# CLEVR-Dialog: A Diagnostic Dataset for Multi-Round Reasoning in Visual Dialog

**Satwik Kottur**[1], **José M.F. Moura**[1], **Devi Parikh**[2,3], **Dhruv Batra**[2,3], **Marcus Rohrbach**[2]
[1]Carnegie Mellon University, [2]Facebook AI Research, [3]Georgia Institute of Technology
{skottur,moura}@andrew.cmu.edu,{parikh,dbatra}@gatech.edu,mrf@fb.com

## Abstract

Visual Dialog is a multimodal task of answering a sequence of questions grounded in an image (using the conversation history as context). It entails challenges in vision, language, reasoning, and grounding. However, studying these subtasks in isolation on large, real datasets is infeasible as it requires prohibitively-expensive complete annotation of the 'state' of all images and dialogs.

We develop CLEVR-Dialog, a large diagnostic dataset for studying multi-round reasoning in visual dialog. Specifically, we construct a *dialog grammar* that is grounded in the scene graphs of the images from the CLEVR dataset. This combination results in a dataset where all aspects of the visual dialog are fully annotated. In total, CLEVR-Dialog contains 5 instances of 10-round dialogs for about 85$k$ CLEVR images, totaling to 4.25$M$ question-answer pairs.

We use CLEVR-Dialog to benchmark performance of standard visual dialog models; in particular, on *visual coreference resolution* (as a function of the coreference distance). This is the first analysis of its kind for visual dialog models that was not possible without this dataset. We hope the findings from CLEVR-Dialog will help inform the development of future models for visual dialog. Our code and dataset are publicly available[1].

## 1 Introduction

The focus of this work is on intelligent systems that can *see* (perceive their surroundings through vision), *talk* (hold a visually grounded dialog), and *reason* (store entities in memory as a dialog progresses, refer back to them as appropriate, count, compare, *etc*.). Recent works have begun studying such systems under the umbrella of *Visual Dialog* (Das et al., 2017a; de Vries et al., 2017), where an agent must answer a *sequence* of questions grounded in an image. As seen in Fig. 1, this entails challenges in – vision (*e.g*., identifying objects and their attributes in the image), language/reasoning (*e.g*., keeping track of and referencing previous conversation via memory), and grounding (*e.g*., grounding textual entities in the image).

In order to train and evaluate agents for Visual Dialog, Das et al. (2017a) collected a large dataset of human-human dialog on real images collected between pairs of workers on Amazon Mechanical Turk (AMT). While such large-scale realistic datasets enable new lines of research, it is difficult to study the different challenges (vision, language, reasoning, grounding) in isolation or to break down the performance of systems over different challenges to identify bottlenecks, because that would require prohibitively-expensive complete annotation of the 'state' of all images and dialogs (all entities, coreferences, *etc*.).

In this work, we draw inspiration from Johnson et al. (2017), and develop a large diagnostic dataset—CLEVR-Dialog—for studying and benchmarking multi-round reasoning in visually-grounded dialog. Each CLEVR image is synthetically rendered by a particular scene graph (Johnson et al., 2017) and thus, is by construction exhaustively annotated. We construct a *dialog grammar* that is grounded in these scene graphs. Specifically, similar to Das et al. (2017b), we view dialog generation as communication between an Answerer (A-er) who can 'see' the image and has the complete scene graph (say $S_a$), and a Questioner (Q-er), who does not 'see' the image and is trying to reconstruct the scene graph over rounds of dialog (say $S_q^t$). As illustrated in Fig. 1, the dialog begins by A-er providing a grounded caption for the image, which conveys some but not all information about $S_a$. The Q-er builds a partial scene graph $S_q^0$ based on the caption, and follows up by asking questions

[1]https://github.com/satwikkottur/clevr-dialog



*Proceedings of NAACL-HLT 2019*, pages 582–595
Minneapolis, Minnesota, June 2 - June 7, 2019. 

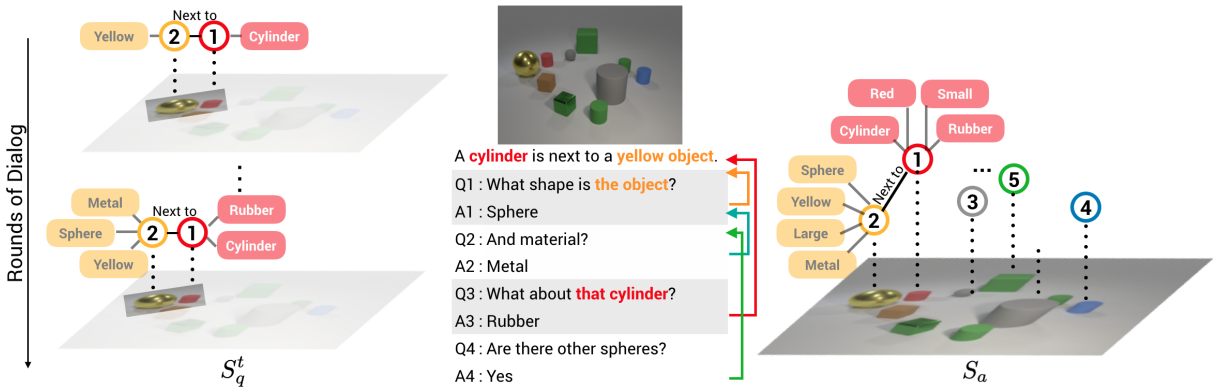



Figure 1: CLEVR-Dialog: we view dialog generation as communication between an Answerer (A-er) who can 'see' the image $I$ and has the complete scene graph $S_a$ (far right), and a Questioner (Q-er), who does not 'see' the image. A-er begins the dialog with a grounded caption (*'A cylinder is next to a yellow object'*). The Q-er converts this caption into a partial scene graph $S_q^0$ (far left, top), follows up with a question grounded in $S_q^0$ (*'What shape is the object?'*), which the A-er answers, and the dialog progresses. Questions at round $t$ are generated based solely on $S_q^t$, *i.e.*, without looking at I or $S_a$, which mimics real-life scenarios of visual dialog. Note that while studying visual dialog on CLEVR-Dialog, models are forced to answer questions with just the image and dialog history as additional inputs, and do not have access to $S_a$.

grounded in $S_q^0$, which the A-er answers, and the dialog progresses. Our dialog grammar defines rules and templates for constructing this grounded dialog. Note that A-er with access to $S_a$ (perfect vision) exists **only** during dialog generation to obtain ground truth answers. While studying visual dialog on CLEVR-Dialog, models are forced to answer questions with just the image and dialog history as additional inputs.

In total, CLEVR-Dialog contains 5 instances of 10-round dialogs for each of $70k$ (train) and $15k$ (val) CLEVR images, totaling to $3.5M$ (train) and $0.75M$ (val) question-answer pairs. We benchmark several visual dialog models on CLEVR-Dialog as strong baselines for future work.

The combination of CLEVR images (with full scene graph annotations) and our dialog grammar results in a dataset where all aspects of the visual dialog are fully annotated. We use this to study one particularly difficult challenge in multi-dialog visual reasoning – of *visual coreference resolution*. A coreference arises when two or more phrases (*coreferring phrases*) in the conversation refer to the same entity (*referent*) in the image. For instance, in the question *'What about that cylinder?'* (Q3) from Fig. 1, the referent for the phrase *'that cylinder'* can be inferred only after resolving the phrase correctly based on the dialog history, as there are multiple cylinders in the image. We use CLEVR-Dialog to diagnose performance of different methods as a function of the history dependency (*e.g.*, coreference distance—the number of rounds between successive mentions of the same object) and find that the performance of a state-of-art model (CorefNMN) is at least 30 points inferior for questions involving coreference resolution compared to those which do not (Fig. 5), highlighting the challenging nature of our dataset. This is the first analysis of its kind for visual dialog that was simply not possible without this dataset. We hope the findings from CLEVR-Dialog will help inform the development of future models for visual dialog.

## 2 Related Work

**Coreference Resolution** is a well studied problem in the NLP community (Ng, 2010; Wiseman et al., 2016; Lee et al., 2017; Clark and Manning, 2016a,b). Our work focuses on *visual* coreference resolution – the referent is now a visual entity to be grounded in visual data. Several works have tackled visual coreference resolution in videos (Ramanathan et al., 2014; Rohrbach et al., 2017) and 3D data (Kong et al., 2014), and have introduced real image datasets for the same (Hodosh et al., 2014).

**Visual Dialog and Synthetic Datasets.** We contrast CLEVR-Dialog against four existing datasets: (1) **CLEVR** (Johnson et al., 2017) is a diagnostic dataset for visual question answering (VQA) (Antol et al., 2015) on rendered images that contain objects like cylinders, cubes, *etc.*, against a plain background (Fig. 1). While CLEVR-Dialog uses the same set of images, the key difference is that of focus and emphasis – the objective of CLEVR-

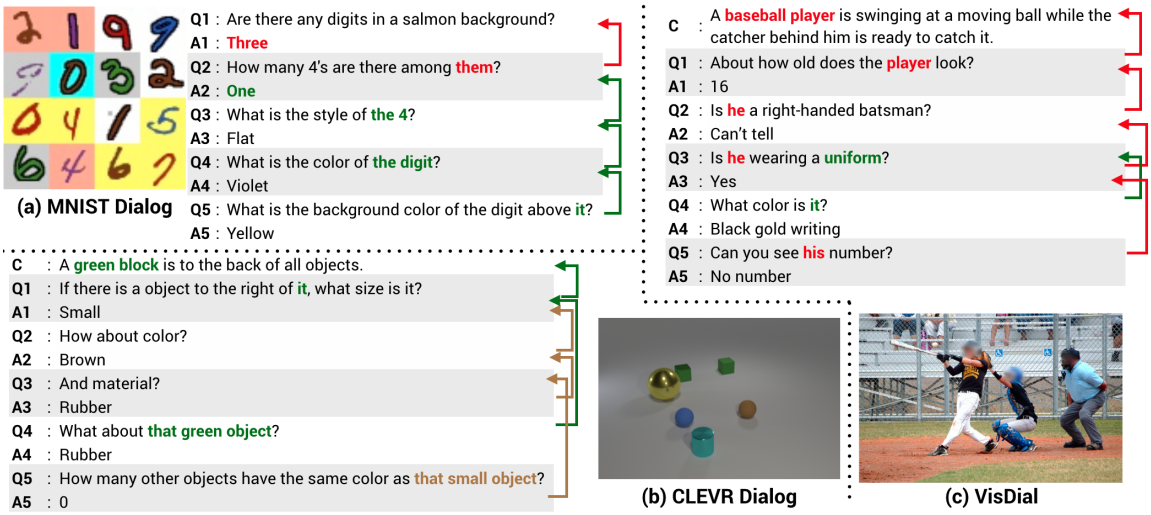



Figure 2: Example dialogs from MNIST Dialog, CLEVR-Dialog, and VisDial, with coreference chains manually marked for VisDial and automatically extracted for MNIST Dialog and CLEVR-Dialog.

VQA questions is to stress-test spatial reasoning in independent single-shot question answering; the objective of CLEVR-Dialog is to stress-test temporal or multi-round reasoning over the dialog history. (2) **CLEVR-Ref+** (Liu et al., 2019) is a diagnostic dataset based on CLEVR images for visual reasoning in referring expressions. CLEVR-Dialog goes beyond CLEVR-Ref+, which focuses on grounding objects given a natural language expression, and deals with additional visual and linguistic challenges that require multi-round reasoning in visual dialog. (3) **MNIST-Dialog** (Seo et al., 2017) is a synthetic dialog dataset on a grid of $4 \times 4$ stylized MNIST digits (Fig. 2). While MNIST-Dialog is similar in spirit to CLEVR-Dialog, key difference is complexity – the distance between a coreferring phrase and its antecedent is always 1 in MNIST-Dialog; in contrast, CLEVR-Dialog has a distribution ranging from 1 to 10. (4) **VisDial** (Das et al., 2017a) is a large scale visual dialog dataset collected by pairing two human annotators (a Q-er and an A-er) on AMT, built on COCO (Lin et al., 2014) images. VisDial being a large open-ended real dataset encompasses all the challenges of visual dialog, making it difficult to study and benchmark progress on individual challenges in isolation. Fig. 2 qualitatively compares MNIST-Dialog, CLEVR-Dialog, and VisDial, and shows coreference chains (manually annotated for this VisDial example, and automatically computed for MNIST-Dialog and CLEVR-Dialog). We can see that the chains in MNIST-Dialog are the simplest (distance always 1). While coreferences in VisDial can be on a similar level of difficulty than CLEVR-Dialog, the difficult cases are rarer in VisDial.

| Name | CLEVR Dialog (ours) | MNIST Dialog | VisDial |
|---|---|---|---|
| # Images | 85*k* | 50*k* | 123*k* |
| # Dialogs | 425*k* | 150*k* | 123*k* |
| # Questions | 4.25*M* | 1.5*M* | 1.2*M* |
| # Unique Q | 73*k* | 355 | 380*k* |
| # Unique A | 29 | 38 | 340*k* |
| Vocab. Size | 125 | 54 | 7.*k* |
| Mean Q Len. | 10.6 | 8.9 | 5.1 |
| Mean Coref Dist. | 3.2 | 1.0 | - |

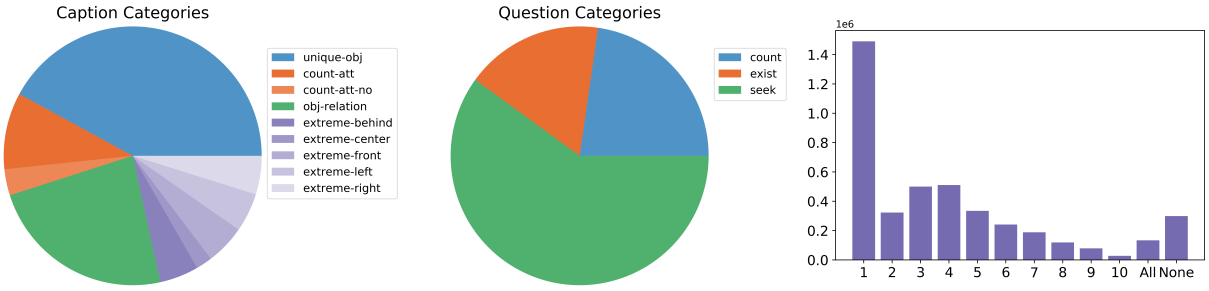
Table 1: Dataset statistics comparing CLEVR-Dialog to MNIST Dialog (Seo et al., 2017). Our dataset has $3\times$ the questions (larger), $206\times$ the unique number of questions (more diverse), $3.2\times$ the mean coreference distance (more complex), and longer question lengths. Similar stats for VisDial are also shown. Coreference distance can not be computed for VisDial due to lack of annotations.

## 3 CLEVR-Dialog Dataset

In this section, we describe the existing annotation for CLEVR images, then detail the generation process for CLEVR-Dialog, and present the dataset statistics in comparison to existing datasets.

**Setup.** Every CLEVR image $I$ has a full scene graph annotation, $S_a$. This contains information about all the objects in the scene, including four major attributes $\{$*color, shape, material, size*$\}$, 2D image and 3D world positions, and relationships $\{$*front, back, right, left*$\}$ between these objects. We only use objects, attributes, and relationships.

**Dialog Grammar.** An important characteristic of visual dialog that makes it suitable for practical applications is that the questioner does not 'see'




(a) Distribution of caption (left) and question (right) categories.

(b) Distribution of coreference distances.

Figure 3: Distribution of caption and question categories, and history dependency in CLEVR-Dialog dataset.

the image (because if it did, it would not *need* to ask questions). To mimic this setup, we condition our question generation at round $t$ only on the partial scene graph $S_q^t$ that accumulates information received so far from the dialog history (and not on $S_a$). Specifically, we use a set of caption $\{T_i^C\}$ and question $\{T_i^Q\}$ templates, which serve as the structural units of our dialog grammar. The role of the caption is to *seed* the dialog and initialize $S_q^0$. Each of the question templates is accompanied by a set of constraints on $S_q^t$, which decide if a particular template can be selected at the current round. For instance, a question *'What shape is the blue object?'* can be only be asked (generated) if the dialog so far has already *mentioned* a 'blue object', *i.e.*, only if $S_q^t$ contains a (unique) 'blue object'. The nature and difficulty of the dataset is highly dependent on these templates, thus making their selection crucial.

To this end, we carefully design four categories of caption templates: (a) `Obj-unique` mentions an object with unique set of attributes in the image, (b) `Obj-count` specifies the presence of a group of objects with common attributes, (c) `Obj-extreme` describes an object at one of the positional extremes of the image (right, left, fore, rear, center), (d) `Obj-relation` talks about the relationship between two objects along with their attributes in a way that allows them to be uniquely identified in the complete scene graph $S_a$.

For the questions, we experiment with three different categories: (a) **Count** questions ask for a count of objects in the image satisfying specific conditions, *e.g.*, *'How many objects share the same color as this one?'*, (b) **Existence** questions are yes/no binary questions that verify conditions in the image, *e.g.*, *'Are there any other cubes?'*, and (c) **Seek** questions query attributes of objects, *e.g.*, *'What color is that cylinder?'*. Note that CLEVR-Dialog represents not just a static dataset but also a recipe for constructing increasingly challenging grounded dialog by expanding this grammar. Refer to the appendix for further details.

**Dialog Generation.** At a high level, dialog generation now 'simply' involves selecting a sequence of templates such that the accompanying constraints are satisfied by $S_q^t$ at all $t$. As a tractable approximation to this exponentially-large constraint satisfaction problem, we use beam search that finds a valid solution *and* enforces additional conditions to make the dialog *interesting* (see Fig. 4). At every round of the dialog (after 3 rounds), we ensure that each of the question template types—count, existence, and seek—falls within a range ($10\% - 20\%$ for count/existence each, and $30\% - 60\%$ for seek). In addition, we identify *independent* questions that do not need history to answer them, *e.g.*, *'How many objects are present in the image?'*, and limit their number to under $10\%$. We found this to be effective both in terms of speed and dialog diversity. Fig. 4 illustrates the diverse set of candidate questions generated at each round for a given image.

**Dataset Statistics.** We compare CLEVR-Dialog to MNIST-Dialog and VisDial in Tab. 1, but the key measure of coreference distance cannot be reported for VisDial as it is not annotated. Overall, CLEVR-Dialog has $3\times$ the questions and a striking $206\times$ the unique number of questions than MNIST-Dialog , indicating higher linguistic diversity. CLEVR-Dialog questions are longer with a mean length of 10.6 compared to 8.9 for MNIST-Dialog. Crucially, supporting our motivation, the mean distance (in terms of rounds) between the coreferring expressions in CLEVR-Dialog is $3.2\times$ compared to 1.0 in MNIST-Dialog. Moreover, the distances (see Fig. 3b) in CLEVR-Dialog vary (min of 1, max of 10), while it is constant (at 1) in MNIST-Dialog, making it easy for models to pick

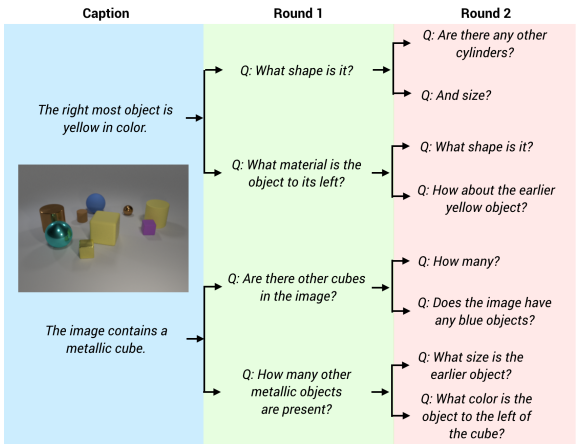



Figure 4: Dialog generation in CLEVR-Dialog. At each round, all valid question templates are used to generate candidates for the next question. However, only a few *interesting* candidates (beams) are retained for further generation, thus avoiding an exploding number of possibilities as rounds of dialog progress.

up on this bias. The distribution of caption and question templates is given in Fig. 3a. See appendix for further analysis.

## 4 Experiments

**Baselines.** To benchmark performance, we evaluate several models on CLEVR-Dialog. **Random** picks an answer at random. **Random-Q** picks an answer at random among valid answers for a given question type (*e.g.*, name of a color for color questions). Further, we adapt the discriminative visual dialog models from Das et al. (2017a): (a) **L**ate **F**usion (**LF**) that models separately encode each of question (Q), history (H), and image (I); and then fuse them by concatenation. (b) **H**ierarchical **R**ecurrent **E**ncoder (**HRE**) that models dialog via both dialog-level and sentence-level recurrent neural networks. (c) **M**emory **N**etwork (**MN**) that stores history as memory units and retrieves them based on the current question. We also consider neural modular architectures: (a) **CorefNMN** (Kottur et al., 2018) that explicitly models coreferences in visual dialog by identifying the reference in the question (textual grounding) and then localizing the referent in the image (visual grounding), (b) **NMN** (Hu et al., 2017), which is a history-agnostic ablation of CorefNMN.

**Results.** We use multi-class classification accuracy for evaluation since CLEVR-Dialog has one-word answers. Tab. 2 shows the performance of different models. The key observations are: (a) Neural models outperform random baselines by a large margin.

| Model | Acc. |
|---|---|
| Random | 3.4 |
| Random-Q | 33.4 |
| LF-Q | 40.3 |
| LF-QI | 50.4 |
| LF-QH | 44.1 |
| LF-QIH | 55.9 |
| HRE-QH | 45.9 |
| HRE-QIH | 63.3 |
| MN-QH | 44.2 |
| MN-QIH | 59.6 |
| NMN | 56.6 |
| CorefNMN | **68.0** |

Table 2: Accuracy (%) on CLEVR-Dialog (higher is better). See text for details.

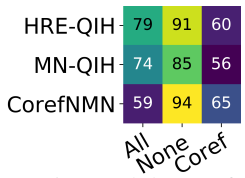



Figure 5: Breakdown of performance by questions that depend on entire history (*All*), require coreference resolution (*Coref*), and are history-independent (*None*).

The best performing model, CorefNMN, outperforms Random-Q by 35%. (b) History-agnostic models (LF-Q, LF-QI, NMN) also suffer in performance, highlighting the importance of history. (c) Finally, we break down the performance of top-3 models on questions which depend on entire history (*All*), require coreference resolution (*Coref*), and are history-independent (*None*), in Fig. 5. We find that CorefNMN is 30% worse on *Coref* than *None* questions, signifying the complexity of CLEVR-Dialog as the former are qualitatively harder to answer than the latter. (d) More interestingly, HRE-QIH, though inferior to CorefNMN on *Coref*, outperforms the latter on *All* questions (*'How many other objects?'*) by around 20%. A possible explanation is that the former, owing to its dialog-level RNN, captures global summaries more efficiently than the latter. This is the first analysis of its kind for visual dialog that was simply not possible without this dataset. Appendix provides a further analysis of model performances.

**Conclusion.** We proposed a large, synthetic dataset called CLEVR-Dialog, to study multi-round reasoning in visual dialog, and in particular the challenge of visual coreference resolution. We benchmarked several qualitatively different models from prior work on this dataset, which act as baselines for future work. Our dataset opens the door to evaluate how well models do on visual coreference resolution, without the need to collect expensive annotations on real datasets.

## References

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 (ICCV)*.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653. Association for Computational Linguistics.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. 2017a. Visual Dialog. In *CVPR*.

Abhishek Das, Satwik Kottur, Jos M. F. Moura, Stefan Lee, and Dhruv Batra. 2017b. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peter Hodosh, Alice Young, Micah Lai, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. 2014. What are you talking about? text-to-image coreference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Satwik Kottur, Jose M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. 2018. Visual coreference resolution in visual dialog using neural module networks. In *The European Conference on Computer Vision (ECCV)*.

Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. 2018. Referring relationships. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Runtao Liu, Chenxi Liu, Yutong Bai, and Alan Yuille. 2019. Clevr-ref+: Diagnosing visual reasoning with referring expressions. *arXiv preprint arXiv:1901.00850*.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1396–1411, Stroudsburg, PA, USA. Association for Computational Linguistics.

V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. 2014. Linking people with "their" names using coreference resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. 2017. Generating descriptions with grounded and co-referenced people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. 2017. Visual reference resolution using attention memory for visual dialog. In *Advances in Neural Information Processing Systems (NIPS)*.

Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004. Association for Computational Linguistics.

## Appendix

The appendix is organized as follows:

- We begin with the description of CLEVR images in Sec. A,
- Sec. B describes further details of the dialog generation,
- Sec. C provides additional statistical analysis for CLEVR-Dialog,
- Diagnostic model performance analysis in given in Sec. D, and finally
- Implementation details can be found in Sec. E.

## A CLEVR Images

First introduced by Johnson et al. (2017), CLEVR images are synthetically rendered, and contain several objects spatial located on a plain background. These objects have four different attributes: (a) *Shape*—cylinder, cube, sphere; (b) *Color*—blue, brown, cyan, gray, green, purple, red, yellow; (c) *Size*—large and small; and finally (d) *Material*—metal and rubber.

## B Generating CLEVR-Dialog Dataset

As noted in the main paper, an important characteristic of visual dialog that makes it suitable for practical applications is that the questioner does not 'see' the image (because if it did, it would not *need* to ask questions). To mimic this setup, we condition our question generation at round $t$ only on the partial scene graph $S_q^t$ that accumulates information received so far from the dialog history (and not on $S_a$). Specifically, we use a set of caption $\{T_i^C\}$ and question $\{T_i^Q\}$ templates (enumerated in Tab. 3), which serve as the basis for our dialog generation. Each of these templates in turn consists of primitives, composed together according to a generation grammar. In what follows, we will first describe these primitives, discuss how they are used to generate a caption or a question at each round, and tie everything together to explain dialog generation in CLEVR-Dialog.

**Grammar Primitives.** The templates used to generate captions and questions are composed of intuitive and atomic operations called primitives. Each of these primitives can have different instantiations depending on a parameter, and also take input arguments. For example, all `Filter` primitives filter out objects from an input set of objects according to certain constraints. In particular, `Filter[color](blue)` filters out blue objects from a given set of objects, while `Filter[shape](sphere)` filters out all spheres. In our work, we use the following primitives:

- `Sample`: sample an object/attribute,
- `Unique`: identify unique objects/attributes,
- `Count`: count the number of input objects,
- `Group`: group objects based on attribute(s),
- `Filter`: filter inputs according to a constraint,
- `Exist`: check for existence of objects,
- `Relate`: apply a relation (*e.g.*, *right of*).

Note that each of these primitives inherently denotes a set of constraints, which when failed leads to a reset of the generation process for the current caption/question in the dialog. For example, if the output of `Filter[color](blue)` is empty due to an absence of blue objects in the input, we abort generation for the current template and move on to the next template.

**Caption Generation.** The role of the caption is to *seed* the dialog and initialize $S_q^0$. In other words, caption gives Q-er partial information about the image so that asking follow-up questions is possible. Because A-er generates the caption, it uses the full scene graph $S_a$. Fig. 6 shows the caption grammar in action, producing three different captions for a given image. Consider the grammar for Fig. 6(c). First, `Sample[attribute]` produces {*shape, color*} used by `Unique` to select objects from $S_a$ with unique shape and color attributes. An object (gray cylinder) is then sampled from these using `Sample[object]`. Next, a relation (*in front of*) is enforced via a `Relate` primitive leading to the green cylinder in front of the gray cylinder. Finally, `Sample[attribute]` samples one of the attributes to give us the caption, *'A green object stands in front of a gray cylinder.'*

**Question Generation.** Unlike the caption, the questions are generated by the Q-er, having access only to a partial scene graph $S_q^t$ at round $t$. This $S_q^t$ is an assimilation of information from the previous rounds of the dialog. The primitives in the question template therefore take $S_q^t$ as the input scene graph, and the generation proceeds in a manner similar to that of the caption explained above. As the dialog is driven by Q-er based on partial scene information, only a few questions are non-redundant (or even plausible) at a given round of the dialog. To this

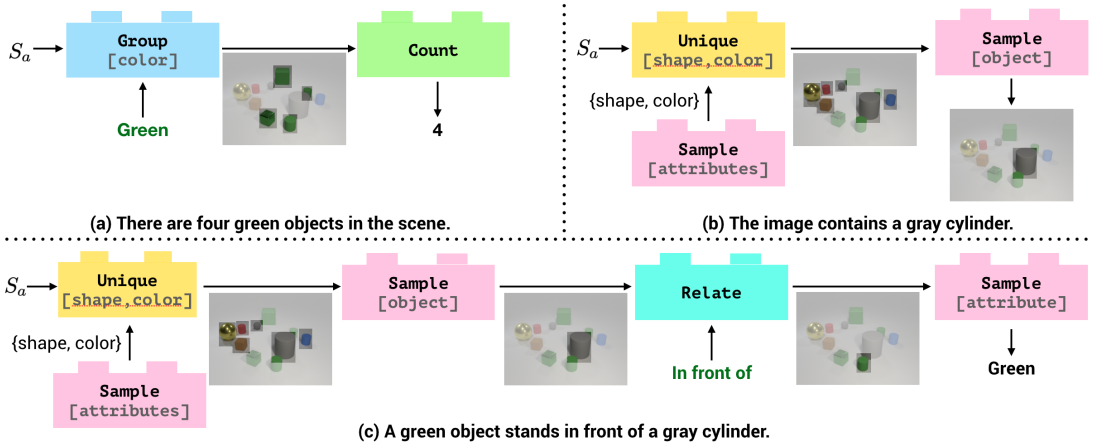



Figure 6: Usage of dialog grammar in caption generation. See text for details.

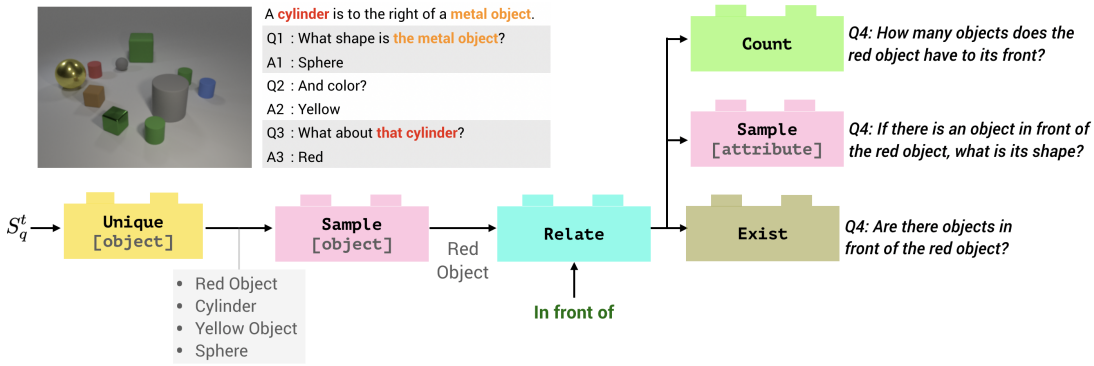



Figure 7: Usage of dialog grammar in question generation. See text for details.

end, the inherent constraints associated with the primitives now play a bigger role in the template selection.

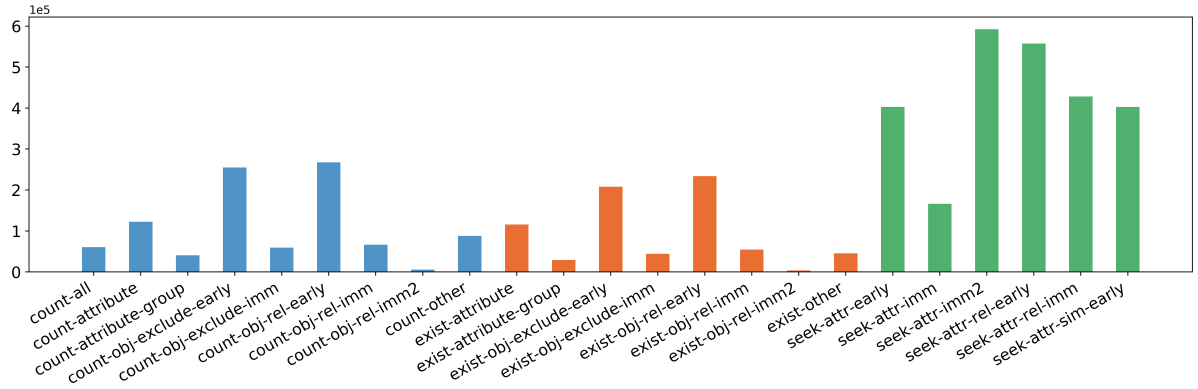
Consider Fig. 7 that shows how the current question is generated using the primitives and grammar, given the caption and dialog history (question-answer pair for the first three rounds). For the current round, the question *'What material is the green object at the back?'* is clearly implausible (Q-er is unaware of the existence of a green object), while the question *'What shape is the red object?'* is redundant. For the templates visualized, `Unique[object]` returns a list of unique known object-attribute pairs (using $S_q^t$). A candidate is sampled by `Sample[object]` and a relation is applied through `Relate(in front of)`. There are multiple choices at this junction: (a) The use of `Count` leads to a counting question (`count-obj-rel-early`), (b) Invoking `Sample[attribute]` results in a seek question (`seek-attr-rel-early`), and finally, (c) `Exist` primitive generates an exist question of type `exist-obj-rel-early`.

**Dialog Generation.** As specified in the main paper, we use beam search as a more tractable alternative to search through the exponential space of possible dialogs, by using additional constrains to retain only *interesting* dialogs. At every round of the dialog (after 3 rounds), we ensure that each of the question template types—count, existence, and seek—falls within a range (10% – 20% for count/existence each, and 30% – 60% for seek). In addition, we identify *independent* questions that do not need history to answer them, *e.g.*, *'How many objects are present in the image?'*, and limit their number to under 10%. Finally, to encourage questions that require reasoning over the history, *e.g.*, `seek-attr-sim-early` and `count-obj-excl-imm`, we tailor our beam search objective so that dialogs containing such questions have a higher value. We use a beam search with 100 beams for each dialog. Fig. 4 illustrates the diverse set of candidate questions generated at each round for a given image.

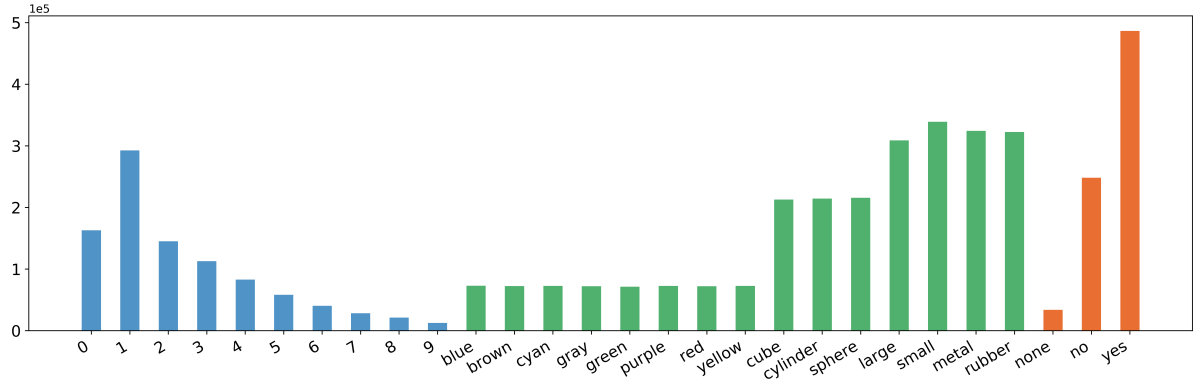
To summarize, the usage of primitives and a dialog grammar makes our generation procedure: (a) modular: each primitive has an intuitive meaning,

| | |
|---|---|
| **Captions** | |
| obj-relation | *'A [Z] [C] [M] [S] stands [R] a [Z1] [C1] [M1] [S1].'*<br>*'A gray sphere stands to the right of a red object.'* |
| obj-unique | *'A [Z] [C] [M] [S] is present in the image.'*<br>*'A red object is present in the image'* |
| obj-extreme | *'The rightmost thing in the view is a [Z] [C] [M] [S].'*<br>*'The rightmost thing in the view is a cylinder.'* |
| obj-count | *'The image has [X] [Z] [C] [M] [S].'*<br>*'The image has four cylinders.'* |
| **Count/Exist Question Type** | |
| count-all | *'How many objects in the image?'* |
| count/ | *'[How many \| Are there] other [Z] [C] [M] [S] in the picture?'* |
| exist-excl | *'[How many \| Are there] other cubes in the picture?'* |
| count/ | *'[If present, how many \| Are there] [Z] [C] [M] [S] objects?'* |
| exist-attr | *'[If present, how many \| Are there] metallic objects?'* |
| count/ | *'[How many \| Are there] [Z] [C] [M] [S] among them?'* |
| exist-attr-group | *'[How many \| Are there] blue cylinders among them?'* |
| count/ | *'[How many \| Are there] things to its [R]?'* |
| exist-obj-rel-imm | *'[How many \| Are there] things to its right?'* |
| count/ | *'How about to its [R]?'* |
| exist-obj-rel-imm2 | *'How about to its left?'* |
| count/ | *'[How many \| Are there] things [R] that [Z] [C] [M] [S]?'* |
| exist-obj-rel-early | *'[How many \| Are there] things in front of that shiny object?'* |
| count/ | *'[How many \| Are there] things that share its [A]?'* |
| exist-obj-excl-imm | *'[How many \| Are there] things that share its color?'* |
| count/ | *'[How many \| Are there] things that are the same [A] as that [Z] [C] [M] [S]?'* |
| exist-obj-excl-early | *'[How many \| Are there] things that are the same size as that round object?'* |
| **Seek Question Type** | |
| seek-attr-imm | *'What is its [A]?'*<br>*'What is its shape?'* |
| seek-attr-imm2 | *'How about [A]?'*<br>*'How about color?'* |
| seek-attr-early | *'What is the [A] of that [Z] [C] [M] [S]?'*<br>*'What is the shape of that shiny thing?'* |
| seek-attr-sim-early | *'What about the earlier [Z] [C] [M] [S]?'*<br>*'What about the earlier box?'* |
| seek-attr-rel-imm | *'If there is a thing to its [R], what [A] is it?'*<br>*'If there is a thing to its right, what color is it?'* |
| seek-attr-rel-early | *'If there is a thing [R] that [Z] [C] [M] [S], what [A] is it made of?'*<br>*'If there is a thing in front of that shiny object, what material is it made of?'* |

Table 3: Example templates for all the caption and question types used to generate CLEVR-Dialog dataset. For each type, we show both: (a) a sample template with placeholders (Z=size, C=color, M=material, S=shape, A=attribute, X=count, R=relation), and (b) a realization with placeholders filled with random values.




(a) Distribution of questions according to the template labels.




(b) Distribution of answers.

Figure 8: Visualization of distributions for question types and answers in our CLEVR-Dialog dataset. See Sec. C for more details.

(b) expressive: complex templates can be broken down into these primitives, (c) computationally efficient: outputs can reused for templates sharing similar primitive structures (as seen in Fig. 7), thus allowing an easy extension to new primitives and templates. We believe that CLEVR-Dialog represents not just a static dataset but also a recipe for constructing increasingly challenging grounded dialog by expanding this grammar.

## C Additional Datasets Analysis

Fig. 8 visualizes the distribution of caption templates, question templates, answers, and the history dependency of questions in CLEVR-Dialog.

**Caption Categories.** As the dialog between Q-er and A-er is initiated by the caption, care must be taken to ensure it is *interesting enough* to spawn clarifying questions from the Q-er. To this end, we carefully design four different categories of caption templates (Fig. 3a): (a) `Obj-unique` mentions an object with unique set of attributes in the image, (b) `Obj-count` specifies the presence of a group of objects with common attributes, (c) `Obj-extreme` describes an object at one of the positional extremes of the image (right, left, fore, rear, center), (d) `Obj-relation` talks about the relationship between two objects along with their attributes in a way that allows them to be uniquely identified in the complete scene graph $S_a$. Example captions are given in Tab. 3. In contrast, MNIST Dialog does not have captions.

**Question Categories and Types.** CLEVR-Dialog contains three broad question categories—count, exist, and seek—with each further containing variants totaling up to 23 different types of questions. In comparison, MNIST-Dialog only has 5 types of questions and is less diverse. The distributions for the question categories and question types are shown in Fig. 3a and Fig. 8a, respectively. Our questions are 60% seek as they open up more interesting follow-up questions, 23% count, and 17% exist.

**History Dependency.** Recall that our motivation for CLEVR-Dialog to create a diagnostic dataset

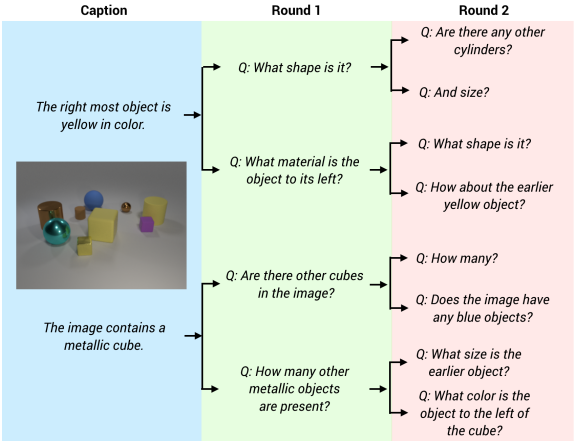



Figure 9: Dialog generation in CLEVR-Dialog. At each round, all valid question templates are used to generate candidates for the next question. However, only a few *interesting* candidates (beams) are retained for further generation, thus avoiding an exploding number of possibilities as rounds of dialog progress.

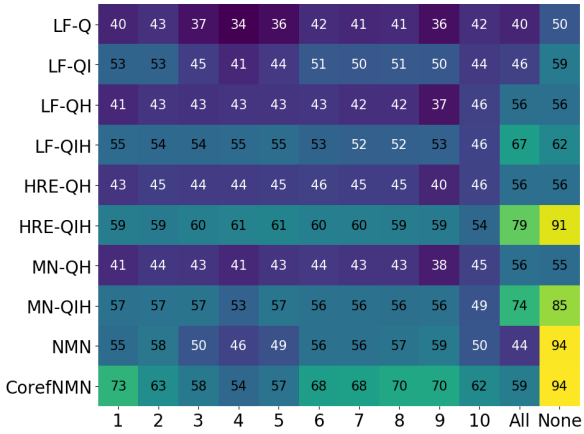



Figure 10: Accuracy breakdown of models according to the history dependency type. While CorefNMN outperforms all methods on questions (average) containing references ($1-10$), it performs poorly on questions that depend on the entire history ('All').

for multi-round reasoning in visual dialog. As a result, a majority of questions in our dataset depend on the dialog history. We identify three major kinds of history dependency for the questions: (a) **Coreference** occurs when a phrase within the current question refers to a earlier mentioned object (referent). We characterize coreferences by measuring the distance between the current and the earlier mention, in terms of dialog rounds. This can range from 1 (*e.g.*, *'What is its color?'*) to 10 (a question in round 10 referring to an entity in the caption). (b) **All**: When the question depends on the entire dialog history, *e.g.*, *'How many other objects are present in the image?'*, (c) **None**: When the question is stand-alone and does not depend on the history, *e.g.*, *'How many spheres does the scene have?'* The distribution of questions characterized according to the history dependency is shown in Fig. 3b. Unlike MNIST Dialog, CLEVR-Dialog contains a good distribution of reference distances beyond just 1, leading to a mean distance of 3.2. Thus, the models will need to reason through different rounds of dialog history in order to succeed.

# D Additional Model Analysis

In this section, we diagnose performance of all the models by breaking it down according to question type and history dependency. We then focus on the best performing model, CorefNMN (Kottur et al., 2018), which explicitly models coreferences in visual dialog by identifying the *reference* in the question (textual grounding) and then localizing the *referent* in the image (visual grounding). We study the behavior of CorefNMN on CLEVR-Dialog both qualitatively and quantitatively. Specifically, we visualize qualitative examples and develop metrics to quantitatively evaluate the textual and visual grounding. Note that such a diagnostic analysis is first of its kind which would not be possible without our CLEVR-Dialog.

## D.1 Accuracy vs History Dependency

The breakdown of model performances based on this history dependency is presented in Fig. 10. The following are the key observations:

- The best performing model, CorefNMN, has a superior performance (on an average) on all question with coreference ($1-10$) compared to all other models. As CorefNMN is designed specifically to handle coreferences in visual dialog, this is not surprising.
- Interestingly, the second best model HRE-QIH has the best accuracy on 'All' questions, even beating CorefNMN by a margin of 20%. In other words, HRE-QIH (and even MN-QIH) is able to answer 'All' questions significantly better than CorefNMN perhaps due to the ability of its dialog-level RNN to summarize information as the dialog progresses.
- Both NMN and CorefNMN perform similarly on the 'None' questions. This observation is intuitive as NMN is a history-agnostic version of CorefNMN by construction. However, the difference becomes evident as CorefNMN outperforms NMN by about 12% overall.

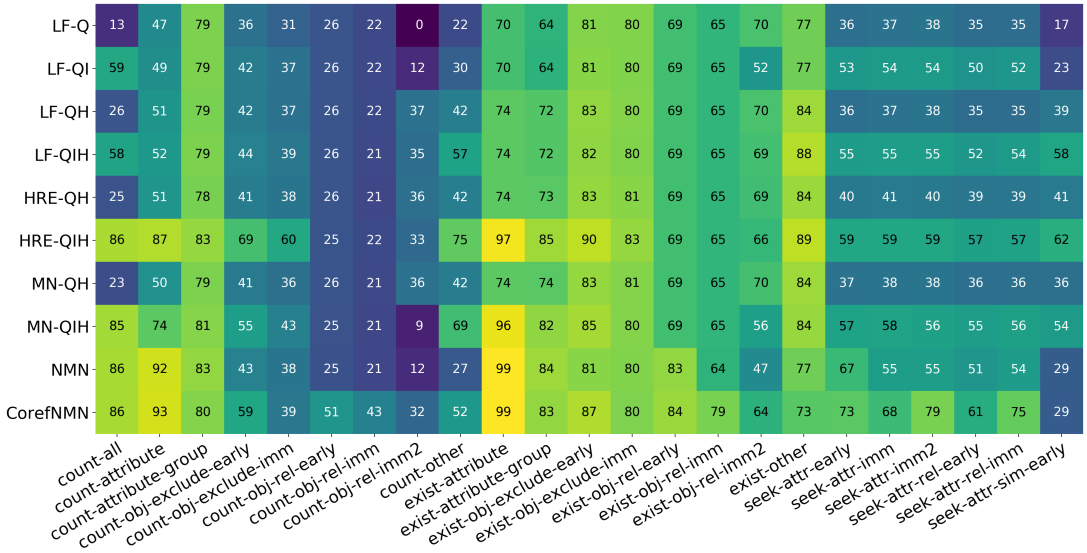



Figure 11: Accuracy breakdown of models according to the question type.

### D.2 Accuracy vs Question Type

Fig. 11 breaks down the performance of all the models according to the question types. An obvious observation is that performance on counting and seek questions is worse than that on exist questions. While this is in part because of the binary nature of exist questions, they are also easier to answer than counting or extracting attributes that need complicated visual understanding.

### D.3 Qualitative Anaylsis for CorefNMN

We now qualitatively visualize (Fig. 12) the best performing model, CorefNMN. In the example shown, CorefNMN first parses the caption *'There is a cyan metal object to the front of all the objects.'* and localizes the right cyan object. While answering Q-1, CorefNMN rightly instantiates the `Refer` module and applies the desired transformation (see module outputs on the right). For Q-2, it accurately identifies the object as the previous one, and extracts the attributes. Finally, the question *'What about that cyan object?'* cannot be answered in isolation as: (a) there are multiple cyan objects, (b) the meaning of the question is incomplete without Q-2. It is interesting to note that even though CorefNMN overcomes (a) by correctly resolving the reference *that cyan object* (in the image), it is unable to circumvent (b) due to its specialization in visual coreferences.

### D.4 Grounding Analysis for CorefNMN

As shown in Fig. 12, CorefNMN identifies a reference phrase in the current question and proceeds to

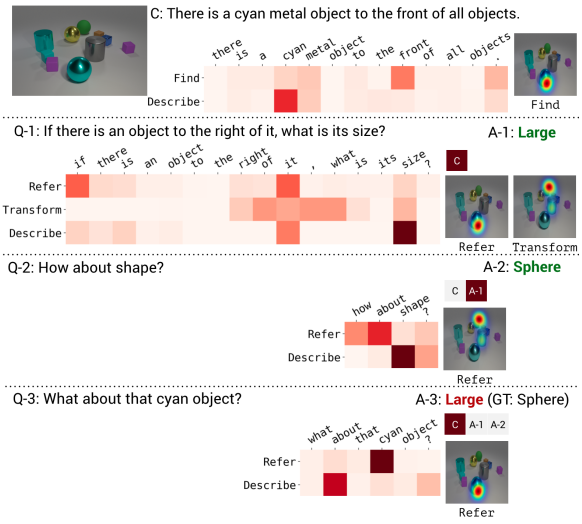
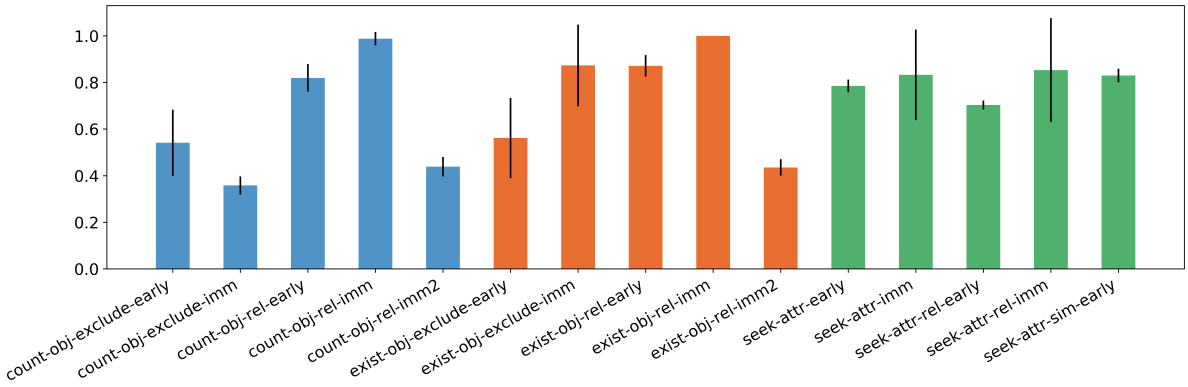



Figure 12: Qualitative visualization of CorefNMN on CLEVR-Dialog.

visually ground the corresponding referent in the image. Such explicit textual and visual grounding at each round allows for an interesting quantitative analysis for CorefNMN, with the help of annotations in our CLEVR-Dialog. To elaborate, CLEVR-Dialog provides coreference annotations for each question, if any, in the form of a reference phrase and its bounding box localization in the image. By comparing these grounding annotations with the output from the model, we can quantitatively assess grounding (both textual and visual) by CorefNMN. In what follows, we first describe the ground annotations, detail the evaluation procedure and then present our observations.

(a) NDCG value for text grounding for various question types.




(b) NDCG value for visual grounding for various question types.

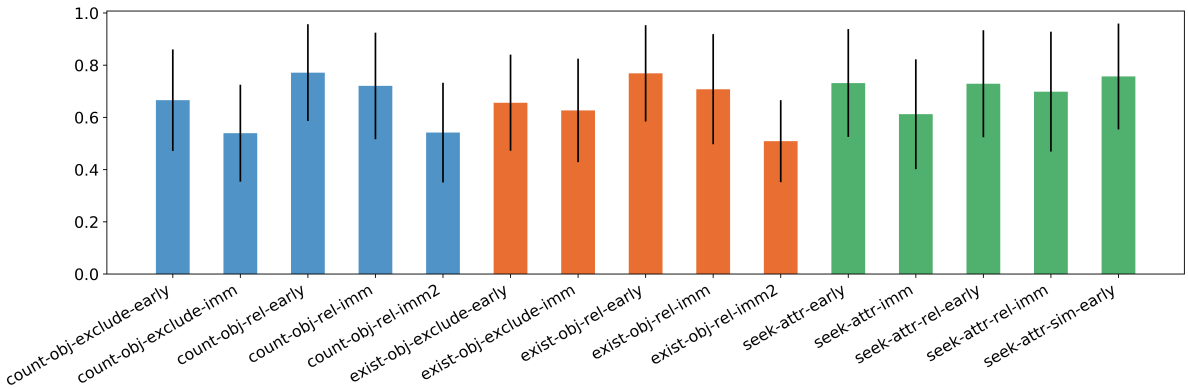



Figure 13: Evaluating the textual (above) and visual (below) grounding of CorefNMN on CLEVR-Dialog, using Normalized Discounted Cumulative Gain (NDCG) for various question types. Higher is better.

**Annotations.** While the original CLEVR dataset (Johnson et al., 2017) does not contain bounding box annotations for the objects in the scene, Krishna et al. (2018) later added these in their work on referring expressions. We leverage these annotations to obtain the ground truth visual groundings ($A_V$) for the referents in our questions. On the other hand, each of the caption and question templates has referring phrase annotations in them, thus giving the ground truth textual groundings ($A_T$). We use the above two groundings for evaluation.

**Evaluation.** For every coreference resolution, CorefNMN produces an visual attention map of size $14 \times 14$ ($\widehat{A}_V$) and a textual attention over the question words ($\widehat{A}_T$). We rank all the $14^2 = 196$ cells in $\widehat{A}_V$ according to their attention values. Next, we obtain the relevant cells among them from an appropriately scaled down ($14 \times 14$) version of $A_V$ and Next, we appropriately scaled down $A_V$ ($14 \times 14$) and consider the cells spanning the bounding box as relevant. To evaluate grounding, we measure the retrieval performance of the relevant cells in the sorted $\widehat{A}_V$ through the widely used Normalized Discounted Cumulative Gain (NDCG)[2]. It is a measure of how highly the relevant cells were ranked in the sorted $\widehat{A}_V$, with a logarithmic weighting scheme to higher ranks, thus higher is better. For the textual grounding, we perform a similar computation between $\widehat{A}_T$ and $A_T$ and report NDCG.

**Observations.** The NDCG values to evalute both textual and visual groundings for CorefNMN are shown in Fig. 13. An important takeaway being that the model is able to accurately ground the references in the question (Fig. 13a) consistently for several question types, as reflected in an higher average NDCG. On the other hand, the visual grounding in Fig. 13b is inferior compared to the ground truth annotations with a mean of around 0.3 and a high variance. This trend remains the same across all the question types. A possible hypothesis is that while the model is able to identify the references

[2] https://en.wikipedia.org/wiki/Discounted_cumulative_gain

in text, it is unable to resolve and ground the referent accurately in the image–an area of potential improvement.

## E Implementation details

Dataset generation was done entirely in Python, without any significant additional package dependencies. To evaluate the models from Das et al. (2017a), we use their open source implementation[3] based on Lua Torch[4]. For the neural module architectures (Hu et al., 2017; Kottur et al., 2018), we use the authors' Python-based, publicly available implementations—NMN[5] and CorefNMN[6]. Questions are encoded by first learning a 128-dimensional embedding for the words, which are then fed into a single layer LSTM of hidden size 128. We use a pretrained convolution neural network, ResNet-101 (He et al., 2016), to extract features for the images. Adam (Kingma and Ba, 2014) steps with a learning rate of 0.0001 are employed to maximize the loglikelihood of the ground truth answer, while training. A small portion (500 images) from the training set is set aside to pick the best performing model via early stopping. Our code and dataset are publicly available[7].

[3] https://github.com/batra-mlp-lab/visdial
[4] http://torch.ch/
[5] https://github.com/ronghanghu/n2nmn
[6] https://github.com/facebookresearch/corefnmn
[7] https://github.com/satwikkottur/clevr-dialog