Dynamic Clue Bottlenecks: Inherently Interpretable VQA with Abductive Reasoning

Anonymous EMNLP submission

Abstract

End-to-end visual question answering (VQA) models are now extremely effective, but we have little insight into why they succeed, undermining trust and applicability in critical domains. While post-hoc explanation offers some insight, these explanations are not guaranteed to be faithful to the model. Instead, we propose the Dynamic Clue Bottleneck Model (DCLUB), a broad coverage method that is designed to be interpretable from the beginning. In DCLUB, an answer to a question is based on a set of visually salient natural language statements of evidence (clues), forming an interpretable information bottleneck. DCLUB connects visual clues to possible answers via conditions, and finally evaluates the overall support for an answer via Natural Language Inference. Our system exposes what evidence it is using, why it supports an answer, and how much these factors contributed to the final prediction. Crucially, relevant clues and conditions are question and image specific, and must be dynamically constructed to be effective. To supervise the steps within DCLUB, we collect a dataset of 1.7k question with visual clues. Evaluations show our system maintains 87.4% of the black-box performance on benchmark data from VQA v2, and even improves by 1.5% on a reasoning-focused subset. Overall, our approach shows it is possible to design and supervise inherently interpetable VQA systems that can achieve comparable performance to black-box systems.

1 Introduction

Recent advances in large transformer-based pretrained models have achieved significant improvements in multi-modal reasoning (Wei et al., 2022a; OpenAI, 2023; Dai et al., 2023; Lu et al., 2019; Li et al., 2020). Yet sources of improvement have also been questioned over time (Saparov and He, 2022; Tang et al., 2023; Agrawal et al., 2018), with it being suggested that such models succeed because of shortcuts or otherwise inappropriate heuristics.

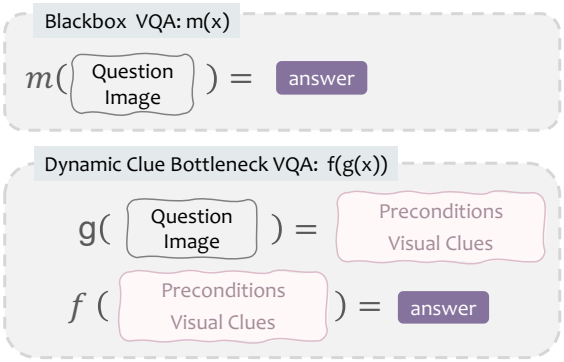

Figure 1: Design differences between de facto blackbox VQA methods (top), and our proposed dynamic clue bottlenecks method (lower). Blackbox models learn a $m(x)$ that maps from image and questions, x to answer directly. In contrast our approach offers interpretability by first generating intermediate bottlenecks with g, in forms of preconditions and visual clues, (defined in Section 3.1), and then deducing the final output with f, that limits itself to information in the bottleneck.

Increased trust in such models could be achieved through explanations of their predictions. Yet most of these models are black boxes, necessitating the use of post-hoc explanation methods (Selvaraju et al., 2016; Park et al., 2018). Problematically, existing post-hoc explanations may not to be faithful to the model they are explaining. Therefore, we focus on designing multi-modal models that are inherently interpretable. The main challenge of such models is that they tend to dramatically underperform their black-box counterparts, and that they usually require human understandable components which are hard to specify both of which we address for visual question answering (VQA).

We propose the Dynamic Clue Bottleneck Model (DCLUB) for interpretable visual question answering. Our system design takes inspiration from the concept bottleneck model (Koh et al., 2020). Unlike end-to-end VQA systems, DCLUB is interpretable because it factors prediction into two stages: a human readable information bottle-

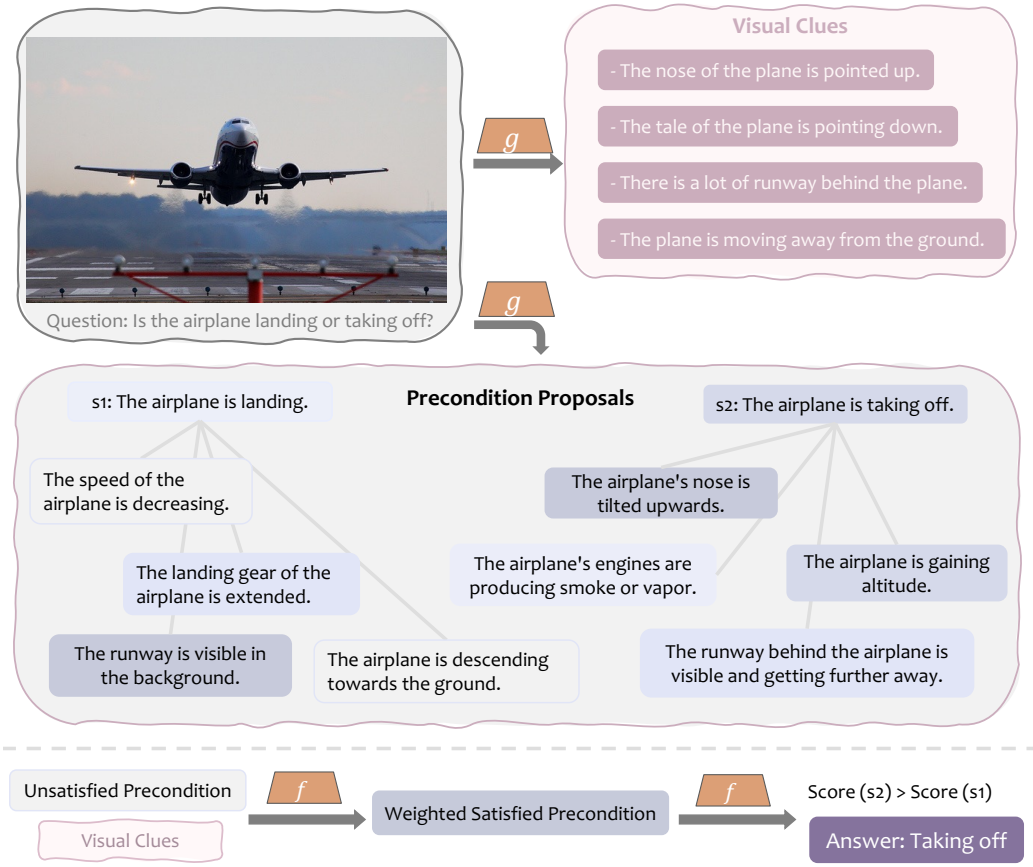

Figure 2: A detailed illustration of our DCLUB system on an example VQA data, with explicit steps of (1) visual clue generation (top), (2) abductive proposals on dynamic preconditions (middle), and (3) entailment based final prediction based on how much visual clues support each precondition (bottom). We use color to reflect support level – the deep blue blocks represent strong support by visual clues, and blank blocks represent no support.

neck, and a predictor that only conditions on the bottleneck. Abstractly, bottleneck models factor the full predictor as $y = f(g(x))$, where g is the bottleneck predicting function, and f forms the final prediction. g is allowed to be arbitrarily complex, adding expressivity, but f must be easy for a person to understand, enforcing interpretability. Previous work manually designed g or specified it with common sense from language models (Yang et al., 2023), for image classification. VQA is uniquely challenging and subsumes classification: information relevant for answering a question is image dependant, the connection between information and answers may be under-specified, and require abstract reasoning.

We address the challenges of building bottlenecks for VQA by creating a system that tries to generate simple natural language arguments for an answer. The bottleneck generating function is responsible for (1) proposing answers, (2) proposing preconditions for answers (3) and extracting natural language clues from an image that may be useful for answering a question. Possible answers and preconditions are generated abductively with GPT models prompted with a question, while clues are generated by a BLIP model fine-tuned to generate such clues. For example, in Figure 2, g generates all visual clues, possible answers (s1,s2), preconditions, and their connections. The final predictor, f, iterates over this structure with an NLI system evaluating how much clues support preconditions for answer proposals, returning the answer with highest aggregate support.

We also collect a dataset of 1.7k VQA instances requiring reasoning (Selvaraju et al., 2020; Goyal et al., 2017) to annotate with visual clues. The dataset is used to fine tune our visual clue generator, and as a reasoning focused test set. In comparison to a BLIP-2 model fine tuned on equal data, our system maintains 87.4% of performance on benchmark data from VQA v2 and even improves by 1.5% on our reasoning focused subset. Our results

demonstrate a promising direction toward building inherently interpretable multi-modal systems that perform as well as their black-box counterparts.

2 Related Works

2.1 VQA Interpretability

Some earlier attempts to rationalize VQA decisions (Xiong et al., 2016; Shih et al., 2016; Das et al., 2017) try to answer the question "Where should we look at the image to answer the question?" through attention maps. However, it is by design unclear how focusing on certain parts of the image help answer the question. Some recent VQA datasets are designed to encourage interpretability (Fu et al., 2022). Several previous works in this direction have achieved higher interpretability by generating visual attention scores as intermediate steps of black box models (Xiong et al., 2016; Shih et al., 2016; Das et al., 2017; Hendricks et al., 2018; Anderson et al., 2018), or by generating post-hoc natural language explanations (Hendricks et al., 2016; Dua et al., 2021; Marino et al., 2019; Schwenk et al., 2022), so as to reason between vision and language inputs at the same time. Considering that attention scores are not interpretable to humans since they cannot clearly state how the attended area connects to the final answer and can under-represent reasoning-required questions and that post-hoc explanations are not proven to be the exact reasons for the model to make the final predictions, neither of the two lines of methods is ideal.

2.2 Textual Interpretability

With the rapid development of LLMs, the community has spent major effort on language model reasoning (Wei et al., 2022b; Creswell et al., 2023; Yao et al., 2023; Hong et al., 2023). Some papers propose a decomposition process with LMs (Zhou et al., 2022; Khattab et al., 2022), or use post-hoc explanations from LLMs as a training or inference signal (Feng et al., 2023). These methods have shown large improvement and huge potential, but cannot be simply applied to the multi-modal area.

3 Dynamic Clue Bottlenecks VQA

As illustrated in Figure 1, blackbox VQA methods in general learn a function m for the answer prediction $y = m(x)$ where x denotes the question and image input. In contrast, our DCLUB composes two functions f and g and predicts following $y = f(g(x))$, where $g(x)$ is our bottleneck model induced by combinations of visual clues and preconditions, and f is a weighted summation function constructed via NLI. To better illustrate our overall method, we first define the aforementioned two terms as following.

Visual Clues. We define *visual clue* as natural language descriptions that help to answer the question while entirely grounded inside the corresponding image. This is part of the intended output of $g(x)$. For clarification, for cases with same question and different images, or with same image and different questions, visual clues should be different. Example can be found in Figure 6.

Preconditions. We define *preconditions* as the set of natural language descriptions of the entailing premises for all the possible answers to the question, which is independent of the image. Note that one precondition exclusively supports one possible answer. This is part of the intended output of $g(x)$.

Formulation. The overview of the proposed model is shown in Figure 2. Specifically, our method consists of three steps: abduction for generating possible answers and preconditions and a visual clue generation for learning $g(x)$, and symbolic answering component based on natural language inference for learning the $f(x)$.

Given a visual question answering pair (v, q, a) where v denotes the image, q denotes the question, and a represents the answer respectively, we generate a set of possible answer proposals $\hat{A} = \{\hat{a_1}, \hat{a_2}, ...\}$ based on v and q. Then, we propose a set of preconditions C_k for each possible answer $\hat{a_k}$ using frozen LLM. Finally, we deplop a novel entailment-based method to score preconditions, the visual clues, and answer and conduct a symbolic weighted summation over these cores to deduce the final answer.

3.1 Possible Answer Proposals

With the intention to build a dynamic clue bottleneck model, we find it difficult to keep the system interpretable by design without grounding the open-ended generation problem into a classification-like setting. For this purpose, our collect answer proposals, which are all the possible answers for (v, q) using world knowledge to constrain our output. Note that the possible answers can include incorrect ones, and our goal is to maximize recall with fewest number of answer proposals. We present two different kinds of methods – finetuning a multi-modal model such as BLIP-2 (Li et al., 2023), and

What are the most likely answers for the given question? Avoid unsure answers. Keep the answers as short as possible and separate with '; '

Question: Is the banana ripe enough to eat?
Possible answers: Yes; No

Question: Is the ball headed toward the batter or away from him?
Possible answers: Toward; Away

Question: Why are there candles on the cake?
Possible answers: Birthday; Outage

Question: What are they celebrating?
Possible answers: Birthday; Christmas; Thanksgiving; Wedding

Question: Is the plane landing or taking off?
Possible answers:

Landing; Taking off

Figure 3: An example prompt we use to generate answer proposals through frozen LLMs. The bottom part is GPT output for the example in Figure 2.

in-context prompting a frozen pre-trained LLM.

Answer Proposals with BLIP-2. We fine-tune the BLIP-2 model with all possible answers concatenated as the target output. Because the VQA dataset provides 10 correct answers for each question, we use it conveniently as the supervision.

Answer Proposals with LLM. Visual-based candidate answer generation may not be comprehensive since visual models are usually not as strong an abductive reasoner as LLMs. Alternatively, we compliment the candidate answer generation process with LLM prompting following (Fu et al., 2023) with a fixed prompt shown in Fig. 3.

3.2 Preconditions Proposals

We use the abductive power of LLMs to dynamically generate preconditions or hypothetical premises for each candidate answer to hold with respect to the question. This process uses only textual reasoning and is independent of the image, which serves as a main information bottleneck to prevent visual biases. Specifically, given $(v, q, \hat{A})$, we first combine q and $\hat{a_k}$ into a statement s_k for each $\hat{a_k} \in \hat{A}$ using prompted LLM. In the case for the example in Figure 3, the following two statements will be generated: "The plane is landing", and "The plane is taking off". Then we give the statement to the LLM again and ask it to generate all the essential preconditions that can make this statement true inside an image. See the examples as in Figure 4. These preconditions serve as

For the statement about an image below, provide an as short as possible list of visual description sentences, which are hypothetical premises so that if these premises are true, the given statement will also be true. These premises must be verifiable through only visual clues from an image.

Statement: The plane is landing.
Premises:

---

1. The plane is descending towards the ground.
2. The landing gear of the airplane is extended.
3. The runway is visible in the background and getting closer.
4. The flaps on the wings are extended
5. The speed of the airplane is decreasing.
6. The runway lights are illuminated.

Figure 4: The prompt we use to generate dynamic preconditions through GPT model. The numbered premises below the line are GPT outputs for the example in Figure 2.

the basis for our reliable intermediate interpretable structures for any open-ended questions.

3.3 Visual Clue Generator

This step aims to develop a visual-language model to reason over the image and question and generate visual clues that help lead to an answer. For this purpose, we use a fine-tuned BLIP-2 model (Li et al., 2023) following similar settings as Dai et al. (2023). Specifically, we provide the Q-former module with the question to help it better extract image features and include the question again in input to the frozen LLM inside BLIP-2, as illustrated in Figure 5. The frozen LLM input includes image features and a prompt as "Question: $\{q\}$. Visual Clues: " during both training and inference.

Note that there are some differences between our visual clue generator and the original BLIP-2 model. When the original BLIP-2 model was trained for image captioning, each image with N gold captions will compose N input data with one single caption in each training input. In our case, we find that the generated clues trained this way always share high similarity and have same starting tokens. Instead, for a data with N gold visual clues, we follow previous works Klein et al. (2022); Chen et al. (2023) and compose N training inputs, each containing one unique permutation of concatenated visual clues. The BLIP-2 clue generation model is trained with standard language modeling loss to directly generate all the visual clues at the same time given an image and question. This strategy is used to encourage the diversity and comprehensiveness

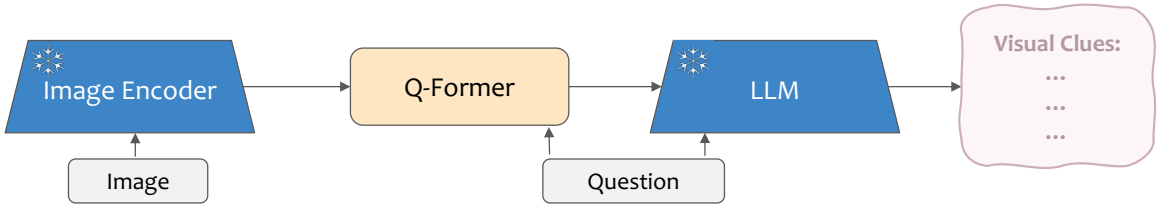

Figure 5: The modified Blip-2 model we use to generate visual clues given an image question pair.

Question: What shot is this player hitting?

Visual Clues:
- The player's hand is pointing towards the ball.
- The ball is straight above the player.

Question: Is this inside or outside?

Visual Clues:
- There is a tennis court.
- The ground is wet.

Figure 6: Examples where different questions for the same image should have different visual clues.

of generated clues.

3.4 Symbolic Answer Reasoning by NLI

With abductive proposals and visual clues generated, we construct our inference function f as a weighted average function constructed via natural language inference methods. As shown in the bottom part of Figure 2, we apply and see if a precondition can be satisfied by the visual clues, and keep the entailment score ES_c for precondition c. Specifically, we concatenate all visual clues together and either prompt a GPT model to rate how likelihood of the condition being satisfied, or similarly for a finetuned NLI model. Then we calculate the entailment score again to see whether the visual clues can infer the statements s generated from the question and answer proposals, and denote as ES_s. Since some preconditions can be unreliable or unnecessary while others can be always essential, we use w_s and w_c to represent the weight for entailment scores. Thus, the score for answer proposal a_k is simply the summation of entailment scores on all its preconditions and answer statement:

$$scores(a_k) = \sum_c w_c \cdot ES_c + w_s \cdot ES_s$$

where c is one dynamic precondition for a_k. Finally, we deduce the final answer $\hat{y}$ by selecting the answer proposal with highest score:

$$\hat{y} = \text{argmax}\left(scores(\hat{A})\right)$$

The NLI is done using a LLM model, and the prompt we use is as follows in Figures 7. In the experiments, we fix w_c to be 1.0 and w_s to be 1.5.

How likely is that "The airplane's nose is tilted upwards." if you are given the description about an image "The nose of the plane is pointed up, The tale of the plane is pointing down..."? Rate on a scale of 9. Give me the rate only.

Rate:

8

Figure 7: The prompt we use to calculate entailment scores through GPT model as an example for the question in Figure 2. Here, "The airplane's nose is tilted upwards." is one dynamic precondition.

4 Dataset Collection

Since there is no existing data ready for our proposed DCLUB system, we use Amazon Turk to collect 1.7K high-quality data for learning, evaluation, and analysis purposes, while focusing on questions that require reasoning besides simple recognition or perception following (Selvaraju et al., 2020). Given a question, image, and answer data pair, we ask the annotators to provide explicit visual clues that are entirely grounded in the image, and inferences that connect the visual clues to the answer.

Note that before the annotation, we first remove highly-ambiguous questions from our set through the first round of annotation, by asking "if there is significant direct evidence in the image that supports a different answer". The detailed annotation guidelines and an example data entry are shown in Appendix A.1.

4.1 Dataset Statistics

Dataset statistics can be found in Table 1. We try to have a more balanced set by maintaining a similar boolean question ratio (around 40%) as in the VQA V2 dataset.

Dataset	Train	Dev	Test	All
DCLUB	1,143	302	291	1,736

Table 1: Dataset statistics for DCLUB.

Data	Dev Acc.	Test Acc.
GPT-3.5		
question only	21.74	20.92
visual clues	62.69	64.02
visual clues + inferences	79.69	78.05
GPT-4		
question only	48.12	50.17
visual clues	82.75	82.82
visual clues + inferences	93.78	91.07

Table 2: An analysis on our annotated DCLUB data quality by using the gold visual clues and inferences to replace the image, and directly query GPT model for answer prediction.

4.2 Dataset Quality

As shown in Table 2, we test the annotated data quality using GPT models, by checking whether the GPT model can get the correct answer given the gold visual clues and inferences only, without the image. Specifically, we concatenate the gold clues, and gold inferences together, as shown in Figure 9. Note that we use 4-shot in-context learning examples retrieved from the training set according to question similarity by sentence bert. While the GPT-3.5 performance are not high enough, since the scores are approaching 100% for GPT-4, we believe the annotated data, especially the visual clues, are high quality.

5 Experiments

We evaluate on both of our annotated DCLUB dataset and the VQA v2 benchmark. However, since our system heavily depends on GPT models for intermediate precondition generations and is therefore limited by the GPT query speed, we randomly select a subset of size 300 from VQA v2 for computation efficiency concerns. This section includes VQA end-task results, answer proposal results, and visual clue generation results.

5.1 VQA Results

We report the end task performance calculated following the standard VQA evaluation metric in (Antol et al., 2015). The baseline black-box method we compare with is a fine-tuned BLIP-2 model on direct VQA with supervision from answers found in the DCLUB training set. We keep the pre-trained BLIP-2 checkpoint the same to be "pretrain_flant5xl", and the training data supervision as same amount. In Table 3, we demonstrate that by our dynamic clue bottleneck model achieves a comparable result with blackbox BLIP-2 model on both datasets.

Implementation Details We implement our visual clue generator based on the BLIP-2 model in LAVIS library (Li et al., 2023). In our experiments, we adopt the image encoder ViT-L/14 from CLIP (Radford et al., 2021)) and frozen LLM: FlanT5-XL (3B), by using the pre-trained checkpoints released from the original BLIP-2 paper. Training details including hyperparameters can be found in Appendix A.3. All models are trained utilizing NVIDIA RTX A6000 (48G) GPUs. The clue generation training can be completed within two days on average with single GPU resource.

6 Ablation Studies

6.1 Answer Proposals Results

We present two kinds of prompting method, with a fixed prompt as shown in Figure 3, and with few-shot in-context learning examples. For the latter, we first search for all the correct answers for each unique question, and then deploy sentence bert (Reimers and Gurevych, 2019) to calculate question similarity and select most similar unique training questions as examples.

We present a detailed experiment result Table 5 about answer proposal step in Section 3.1. We show that for GPT-based methods, adding in-context

System	Answer Proposals	NLI	Dev Acc.	Test Acc
Blackbox: zero-shot	-	-	65.23	66.78
Blackbox: fine-tuned	-	-	75.17	69.43
Ours	GPT-3.5	GPT-3.5	72.30	63.45
Ours	GPT-3.5	GPT-4	71.74	66.90
Ours	GPT-3.5	roberta	68.43	64.71
Ours	BLIP-2	GPT-3.5	**74.06**	**73.31**
Ours	BLIP-2	roberta	72.74	72.74

Table 3: VQA end-task performance comparisons between black-box end-to-end baseline model and our dynamic clue bottleneck model on DCLUB dataset. The roberta NLI here a roberta-based model fine-tuned for NLI tasks from (Nie et al., 2020). The blackbox model we use is BLIP-2 and details can be found in Section 5.

System	Answer Proposals	NLI	Acc.
Blackbox: zero-shot	-	-	48.67
Blackbox: fine-tuned	-	-	58.10
Ours	GPT-3.5	GPT-3.5	48.11
Ours	GPT-3.5	roberta	46.11
Ours	BLIP-2	GPT-3.5	**50.78**
Ours	BLIP-2	roberta	48.67

Table 4: VQA end-task performance comparisons between black-box end-to-end baseline model and our dynamic clue bottleneck model on the randomly selected subset of VQA V2 benchmark. The roberta NLI here a roberta-based model fine-tuned for NLI tasks from (Nie et al., 2020). The blackbox model we use is BLIP-2 and details can be found in Section 5.

learning example number cannot necessarily boost the recall rate, and that the fine-tuned BLIP-2 will always have a high recall rate while keeping a small average number of possible answers.

6.2 Visual Clue Generation Results

Since the training data size is only about 1.1k, we augmented our training set with examples from existing dataset (Selvaraju et al., 2020) where each image-question pair are equipped with sub-questions and sub-answers. We turn the sub-questions and sub-answers into statements, and serve as low-quality visual clues for the original data. We fine-tune our visual clue generation model following a two-stage training strategy: first on the large weak supervision set, then on DCLUB. We demonstrate a random select of generated clues examples from the DCLUB data dev set as in Figure 8, with an example full output in Figure 13. Overall clue quality is good, but sometimes the model struggles to avoid directly answer the question with a clue. Increasingly complex question benefit more from our bottleneck design.

Answer Proposal	Shot	Recall	# of proposals
DCLUB dataset			
BLIP-2	-	76.72	1.45
GPT-3.5	4	81.60	2.65
GPT-3.5	16	93.4	6.60
GPT-3.5	32	93.7	6.50
GPT-3.5 with visual clues	4	84.82	3.43
VQA v2 dataset			
BLIP-2	-	48.33	1.67
GPT-3.5	4	60.00	3.95
GPT-3.5 with visual clues	4	67.00	4.73

Table 5: Answer proposal recall comparisons between different methods discussed in Section 3.1 and 6.1. For DCLUB data, we put average numbers calculated over the dev and test sets.

7 Conclusion

In conclusion, we have presented DCLUB, an inherently interpretable VQA system that tackles the challenge of model interpretability in the multi-modal setting of vision and language. By breaking down the reasoning process into abduction propos-

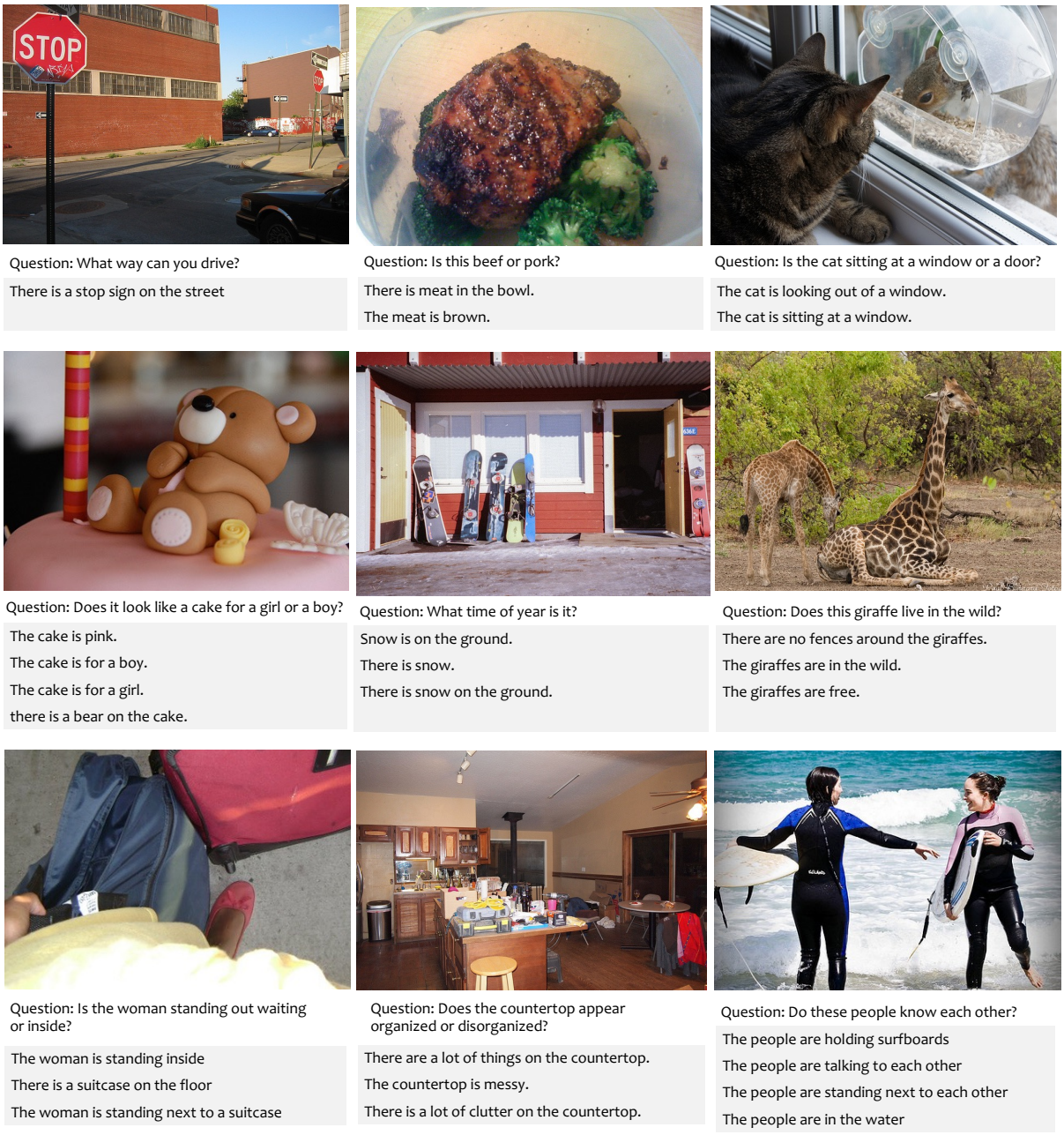

Figure 8: Examples of generated visual clues (in grey boxes) using DCLUB.

als, visual clues, and preconditions, our approach enables a deeper understanding of how the model arrives at a final answer while maintaining competitive performance compared with state-of-the-art black-box VQA systems. To facilitate the development and evaluation of our approach, we have collected a 1.7k dataset using Amazon Turk, focusing on questions that require deeper reasoning abilities. Our results demonstrate the effectiveness of our approach in providing interpretable, step-by-step rationales that can be trusted by humans.

This work represents a significant advancement in the field of visual question answering, as it addresses the pressing need for more transparent, trustworthy, and understandable AI systems. Our approach points toward a promising direction for implementing more reliable and interpretability-focused multi-modal AI models. By making AI more accessible and comprehensible to users, we ultimately enhance the potential for beneficial applications across a variety of real-world domains, such as healthcare, finance, and more.

Limitations

Our approach depends on extremely large OpenAI models, limiting how broadly such models could be adopted given the number of API calls required to create out bottleneck models. This may be cost

prohibitive for some, and bears a significant environmental burden. Cost limitations also prevented us from validating many natural choices for prompting, and different model sizes.

The VQA resources we evaluate on are limited to English. While the pretrained model in our approach may be able to extend beyond English given their pretraining data, we did not investigate such directions because our focus was primarily developing interpretability methods. Our main evaluation is end performance, and we assumed the bottlenecks we generate are fluent, without deep evaluation. Finally, our approach pipelines many large models, leaving the possibility that any one component failure on new data negatively impacts the performance of the rest.

Ethics Statement

As introduced in Section 4, we annotated the data using crowd-workers through Amazon Mechanical Turk. They are voluntary participants who were aware of any risks of harm associated with their participation. We require the workers to be located in either Australia, Canada, Great Britain or the United States such that they are English speakers. We also require the workers to have HIT Approval Rate (%) for all Requesters' HITs greater than or equal to 95%. All crowd-workers were compensated by a fair wage determined by estimating the average completing time of each annotation task. Each worker earn $7 per 10 queries and each query should take less than 2 minutes to annotate.

Finally, our methods rely on pretrained models that took may have had many negative environmental impacts during their training. During development of our models, we took care to avoid replicating such harms by needlessly retraining or finetuning models already available, although our continued dependence on large models may encourage new training runs of even larger models.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4971–4980.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning.

Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. 2017. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163:90–100.

Radhika Dua, Sai Srinivas Kancheti, and Vineeth N Balasubramanian. 2021. Beyond vqa: Generating multi-word answers and rationales to visual questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1623–1632.

Yu Feng, Ben Zhou, Haoyu Wang, Helen Jingshu Jin, and Dan Roth. 2023. Generic temporal reasoning with differential analysis and explanation. *ACL*.

Xingyu Fu, Sheng Zhang, Gukyeong Kwon, Pramuditha Perera, Henghui Zhu, Yuhao Zhang, Alexander Hanbo Li, William Yang Wang, Zhiguo Wang, Vittorio Castelli, Patrick Ng, Dan Roth, and Bing Xiang. 2023. Generate then Select: Open-ended Visual Question Answering Guided by World Knowledge. In *Findings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xingyu Fu, Ben Zhou, Ishaan Preetam Chandratreya, Carl Vondrick, and Dan Roth. 2022. There's a time and place for reasoning beyond the image. *ACL*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 3–19. Springer.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. 2018. Grounding visual explanations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 264–279.

Ruixin Hong, Hongming Zhang, Hong Zhao, Dong Yu, and Changshui Zhang. 2023. Faithful question answering with monte-carlo planning. *arXiv preprint arXiv:2305.02556*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Ayal Klein, Eran Hirsch, Ron Eliav, Valentina Pyatkin, Avi Caciularu, and Ido Dagan. 2022. QASem parsing: Text-to-text modeling of QA-based semantics. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7742–7756, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8779–8788.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *ArXiv*, abs/2210.01240.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 146–162. Springer.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Ramprasaath R Selvaraju, Purva Tendulkar, Devi Parikh, Eric Horvitz, Marco Tulio Ribeiro, Besmira Nushi, and Ece Kamar. 2020. Squinting at vqa models: Introspecting vqa models with sub-questions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10003–10011.

Kevin J Shih, Saurabh Singh, and Derek Hoiem. 2016. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4613–4621.

Xiaojuan Tang, Zilong Zheng, Jiaqi Li, Fanxu Meng, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. 2023. Large language models are in-context semantic reasoners rather than symbolic reasoners. *ArXiv*, abs/2305.14825.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed 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*, volume 35, pages 24824–24837. Curran Associates, Inc.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406. PMLR.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. 2023. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Ben Zhou, Kyle Richardson, Xiaodong Yu, and Dan Roth. 2022. Learning to decompose: Hypothetical question decomposition based on comparable texts. *EMNLP*.

A Appendix

A.1 Dataset Annotation Details

For every the crowd-source annotation, we offer \$0.7 for each single entry, which is equivalent to \$14 hour pay. The annotation guidelines are as shown in Figures 10 and 11. An example annotated data entry is shown as in Figure 12. And FIgure 9 denotes an example prompt we use to evaluate dataset quality with LLMs.

A.2 DCLUB Example Output

An example output of our DCLUB system is illustrated in Figure 13.

A.3 Blip-2 Hyperparameters

When fine-tuning on the large weak supervision dataset, we apply a linear warmup of the learning rate during the initial 3K steps, increasing from $1e{-}8$ to $1e{-}6$, followed by a cosine decay with a minimum learning rate of $1e{-}8$.

Please answer the questions.

Question: Is the airplane landing or taking off?
Clues: 1: The nose of the plane is pointed up. 2: The tale of the plane is pointing down. 3: There is a lot of runway behind the plane. 4: The plane is moving away from the ground.
Reasonings: 1. It is pointing towards the sky trying to reach it. 2 : Because it has yet to get stabilized away from the ground as it moves towards the sky. 3 : It just finished taxing as it took off. 4 : Instead of towards it as it would with a landing.
Answer:

The airplane is taking off.

Figure 9: Example prompt we use to query the GPT model for direct visual question answering, for the example data entry in Figure 12. We hide the 4-shot in-context examples here for space concerns. The answer below the line is GPT output.

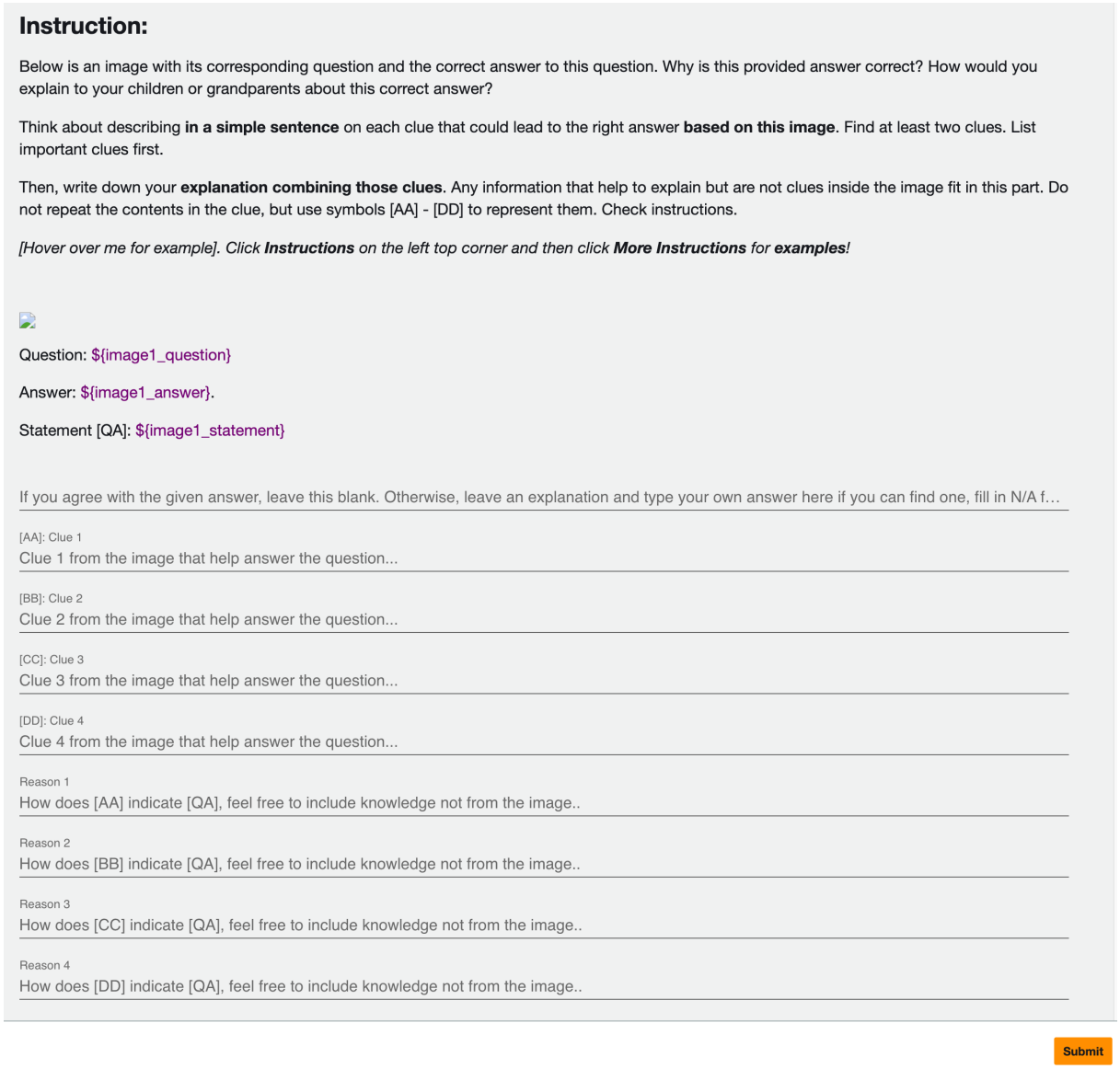

Figure 10: The annotation guidelines we give to mturk workers.

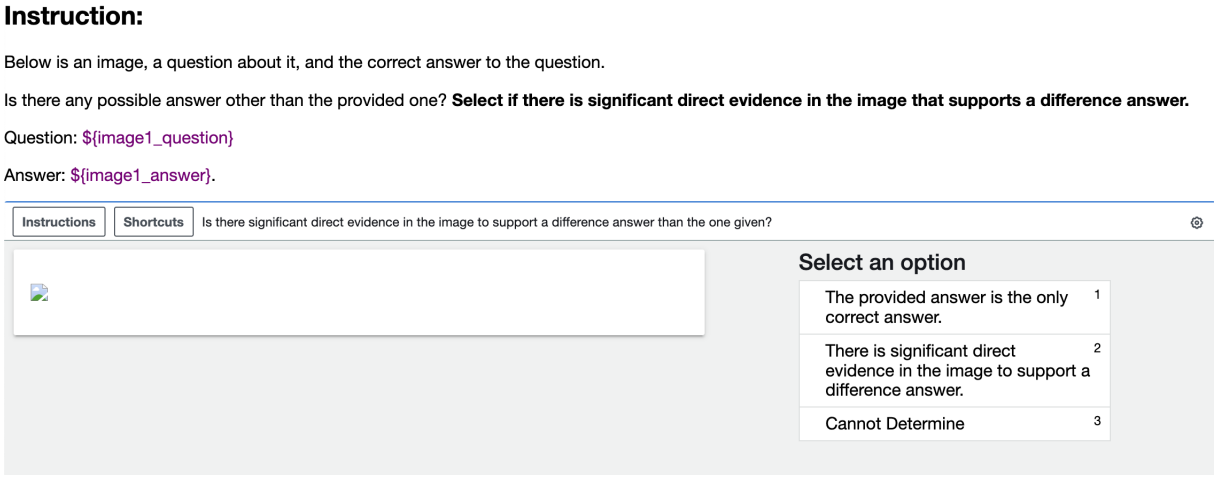

Figure 11: The annotation guidelines we give to mturk workers for filtering out ambiguous ones.

Question: Is the airplane landing or taking off?

Gold answer: Taking off

Clue1: The nose of the plane is pointed up.

→ **Reason1** : It is pointing towards the sky trying to reach it.

Clue2: The tale of the plane is pointing down.

→ **Reason2** : Because it has yet to get stabilized away from the ground as it moves towards the sky.

Clue3: There is a lot of runway behind the plane.

→ **Reason3** : It just finished taxing as it took off.

Clue4: The plane is moving away from the ground.

→ **Reason4** : Instead of towards it as it would with a landing.

Figure 12: An example annotated data entry.

Question: Is this a modern plane?

clues1: The plane is blue.

clues2: The plane is not modern.

clues3: The plane has propellers.

clues4: The plane is old.

yes

Preconditions	scores
The plane has sleek, aerodynamic lines.	2.0
The design includes jet engines.	1.0
The cockpit has a large, curved windshield.	9.0
The wings are swept back.	9.0
The tail section is angled upwards.	9.0
This is a modern plane.	1.0
average	5.25

no

Preconditions	scores
The plane has propellers instead of jet engines.	9.0
The windows have a round shape instead of being rectangular.	1.0
The cockpit has a more vintage design.	8.0
The plane has a single tail fin instead of a double one.	9.0
The color scheme and logo suggest an older era of aviation.	8.0
The plane lacks modern features such as winglets or advanced avionics.	9.0
This is not a modern plane.	9.0
average	8.21

Predicted answer: no

Figure 13: Examples output of DCLUB.

Clue Generation Stage 1	
LLM	FlanT5 XL
Fine-tuning epochs	5
Warmup steps	3081
Learning rate	$5e-6$
Weight decay	0.01
Image resolution	224
Prompt	"Question: {} Related Clues:"
Inference beam size	5
Clue Generation Stage 2	
LLM	FlanT5 XL
Fine-tuning epochs	3
Warmup steps	255
Learning rate	$5e-8$
Weight decay	0.01
Image resolution	224
Prompt	"Question: {} Related Clues:"
Inference beam size	5
Answer Proposal	
LLM	FlanT5 XL
Fine-tuning epochs	5
Warmup steps	1000
Learning rate	$1e-5$
Weight decay	0.05
Image resolution	224
Prompt	"Question: {} Possible Answers:"
Inference beam size	5
Baseline VQA	
LLM	FlanT5 XL
Fine-tuning epochs	5
Warmup steps	1000
Learning rate	$1e-5$
Weight decay	0.05
Image resolution	224
Prompt	"Question: {} Answer:"
Inference beam size	5

Table 6: Hyperparameters for fine-tuning BLIP-2 (continued).