19]) datasets.

Table 1: Comparison of model performance on benchmarks that evaluates comprehensive reasoning capabilities (*i.e.*, knowledge, compositionality, recognition, *etc.*).

Method	MM-Vet	MME	MMMU
LLaVA-1.5	31.1	1510.7	37.4
+ AutoCoT [23]	35.9	1534.8	38.5
+ AP [3]	39.4	1519.6	39.3
+ DDCoT [24]	41.8	1568.9	40.6
+ Ours (GQA)	40.8	1556.3	37.6
+ Ours	**44.5**	**1572.9**	**42.1**
GPT-4V	56.8	1409.4	56.8
+ AutoCoT [23]	58.5	1459.2	57.0
+ AP [3]	58.7	1453.2	57.2
+ DDCoT [24]	58.8	**1484.5**	57.2
+ Ours (GQA)	58.6	1472.8	56.9
+ Ours	**59.0**	1478.1	**57.4**

Tab. 1 summarizes the comparative results of various CoT methods applied to LLaVA-1.5 and GPT-4V. Our method surpasses all CoT-based approaches on both LVLMs, achieving the best performance on 5 out of 6 benchmarks. It is only outperformed by DDCoT [24] with GPT-4V on MME, likely due to its exposure to more language tasks during fine-tuning with SQA [11]. These findings highlight the critical role of bias-robust reasoning in improving the overall reasoning abilities of LVLMs. The results also demonstrate the benefits of prompting with pairs of original and counterfactual in-context examples for reasoning tasks. The combination of CoBRa and CoCT consistently outperforms the GQA counterpart, exhibiting the most significant improvement on the MMMU benchmark (*e.g.*, 37.6→42.1 with LLaVA-1.5). This suggests that exposing LVLMs to diverse scenarios, achieved through counterfactual prompts with step-by-step reasoning processes, strengthens their ability to reason and mitigates potential biases learned from internet data.

4.2 Results on C-VQA and VCR

Tab. 2 shows the model performance on VCR [20] and the newly released C-VQA [21] on query-contemplated counterfactual scenarios. Our CoCT(CoBRa) significantly improves LLaVA-1.5-7B and GPT-4V performance on VCR and C-VQA, validating its bias-mitigation and reasoning capabilities in more general settings.

4.3 Ablation Study on Number of In-Context Examples

To optimize CoCT's performance, we present an ablation study focusing on the number of in-context examples used during prompting. This section analyzes how

Table 2: Comparison of model performance on VCR and C-VQA dataset.

Dataset	VCR			C-VQA		
Method	Q→A	QA→R	Q→AR	Num-D.	Num-I.	Bool
LLaVA-1.5-7B	86.2	88.9	76.1	-23.9	-24.8	-27.7
+CoCT(CoBRa)	**86.8**	**89.3**	**77.9**	**-20.7**	**-24.2**	**-27.1**
GPT-4V	97.6	89.2	77.9	-35.8	-16.1	-17.9
+CoCT(CoBRa)	**97.8**	**89.3**	**78.2**	**-33.4**	**-14.6**	**-17.4**

Table 3: Impacts of different numbers of in-context examples.

# of Examples	CoBRa-O	CoBRa-C	$\Delta \downarrow$	HB	Bingo	SQA
4	68.5	27.1	41.4	29.4	18.8	67.9
8	70.2	30.0	40.2	29.9	18.8	68.7
16	73.8	34.2	39.6	30.8	23.4	70.4
24	**76.4**	**38.3**	**38.1**	**32.0**	**29.8**	**71.3**

this hyperparameter affects the reasoning performance of LVLMs. We evaluate model performance on CoBRa, HB [10], and Bingo [2] and SQA [11] datasets.

The results of this ablation study are presented in Tab. 3, revealing a general trend: incorporating more in-context examples can improve reasoning on counterfactual examples. This is likely because a larger number of examples provides the model with a richer pool of information and reduces the influence of potential biases within individual examples. However, processing a larger prompt with more examples naturally takes more computational resources. Besides, many LVLMs have limitations on the maximum number of tokens allowed in a single prompt. To balance these factors, CoCT adopts a setting of 24 in-context examples. This choice prioritizes achieving good performance while maintaining reasonable inference time and adhering to common prompt length limitations within the CoT literature [3, 23]. For a consistent comparison across all experiments, the results presented throughout this work are all based on 24 in-context examples (12 original-counterfactual pairs).

4.4 Qualitative Examples

In the main paper, we have introduced CoCT's methodology and its effectiveness in mitigating bias. Here, we delve deeper with two qualitative examples (presented in Tab. 4 and Tab. 5) to illustrate how CoCT reasons through challenging questions.

The example shown in Tab. 4 revisits an example from the methodology section (Section 4) of the main paper, focusing on the question "Is the baby a teacher?". Here is a breakdown of CoCT's reasoning process: First, CoCT explicitly predicts the reasoning process required to answer the question. In this case,

it identifies the two steps: "Find(baby)" followed by "Relate(teacher)". Based on the predicted reasoning process, CoCT finds pairs of original and counterfactual examples that share similar reasoning functions in diverse scenes (*e.g.*, a man holding a basketball *vs.* a soccer ball, a diesel train *vs.* a steam train). By analyzing these diverse scenarios, CoCT combats the potential bias that babies are primarily associated with learning rather than teaching.

Thie example in Tab. 4 demonstrates CoCT's ability to answer counting questions while mitigating bias. The question is "How many cows are there in the image?" and here is how CoCT tackles this challenge: It first predicts the

Table 4: The CoCT-generated prompt for the question "Is the baby a teacher?" using two pairs of in-context examples.

Please mimic these examples to answer the test question.		
V: 	Q: Is he shooting a basketball? R: 1. Find(man) 2. Relate(shoot) 3. Identify(basketball) A: Yes	K: **man-shoot-basketball** man-wear-red shirt shirt-locationOf-text **basketball-has-"Wilson"** background-include-spectator spectator-sit in-bleacher gymnasium-locationOf-indoor man-play-basketball game
V: 	Q: Is he shooting a basketball? R: 1. Find(man) 2. Relate(shoot) 3. Identify(basketball) A: No	K: **man-hold-soccer ball** man-wear-red shirt shirt-locationOf-text **soccer ball-has-colorful pattern** background-include-spectator spectator-sit in-bleacher gymnasium-locationOf-indoor **man-play-sport**
V: 	Q: What is the type of this train, diesel or steam? R: 1. Find(train) 2. Relate(poweredBy) 3. Identify(diesel) A: Diesel Train	K: **diesel locomotive-has-"9724"** railway track-is-straight **train-composedOf-cargo** **locomotive-paintedIn-red/yellow** **cargo-load-materials** building-locationOf-gabled roof building-locationOf-railway track people-locationOf-train mountain-behind-tree trees-line-railway tracks sky-above-scene
V: 	Q: What is the type of this train, diesel or steam? R: 1. Find(train) 2. Relate(poweredBy) 3. Identify(diesel) A: Steam Train	K: **steam locomotive-emit-smoke** railway track-is-straight train-on-railway track **train-paintedIn-black and yellow** trees-line-railway tracks building-locationOf-gabled roof building-locationOf-railway track people-standOn-train mountain-behind-tree sky-above-scene
V: 	Q: Is the baby a teacher? R: 1. Find(baby) 2. Relate(teacher) 3. Identify() A:	GPT-4V: **Yes**

Table 5: The CoCT-generated prompt for the question "Is the baby a teacher?" using two pairs of in-context examples.

Please mimic these examples to answer the test question.		
V: 	Q: How many frogs are there in the image? R: 1. Find(frog) 2. Count() A: 6	K: frog1-leftOf-frog2 frog1-topOf-frog3 frog3-leftOf-frog4 frog2-topOf-frog4 frog3-topOf-frog5 **frog5-leftOf-frog6** **frog4-topOf-frog6**
V: 	Q: How many frogs are there in the image? R: 1. Find(frog) 2. Count() A: 6	K: frog1-leftOf-frog2 frog1-topOf-frog3 frog3-leftOf-frog4 frog2-topOf-frog4 frog3-topOf-frog5 **frog5-leftOf-snake** **frog4-topOf-snake** **frog3-hasColor-light green**
V: 	Q: How many butterflies are there in the image? R: 1. Find(butterfly) 2. Count() A: 4	K: butterfly1-leftOf-butterfly2 **butterfly1-topOf-butterfly3** **butterfly3-leftOf-butterfly4** butterfly2-topOf-butterfly4
V: 	Q: How many butterflies are there in the image? R: 1. Find(butterfly) 2. Count() A: 3	K: butterfly1-leftOf-butterfly2 **butterfly1-topOf-maple leaves** **maple leaves-leftOf-butterfly4** butterfly2-topOf-butterfly4 **maple leaves-hasColor-red**
V: 	Q: How many cows are there in the image? R: 1. Find(cow) 2. Count() A:	GPT-4V: **1**

reasoning process as "Find(cow)" by "Count()" for the target object (cow). CoCT then selects in-context examples requiring the reasoning process of identifying and counting similar objects (*e.g.*, frogs and butterflies) amidst distractions. By scrutinizing the examples and observing how the model differentiates between target objects and distractions, CoCT learns to focus on the relevant objects and avoid biases that might lead to miscounting due to the presence of other objects.

These examples demonstrate two important aspects of CoCT's role in promoting bias-robust reasoning: It effectively identifies in-context examples similar to the test examples in terms of the reasoning process, and combats potential biases by exposing the model to diverse counterfactual examples.

References

1. Chen, S., Jin, Q., Wang, P., Wu, Q.: Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. CoRR **abs/2003.00387** (2020), https://arxiv.org/abs/2003.00387 2
2. Cui, C., Zhou, Y., Yang, X., Wu, S., Zhang, L., Zou, J., Yao, H.: Holistic analysis of hallucination in gpt-4v (ision): Bias and interference challenges. arXiv preprint arXiv:2311.03287 (2023) 10
3. Diao, S., Wang, P., Lin, Y., Zhang, T.: Active prompting with chain-of-thought for large language models (2023) 9, 10
4. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Qiu, Z., Lin, W., Yang, J., Zheng, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023) 8
5. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. CoRR **abs/1508.06576** (2015), http://arxiv.org/abs/1508.06576 7
6. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics 6
7. Lample, G., Conneau, A.: Cross-lingual language model pretraining. CoRR **abs/1901.07291** (2019), http://arxiv.org/abs/1901.07291 7
8. Lample, G., Ott, M., Conneau, A., Denoyer, L., Ranzato, M.: Phrase-based & neural unsupervised machine translation. CoRR **abs/1804.07755** (2018), http://arxiv.org/abs/1804.07755 7, 8
9. Li, Y., Ouyang, W., Zhou, B., Wang, K., Wang, X.: Scene graph generation from objects, phrases and region captions. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 2
10. Liu, F., Guan, T., Li, Z., Chen, L., Yacoob, Y., Manocha, D., Zhou, T.: Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multimodality models. arXiv preprint arXiv:2310.14566 (2023) 10
11. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. In: The 36th Conference on Neural Information Processing Systems (NeurIPS) (2022) 9, 10
12. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(3) (2022) 6
13. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 6
14. Shi, J., Zhang, H., Li, J.: Explainable and explicit visual reasoning over scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8376–8384 (2019) 3
15. Srinivasan, K., Raman, K., Chen, J., Bendersky, M., Najork, M.: Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 2443–2449. SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3404835.3463257, https://doi.org/10.1145/3404835.3463257 2
16. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019) 2

17. Vasiliev, Y.: Natural language processing with Python and spaCy: A practical introduction. No Starch Press (2020) 2
18. Yu, W., Yang, Z., Li, L., Wang, J., Lin, K., Liu, Z., Wang, X., Wang, L.: Mm-vet: Evaluating large multimodal models for integrated capabilities. arXiv preprint arXiv:2308.02490 (2023) 8
19. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., Chen, W.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. arXiv preprint arXiv:2311.16502 (2023) 8
20. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: Visual commonsense reasoning. In: CVPR. pp. 6713–6724 (2019) 9
21. Zhang, L., Zhai, X., Zhao, Z., Zong, Y., Wen, X., Zhao, B.: What if the tv was off? examining counterfactual reasoning abilities of multi-modal language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21853–21862 (2024) 9
22. Zhang, Y., Jiang, M., Zhao, Q.: New datasets and models for contextual reasoning in visual dialog. In: European Conference on Computer Vision. pp. 434–451. Springer (2022) 5
23. Zhang, Z., Zhang, A., Li, M., Smola, A.: Automatic chain of thought prompting in large language models (2022) 9, 10
24. Zheng, G., Yang, B., Tang, J., Zhou, H.Y., Yang, S.: Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models (2023) 9