Attribute Diversity Determines the Systematicity Gap in VQA

Ian Berlot-Attwell[*]
University of Toronto
Vector Institute
ianberlot@cs.toronto.edu

Kumar Krishna Agrawal
University of California, Berkeley
kagrawal@berkeley.edu

A. Michael Carrell
University of Cambridge
ac2411@cam.ac.uk

Yash Sharma[†]
University of Tübingen
yash.sharma@bethgelab.org

Naomi Saphra[†]
The Kempner Institute at Harvard University
nsaphra@fas.harvard.edu

Abstract

Although modern neural networks often generalize to new combinations of familiar concepts, the conditions that enable such compositionality have long been an open question. In this work, we study the systematicity gap in visual question answering: the performance difference between reasoning on previously seen and unseen combinations of object attributes. To test, we introduce a novel diagnostic dataset, CLEVR-HOPE. We find that the systematicity gap is not reduced by increasing the quantity of training data, but is reduced by increasing the diversity of training data. In particular, our experiments suggest that the more distinct attribute type combinations are seen during training, the more systematic we can expect the resulting model to be. We release our data and code at https://github.com/ikb-a/systematicity-gap-in-vqa.

1 Introduction

Systematicity, the ability to handle novel combinations of known concepts, is a type of compositional generalization (Hupkes et al., 2020). While systematicity is crucial to human intelligence (Fodor and Pylyshyn, 1988), conventionally trained neural networks often struggle to generalize systematically (Csordás et al., 2021b,a; Csordás et al., 2022).

Inspired by prior work investigating compositionality failures in language models (Press et al., 2023), we study the *systematicity gap* in visual question answering (VQA): the drop in model performance when reasoning about a combination of properties that was held out from both the text and vision modalities at train time. As an example, let us consider MATERIAL and SHAPE as two *attribute types*. If a model was trained without exposure to a particular combination of *attribute values*, e.g., `rubber sphere`, then we say the model composes systematically if it has high performance at test time on data that includes a `rubber sphere`.

Our work empirically demonstrates that systematicity emerges in a neural VQA model if the model is trained with diverse contexts for the attribute values in question (i.e., exposed to many MATERIAL-SHAPE combinations). The intuition for this hypothesis is simple: given many training examples of distinct combinations, the model learns how material and shape interact, and thus systematically generalizes to an unseen combination of MATERIAL and SHAPE. In contrast, a model trained on low-diversity data (i.e., only exposed to a few MATERIAL-SHAPE combinations) fails to learn rules to recombine them.

Using CLEVR-HOPE, a novel dataset for evaluating systematicity on a variety of held-out object attribute value pairs in a controlled setting, we measure the systematic compositionality of multimodal transformer and neurosymbolic models. We find that, while systematicity does not improve with more training data, it does improve with more *diverse* training data. Specifically, attribute types that include more diverse combinations during training can be composed systematically.

2 CLEVR-HOPE Diagnostic Dataset

Our dataset is based on CLEVR (Johnson et al., 2017a), a synthetic experimental setting for testing basic visual reasoning skills. CLEVR comprises English questions (such as “What is the color of the cube on the right side of the yellow sphere?") and corresponding 3D-rendered images of colored blocks. Each block has four attribute types (SIZE, COLOR, MATERIAL, and SHAPE). Our experiments rely on data splits that create distinct in-distribution (ID) and out-of-distribution (OOD) test sets.

We present the CLEVR Held-Out Pair Evaluation (CLEVR-HOPE) dataset for testing the systematicity of VQA models. CLEVR-HOPE is a

[*] Project lead

[†] Joint senior authors

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 9576–9611
November 12-16, 2024 ©2024 Association for Computational Linguistics

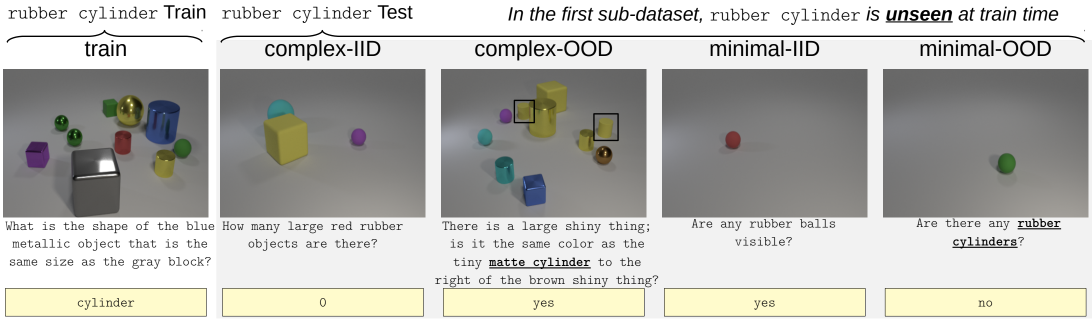
rubber cylinder Train
rubber cylinder Test
In the first sub-dataset, rubber cylinder is unseen at train time
train
complex-IID
complex-OOD
minimal-IID
minimal-OOD
What is the shape of the blue metallic object that is the same size as the gray block?
How many large red rubber objects are there?
There is a large shiny thing; is it the same color as the tiny matte cylinder to the right of the brown shiny thing?
Are any rubber balls visible?
Are there any rubber cylinders?
cylinder
0
yes
yes
no

Figure 1: Example image-question pairs for the sub-dataset of CLEVR-HOPE corresponding to `rubber cylinder`.The test sets are in gray; `rubber cylinder` is omitted visually *and* textually in the train split and the IID test splits; `rubber cylinder` only occurs in the OOD splits; occurrences are emphasized in this figure. The train and complex sets are of comparable visual and textual complexity to CLEVR. The minimal sets consist only of existence questions, checking whether a single object matches a given pair of attribute values.

controlled setting to test whether VQA models generalize to pairs of attribute values that were not seen during either training or fine-tuning. Within CLEVR-HOPE, we refer to an unseen pair of attribute values as a Held-Out Pair (HOP). The dataset is composed of 29 sub-datasets, each for a different HOP (see Tab. 1 for the list of HOPs) .

The 29 HOPs are selected such that there are 5 HOPs from each of the 6 possible pairs of attribute types, with the exception of SIZE+MATERIAL as CLEVR contains only 4 such combinations.

Each HOP has its own train set and 4 test sets. For `rubber cylinder`, in Fig. 1, these datasets are:

train: 560k image-question pairs in the training/finetuning set. The data distribution is similar to CLEVR, but any images or questions involving `rubber cylinder` have been removed.

complex-IID test: Test data sampled from the train distribution (i.e., `rubber cylinder` is filtered out). This is a standard IID test set; it's primarily used as a point of comparison for the complex-OOD test split.

complex-OOD test: Test data sampled from the CLEVR distribution filtered to always have (i) at least one object matching `rubber cylinder`, and (ii) `rubber cylinder` in the question. It is OOD as `rubber cylinder` is a combination never seen visually or textually in the train data. This split requires the model to behave systematically (i.e., generalize to `rubber cylinder`) while reasoning over scenes with several objects.

minimal-IID test: Minimal image-question pairs that check whether a model can recognize pairs of attribute values, corresponding to `rubber cylinder`'s attribute types, that were seen in the train set. The scene contains a single object, and the question only asks if there is an object matching a given pair of attributes. This split is called IID as it only contains pairs of attribute seen at train time. This split provides a point of comparison for the minimal-OOD test split.

minimal-OOD test: Minimal image-question pairs that check recognition of `rubber cylinder`. This split tests a model's systematicity independent of reasoning or visual clutter. If a model had some systematic behaviour which degraded when reasoning was required, then we'd expect to see a small performance gap between minimal-OOD test and minimal-IID test, and a large gap between complex-OOD test and complex-IID test. By construction, always returning false would yield 75% accuracy (See Appx. B.1 for details).

Summarizing the split naming convention: "complex" splits contain CLEVR-like visual and textual complexity. "minimal" splits contain single-object images, and only recognition questions (e.g., "Are any rubber cylinders visible?"). IID splits contain only the attribute pairs seen at train time. OOD splits contain the HOP in both the question, and in at least one object in the scene.

Appendix B includes dataset details. Note, CLEVR-HOPE omits validation sets to prevent tuning for specific task (Teney et al., 2020); instead, hyperparameters should be chosen using CLEVR.

3 Models & Training

Models: Our analysis focuses on LXMERT (Tan and Bansal, 2019), a multi-modal transformer-based (Vaswani et al., 2017) architecture.We also run experiments on a neurosymbolic model, TensorNMN (Johnson et al., 2017b), a neural module network (Andreas et al., 2016) that decomposes a task

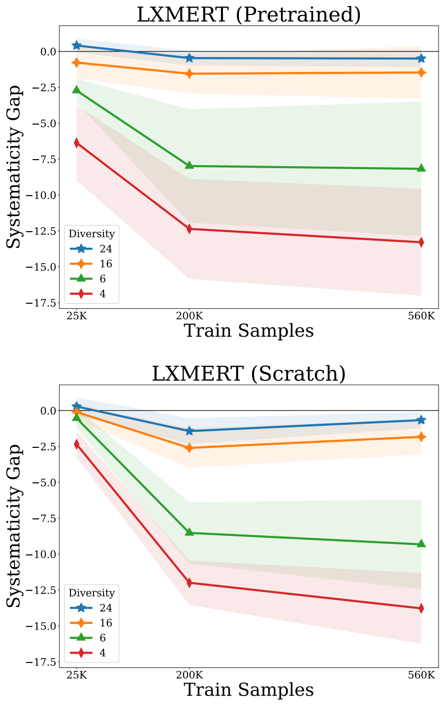

Figure 2: Systematicity gap (difference between OOD and IID accuracy) on the *complex* test split, averaged by (HOP) diversity for 29 HOPs, each with 3 runs.

into a composition of subtask-specific modules.

Training: For each HOP, we subsample the training set to test the impact the amount of training data has on performance. For 3 random seeds per HOP, we finetune pretrained LXMERT (LXMERT-p) and train LXMERT from scratch (LXMERT-s). We also train Tensor-NMN from scratch, again for three runs, though only for the first 6 HOPs, combinations of {`large`, `cyan`, `rubber`, `cylinder`}.

For hyperparameter selection, we perform a grid search on the original CLEVR dataset (Johnson et al., 2017a). For further details, see Appendix C.

4 Results

4.1 Evidence of Systematic Behaviour

With sufficient training data, over 93% of the tested model-HOP combinations exceed 75% accuracy on the minimal-OOD test set, with some reaching 100% (see Fig. 3). The VQA models have a wide range of accuracies generalizing to different held out pairs. On all models tested, this accuracy varies by around 25% across different HOPs.

Performance on the complex-OOD test set generally increases with the amount of training data; OOD accuracies across HOPs are similarly distributed (see Fig. 4). We conclude that the models consistently exhibit at least some degree of systematicity and we observe the same trend for Tensor-NMN (see Appx. Figs. 13 and 15).

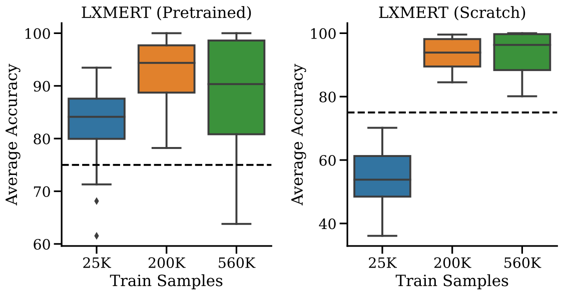

Figure 3: Box plots of **minimal-OOD test** set performance on all 29 HOPs. The average performance for each HOP is produced by averaging over 3 trials. The variation captured by this boxplot is from the difference in average performance between HOPs, rather than from the variation within the 3 trials.

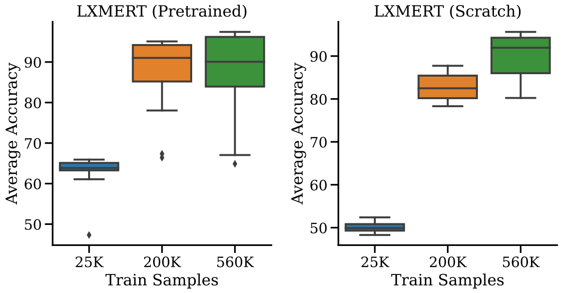

Figure 4: Box plots of **complex-OOD test** set performance on all 29 HOPs. As in Fig. 3, each HOP is individually averaged over 3 trials.

4.2 Systematicity Gap

Knowing that our models can behave systematically, we now ask whether there is any trend in the difference between in- and out-of-distribution performance: i.e., as the size of the training set increases (and thus the model's performance generally improves), does its performance on held-out combinations approach its performance on the combinations already seen at train time? We call this performance difference, between the OOD and IID combinations, the *systematicity gap*.

For example, if a model has an IID accuracy of 95%, but only 80% for data that requires the model to systematically compose `rubber` and `cylinder` into the held out pair `rubber cylinder`, then the *systematicity gap* is -15% (i.e., a 15% drop).

Given that the models are somewhat systematic, and that performance in general improves with more training data, one might expect that the systematicity gap would trend to zero. To the contrary, we find that, averaging over all HOPs, the

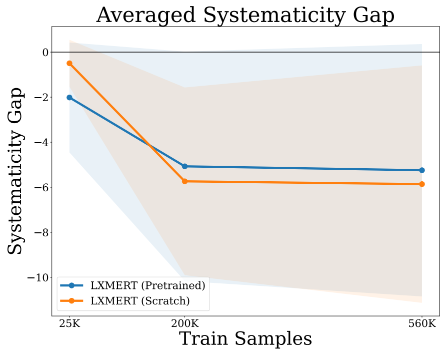

Figure 5: Average systematicity gap on **complex** examples (i.e., complex-OOD test accuracy minus complex-IID test accuracy) with 1 standard deviation; averaged over 3 runs on each of the 29 HOPs. The systematicity gap plateaus, suggesting that the performance drop when generalizing to unseen combinations does not improve with additional training data.

LXMERT systematicity gap plateaus to a drop of 5-6% (see Fig. 5). On the minimal test sets, the systematicity gap again plateaus, to a drop of 6-8% (see Appx. Fig. 19). The same trends are observed in Tensor-NMN (see Appx. Figs. 20 and 21), though the systematicity gap on minimal examples widens with additional training data.

With that said, the standard deviation of the observed systematicity gap is quite high – in the following section we make the case that the nature of the training data, specifically the attribute diversity seen at train time, is responsible.

4.3 Train-time conceptual diversity impacts systematicity

We define **attribute diversity** as the number of possible attribute values corresponding to the unseen combination's attribute types. For example, if the unseen combination is `rubber cylinders`, that corresponds to the material and shape attribute types. Given there are 2 possible materials and 3 possible shapes in the training set, there are $2 \times 3 = 6$ possible material-shape combinations; thus the attribute diversity is 6.

Tab. 1 lists the attribute diversity of the HOPs. Since the CLEVR training distribution is uniform across object attribute values, for a train set of fixed size, as attribute diversity increases, the number of examples per combination decreases.

Fig. 2 again illustrates the systematicity gap, but now only averages over HOPs of the same diversity (rather than over *all* HOPs as in Sec. 4.2). With this, we see that the systematicity gap is stratified by the diversity of the combinations seen at train time. Specifically, as the diversity of the training data increases, the systematicity gap narrows. In fact, the gap is typically near or within a standard deviation of zero for diversities of 16 or above. In comparison, diversities of 6 show a a plateauing systematicity gap stabilizing at 7-14%. We observe similar results with the systematicity gap of the minimal test sets (see Appx. Fig. 22).

HOP	Attribute Types	Diversity
`Large rubber`	Size + Material	4
`Small rubber`	Size + Material	4
`Large metal`	Size + Material	4
`Small metal`	Size + Material	4
`Rubber cylinder`	Material + Shape	6
`Metal cylinder`	Material + Shape	6
`Rubber cube`	Material + Shape	6
`Metal cube`	Material + Shape	6
`Rubber sphere`	Material + Shape	6
`Large cylinder`	Size + Shape	6
`Small cylinder`	Size + Shape	6
`Small cube`	Size + Shape	6
`Large cube`	Size + Shape	6
`Small sphere`	Size + Shape	6
`Rubber cyan`	Material + Color	16
`Rubber brown`	Material + Color	16
`Rubber purple`	Material + Color	16
`Metal red`	Material + Color	16
`Metal gray`	Material + Color	16
`Large cyan`	Size + Color	16
`Small brown`	Size + Color	16
`Small purple`	Size + Color	16
`Small red`	Size + Color	16
`Large gray`	Size + Color	16
`Cyan cylinder`	Color + Shape	24
`Brown sphere`	Color + Shape	24
`Red cylinder`	Color + Shape	24
`Gray cube`	Color + Shape	24
`Purple sphere`	Color + Shape	24

Table 1: HOP Diversity; i.e., number of attribute values corresponding to the HOP's attribute types.

For Tensor-NMN, we also find stratification by diversity for complex examples (see Appx. Fig. 26). The trend on minimal examples is noisier, but converges to the expected ordering (see Appx. Fig. 27).

4.4 Controlling for attribute category

A particular attribute can introduce confounders, such as the overall difficulty of learning its category. We ran additional experiments explicitly controlling for the attribute category to verify diversity's impact on the systematicity gap. In our prior experiments, attribute diversity is intrinsically tied to attribute type. As seen in Tab. 1, the most diverse pairs are always color-shape combinations, and the least diverse pairs are always size-material combinations. Thus, it is possible that we are actually measuring the effects of attribute type on generalization, rather than diversity. To address this, here we vary the attribute diversity while keeping the attribute type combination fixed.

We focused on SHAPE-COLOR combinations and generated multiple datasets with varying levels of diversity [4, 8, 16, 24] by varying the unique color-shape combinations present during training. We trained separate instances of LXMERT-s on these datasets and evaluated performance on corresponding HOPs (averaged across 3 random seeds). In Fig. 6, we see that lower attribute diversity led to worse systematicity gap.

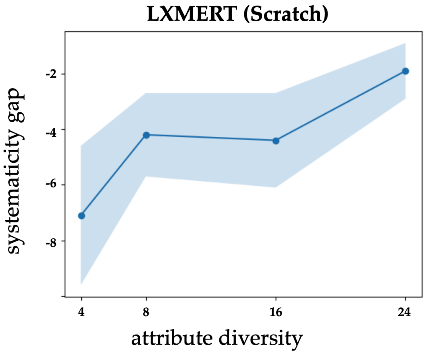

Figure 6: For attribute pair COLOR + SHAPE, we control the diversity by subsampling fixed number of combinations (one of [4, 8, 16, 24]), and finetuning the model accordingly. On the complex test sets, we observe that increasing attribute diversity reduces systematicity gap.

5 Related work

While compositionality in VQA has been studied, prior work has focused on generalization to new question structures (de Vries et al., 2019; Vani et al., 2021; Bogin et al., 2021), task-object combinations (Whitehead et al., 2021), or question-answer combinations (Agrawal et al., 2017), rather than new attribute combinations. Systematicity has often been investigated through synthetic datasets to control for the model's exposure to particular attribute combinations. Lake and Baroni (2018) introduced the SCAN benchmark to evaluate compositionality in sequence-to-sequence models, revealing a lack of systematicity. Followup (Patel et al., 2022; Jiang et al., 2022) and concurrent (Zhou et al., 2023) seq2seq works have shown that the conceptual diversity of the training set significantly affects systematicity—our work extends these findings to the multi-modal domain of VQA.

The closest prior work is the CLEVR-CoGenT dataset: Johnson et al. (2017a) created a train-test CLEVR split where at train time cubes and cylinders are restricted to limited color palettes, that are reversed at test time. They observed that model performance declined on held-out attribute combinations. But, unlike CLEVR-HOPE, CLEVR-CoGenT does not change the question distribution at train time—held-out combinations can leak by appearing in text at train time. Furthermore, CLEVR-CoGenT has only a single train set with held-out COLOR-SHAPE combinations—whereas CLEVR-HOPE expands the set of held-out combinations to 29 train sets, covering all possible pairs of attribute types. CLEVR-HOPE also independently assesses each HOP, including in a minimal setting. In combination, these improvements allow us to study the impact of train-time diversity.

Our results align with concurrent work on the effects of training diversity in VQA: Rahimi et al. (2023) modify CLEVR to study the related question of productivity, concluding that increasing the diversity of question combinations increases productivity. Unlike our work, they do not use a transformer architecture, instead studying MAC (Hudson and Manning, 2018), FiLM (Perez et al., 2018), and Vector-NMN (de Vries et al., 2019). Additionally, as they study a fundamentally different question, their dataset only alters the question distribution—their image distribution is unchanged between train and test time. Given that systematicity and productivity are both aspects of compositional generalization (Hupkes et al., 2020), the growing evidence across task settings and facets of compositionality (Oren et al., 2021; Levy et al., 2023) suggests a close relationship between train-time diversity and compositional generalization as a broad phenomenon.

We encourage further study, and release our code and data to facilitate these efforts. Confirming our findings in newer vision-language models, quantifying them with scaling laws, exploitation in data curation pipelines, and using these effects to predict model reliability, are all important future work.

6 Conclusions

Using CLEVR-HOPE, we demonstrate that several models exhibit a degree of systematic generalization to held-out object attribute pairs. Furthermore, we illustrate that the systematicity gap (the difference between in- and out-of-distribution performance) does not improve with more data, but does with more attribute diverse data—i.e., the number of attribute pairs of the same type seen at train time.

Limitations

First and foremost, while the synthetic nature of CLEVR-HOPE allows for a more controlled study of models, it raises the question whether the observed results will hold in other vision-language tasks, or in more complex and diverse real-world settings. Based on similar findings in text domains (Patel et al., 2022; Jiang et al., 2022; Zhou et al., 2023; Oren et al., 2021; Levy et al., 2023), and evidence of a link between productivity and diversity in VQA (Rahimi et al., 2023), we anticipate that our findings would apply in other multimodal settings like image or caption generation – although we leave that to future work. The investigation of scaling laws in which these diversity requirements breakdown is an important direction for future work.

The second major limitation arises from the choice of models. LXMERT uses a pretrained F-RCNN (Ren et al., 2015) for object detection, which we do not alter. As the F-RCNN is pretrained, it may already possess implicit knowledge of the attributes (e.g., shape), and may contribute systematic structure to LXMERT. Any such visual knowledge or biases are therefore given to both LXMERT-p and LXMERT-s. In contrast, note that the language component of LXMERT-s is randomly initialized—whereas (Tan and Bansal, 2019) initialized their language transformer with BERT (Devlin et al., 2019) when pretraining from scratch. Similarly, Tensor-NMN uses a frozen pretrained ResNet (He et al., 2016) as its vision backbone, and its language components and modules are initialized from scratch. A related limitation is that LXMERT-p may have been exposed to the held-out attribute during its pretraining phase; we control for this via the LXMERT-s experiments where no vision-language pretraining is performed.

More generally, the models we studied are not the strongest. However, despite this we find that they yield high accuracy across the board (see Appx. Fig. 17), which indicates to us that analyzing the systematicity gap for these models still provides meaningful insight. While we do not know for certain whether our results will extrapolate to stronger models, we recommend analysis for stronger models as an important direction for future work, and enable such by releasing our data and code. The literature studying the impact of data distribution on learning often proposes and demonstrates an effect in toy settings initially, and it is this area that our paper falls within.

Beyond stronger models, it may be interesting to confirm our findings on other NMN variants such as N2NMN (Hu et al., 2017), or on scene-graph based architectures such as XNM (Shi et al., 2019) or the Neural State Machine (Hudson and Manning, 2019b) which were also designed to encourage compositionality.

Finally, due to resource limitations, we only evaluate Tensor-NMN on 6 of the 29 total HOPs, one for each attribute type combination.

Ethics Statement

We judge that our work has very low risk. The primary risk is of using our dataset to measure model systematicity in models that are not trained on our train/test split. We have provided a highly specific diagnostic dataset that is designed to provide a data split for testing generalization claims, and our OOD set is not useful to measure generalization in arbitrary VQA models. This concern is documented in the dataset datasheet in Section H of the Appendix.

Our dataset has the disclosed bias that it only contains English questions, however this is reasonably overshadowed by the synthetic nature of the text (see the Limitations section). We also believe that given the functional forms of the questions, it should be fairly straightforward to convert the functional forms into synthetic natural language equivalents for a given target language. As to the images, any cultural biases in the specific choices of shape, colour, size, materials, position, and count of the objects are inherited from the original CLEVR (Johnson et al., 2017a).

Apart from the dataset, our contributions are findings on the impact of train-time diversity on model systematicity. To the best of our understanding, this finding does not introduce any new capabilities to the models in question, nor does it affect the accessibility or impact of these models.

Acknowledgements

Resources used in preparing this research were provided, in part, by the Department of Computer Science at the University of Toronto, the Province of Ontario, the Government of Canada through CIFAR, companies sponsoring the Vector Institute (www.vectorinstitute.ai/partnerships/current-partners/), the Hyundai Motor Company (under the project Uncertainty in Neural Sequence Modeling), the Samsung Advanced Institute of Technology (under the project Next Generation Deep Learning: From Pattern Recognition to AI), and by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

Ian Berlot-Attwell is funded by a Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship-Doctoral, and a Vector Institute Research Grant. A. Michael Carrell is funded in part by a Microsoft Research scholarship. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Yash Sharma.

We appreciate the invaluable ideas and discussion from Preetum Nakkiran, Spencer Frei, Nicholas Schiefer, and Ferenc Huszár who helped shape the early stages of this work, and Preetum in particular for enabling this collaboration. We would also like to thank Jonathan Shi for his early contributions to our code base, including help with, and modification to, the CLEVR dataset generation code. We would also like to thank Frank Rudzicz for being a wonderful PhD supervisor, and for his suggestions and advice throughout the work.

References

Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. 2017. C-VQA: A compositional split of the visual question answering (VQA) v1.0 dataset. *CoRR*, abs/1704.08243.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

Ben Bogin, Shivanshu Gupta, Matt Gardner, and Jonathan Berant. 2021. COVR: A test-bed for visually grounded compositional generalization with real images. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9824–9846, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021a. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2021b. Learning adaptive control flow in transformers for improved systematic generalization. In *Advances in Programming Languages and Neurosymbolic Systems Workshop*.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. 2022. The neural data router: Adaptive control flow in transformers improves systematic generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Harm de Vries, Dzmitry Bahdanau, Shikhar Murty, Aaron C. Courville, and Philippe Beaudoin. 2019. CLOSURE: Assessing systematic generalization of CLEVR models. In *Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019*.

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

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Christian Garbin. 2021. Datasheet for dataset template. `https://www.overleaf.com/latex/templates/datasheet-for-dataset-template/jgqyyzyprxth`. Accessed: 2023-12-10.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

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)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

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 *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 804–813. IEEE Computer Society.

Drew A. Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Drew A Hudson and Christopher D Manning. 2019a. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Drew A. Hudson and Christopher D. Manning. 2019b. Learning by abstraction: The neural state machine. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5901–5914.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *J. Artif. Intell. Res.*, 67:757–795.

Yichen Jiang, Xiang Zhou, and Mohit Bansal. 2022. Mutual exclusivity training and primitive augmentation to induce compositionality. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11778–11793, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017a. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017b. Inferring and executing programs for visual reasoning. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3008–3017. IEEE Computer Society.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. Association for Computational Linguistics.

Inbar Oren, Jonathan Herzig, and Jonathan Berant. 2021. Finding needles in a haystack: Sampling structurally-diverse training sets from synthetic data for compositional generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10793–10809, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, Phil Blunsom, and Navin Goyal. 2022. Revisiting the compositional generalization abilities of neural sequence models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 424–434, Dublin, Ireland. Association for Computational Linguistics.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3942–3951. AAAI Press.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Amir Rahimi, Vanessa D'Amario, Moyuru Yamada, Kentaro Takemoto, Tomotake Sasaki, and Xavier Boix. 2023. D3: Data diversity design for systematic generalization in visual question answering. *arXiv preprint arXiv:2309.08798*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Jiaxin Shi, Hanwang Zhang, and Juanzi Li. 2019. Explainable and explicit visual reasoning over scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8376–8384. Computer Vision Foundation / IEEE.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. 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 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Damien Teney, Ehsan Abbasnejad, Kushal Kafle, Robik Shrestha, Christopher Kanan, and Anton Van Den Hengel. 2020. On the value of out-of-distribution testing: An example of goodhart's law. *Advances in neural information processing systems*, 33:407–417.

Ankit Vani, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville. 2021. Iterated learning for emergent systematicity in VQA. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Spencer Whitehead, Hui Wu, Heng Ji, Rogério Feris, and Kate Saenko. 2021. Separating skills and concepts for novel visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5632–5641. Computer Vision Foundation / IEEE.

Xiang Zhou, Yichen Jiang, and Mohit Bansal. 2023. Data factors for better compositional generalization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14549–14566, Singapore. Association for Computational Linguistics.

A Extended Related Work

While compositionality in VQA has been studied, prior work has focused on generalization to new question structures (de Vries et al., 2019; Vani et al., 2021; Bogin et al., 2021), task-object combinations (Whitehead et al., 2021), or question-answer combinations (Agrawal et al., 2017), rather than new attribute combinations. One reason for this gap is that, with natural data, it is hard to control for the model's exposure to particular attribute combinations. By using a controlled synthetic setting, we can guarantee that generalization behavior is systematic based on the data split.

Systematicity has often been investigated through synthetic datasets. Lake and Baroni (2018) introduced the SCAN benchmark to evaluate compositionality in sequence-to-sequence models, revealing a lack of systematicity. Followup (Patel et al., 2022; Jiang et al., 2022) and concurrent (Zhou et al., 2023) seq2seq works have shown that the conceptual diversity of the training set significantly affects systematicity—our work extends these findings to the multi-modal domain of VQA.

The closest prior work is the CLEVR-CoGenT dataset: Johnson et al. (2017a) created a train-test CLEVR split where at train time cubes and cylinders are restricted to limited color palettes, that are reversed at test time. They observed that model performance declined on held-out attribute combinations. But, unlike CLEVR-HOPE, CLEVR-CoGenT does not change the question distribution at train time—held-out combinations can leak by appearing in text at train time. Furthermore, CLEVR-CoGenT has only a single train set with held-out COLOR-SHAPE combinations—whereas CLEVR-HOPE expands the set of held-out combinations to 29 train sets, covering all possible pairs of attribute types. CLEVR-HOPE also independently assesses each HOP, including in a minimal setting. In combination, these improvements allow us to study the impact of train-time diversity.

Beyond CLEVR-CoGenT, our results align with concurrent work on the effects of training diversity in VQA: Rahimi et al. (2023) modify CLEVR to study the related question of productivity. Specifically, generalization to questions with more reasoning steps, and generalization to new question combinations (e.g., answering counting questions about `shape`, when all train-time counting questions are about `color` or `size`). They conclude that increasing the diversity of question combinations increases productivity. Unlike our work, they do not use a transformer architecture, instead studying MAC (Hudson and Manning, 2018), FiLM (Perez et al., 2018), and Vector-NMN (de Vries et al., 2019). Additionally, as they study a fundamentally different question, their dataset only alters the question distribution—their image distribution is unchanged between train and test time.

Given that systematicity and productivity are both aspects of compositional generalization (Hupkes et al., 2020), the growing evidence across task settings and facets of compositionality (Oren et al., 2021; Levy et al., 2023) suggests a close relationship between train-time diversity and compositional generalization as a broad phenomenon.

B CLEVR-HOPE: Additional details

The full list of held-out pairs (HOPs) can be found in Table 1. The HOPs were selected by choosing two attribute values from each of `large cyan rubber cylinder`, `small brown rubber sphere`, `small red metal cylinder`, `large gray metal cube`, and `small purple rubber sphere`.

Note that there are only 4 possible MATERIAL-SIZE combinations, as there are only 2 SIZES and 2 MATERIALS. We include all 4 of these, as well as 5 HOPs for every other pair of attribute types.

Before selecting the 5 4-tuples from which we created the HOPs in CLEVR-HOPE, we first created a small set of minimal test questions for testing how well a given model comprehends a given attribute in isolation—CLEVR-PRELIM. For example, for the color cyan we had two types of tests. First, tests similar to the minimal-OOD test tests (i.e., a single object and rephrasings of “Are any cyan objects visible?”). Second, counting tests—all questions were rephrases of “What number of cyan objects are there?”, and images had varying numbers of cyan objects. Specifically, we fixed the position of 5 objects, and created 6 images, each with a different number of objects matching the attribute—i.e., 0, 1, 2, 3, 4, or 5 cyan objects.

Note that, unlike CLEVR-HOPE which studies pairs of attributes values, CLEVR-PRELIM evaluates *only* attribute values in isolation.

Using CLEVR-PRELIM, we performed a zero-shot evaluation of Tan and Bansal (2019)’s VQA2.0 (Goyal et al., 2017) fine-tuned LXMERT checkpoint. From this preliminary study we found that zero-shot model performance was generally poor (e.g., over all attribute values of all types, the highest count performance was 49.1%). Given our interest in studying the impact of the amount of training data, we created our first 4-tuple by individually selecting each attribute value; specifically choosing the attribute value that zero-shot LXMERT had the lowest performance on—this created the 4-tuple `Large cyan rubber cylinder`. The remaining four tuples were selected uniformly at random. Ultimately, as we did not see any significant difference between a small sample of 6 HOPs (those created from attribute pairs in `large cyan rubber cylinder`) and a larger sample of 23 HOPs (those created from random 4-tuples), we present results aggregated over all 29 HOPs.

Note that as two 4-tuples were `rubber spheres` and `small spheres`, we added the HOPs `rubber cube` and `small cube` so that we would maintain five MATERIAL-SHAPE and five SIZE-SHAPE pairs.

For each HOP in CLEVR-HOPE, the approximate size of the corresponding splits is outlined below:

- train set: 62k images, and 560k image-question pairs
- complex-IID test set: 13k images, 120k image-question pairs
- complex-OOD test set: 15k images, 15k image-question pairs
- minimal-IID test set: 2576-3200 images, 8640-11970 image-question pairs (depending on HOP)
- minimal-OOD test set: 448-3840 images, 448-3840 image-question pairs (depending on HOP)

To reduce the resources required to generate the dataset, images are reused throughout the dataset. Specifically, the images are reused across the train sets for the HOPs, and reused from the original CLEVR (Johnson et al., 2017a) training set.

Similarly, each of the test sets reuse images across HOPs. Note that while the complex-IID test and complex-OOD test sets do not reuse eachother’s images, the minimal-IID test and minimal-OOD test sets do for images that do not involve the HOP under consideration.

To ensure that CLEVR can be fairly used for hyperparameter tuning, and to prevent any data leakage, *no* CLEVR validation or test images are reused in CLEVR-HOPE.

For further information, including distribution and maintenance, see the CLEVR-HOPE Datasheet in Section H. The datasheet follows the format outlined by Gebru et al. (2021), and is modified from the template by Garbin (2021).

B.1 CLEVR-HOPE: minimal-OOD test set and minimal-IID test set

All images in the minimal-OOD test and minimal-IID test sets contain only a single object. All questions ask whether there are any objects matching the attribute value pair. E.g., for the HOP `rubber cyan`, some question variants include “Are there any cyan matte things?” and “Are any cyan matte things visible?”.

These splits are designed to test the model in a systematic manner: each image matching the HOP has 3 corresponding images that do not match the HOP. These 4 images share identical question phrasing. The non-matching images maintain the object position, lighting, and the attribute values that are irrelevant to the HOP, but change the first attribute value in the HOP, the second attribute value in the HOP, or both attribute values in the HOP, respectively. See Fig. 7 for an example.

Note that the question template is taken directly from the original CLEVR dataset generation code. The main change is the aforementioned systematic design, and that the images used contain only a single object, whereas the original CLEVR requires at least 3 objects in any scene.

The minimal-IID test split is created in the same way, but testing all other attribute-value pairs of

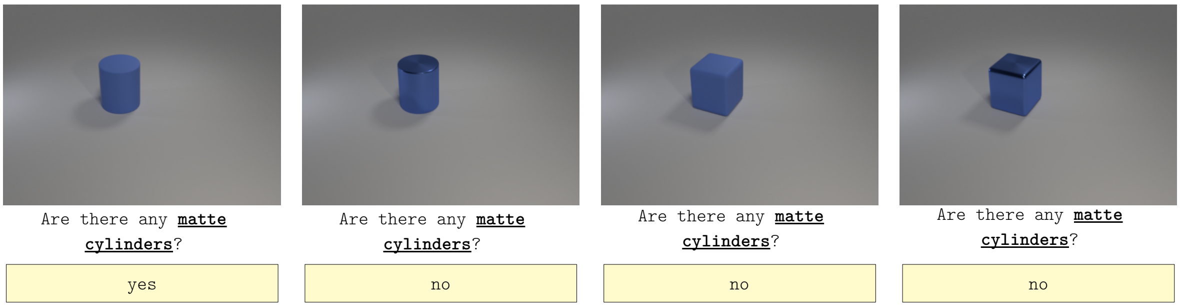
Are there any matte cylinders?
yes
Are there any matte cylinders?
no
Are there any matte cylinders?
no
Are there any matte cylinders?
no

Figure 7: Four example image-question pairs for the minimal-OOD test split of the sub-dataset of CLEVR-HOPE corresponding to the first held-out attribute pair—i.e., rubber cylinder. Note how the first image matches rubber cylinder (MATERIAL=rubber, and SHAPE=cylinder), and the next three image have one attribute value (MATERIAL=metal), the other attribute value (SHAPE=cube), or both (MATERIAL=metal, and SHAPE=cube) attribute values not matching rubber cylinder. This pattern repeats throughout the dataset, with the choice of distractor values, object position, lightning, question-phrasing and the value of the attribute-types not in HOP, all chosen randomly, but fixed within each set of 4 images.

Hyperparameter	LXMERT-p	LXMERT-s
Learning Rate	5e-5	1e-5
Gradient Updates	218,750	481,000
Batch size	32	32

Table 2: Key hyperparameter values used for LXMERT

the same type as the HOP. Note that the distractor attribute values in the negative examples were selected uniformly at random. Since this may create the held-out pair (and indeed, *must* do so for one of the four size-material images), after the initial creation of the minimal-IID test split, we filter it to remove any image-question pairs where the object in the image matches the HOP.

C Training details

All subsets of the train sets (i.e., of size 25k, 200k, and 560k) are created by taking the first however many indices. This corresponds to a random subset of images for 25k, which is consecutively randomly expanded. This is so because the image-question pairs are unsorted, apart from all questions for any given image having contiguous indices. Note that we fix the number of gradient updates across subset sizes, i.e., smaller subsets are trained for more epochs so that the total number of gradient updates is the same.

For LXMERT, the maximum sequence length is increased to 49 so that CLEVR-HOPE questions are not truncated.

For LXMERT-p, we follow Tan and Bansal (2019)'s procedure for finetuning their pretrained LXMERT checkpoint on a VQA dataset. As part of their procedure, the pretrained F-RCNN (Ren et al., 2015) object detector is *not* altered in any way.

LXMERT-p hyperparameters were modified from the hyperparameters used by Tan and Bansal (2019) for finetuning LXMERT for VQA. Specifically, Tan and Bansal (2019) finetuned LXMERT for the VQA tasks of VQAv2 (Goyal et al., 2017), NLVR2 (Suhr et al., 2019), and GQA (Hudson and Manning, 2019a) with a batch size of 32, 4 epochs, and a learning rate of either 1e-5 or 5e-5. We ultimately used a learning rate of 5e-5, and increased the epochs to 10 as we found it yielded better performance.

For LXMERT-s we randomly initialize all LXMERT weights (this *excludes* the pretrained F-RCNN object detector), and apply the LXMERT finetuning procedure (albeit with different hyperparamters) to train this randomly initialized model.

Both LXMERT models contain 209 million trainable parameters, in addition to the frozen F-RCNN object detector (65 million frozen parameters).

LXMERT-s hyperparameter tuning was performed via grid search over learning rate (1e-4, 5e-5, 1e-5) and training steps (218750, 481000, 700000). Note that we ultimately used 481k gradient update steps, as its validation accuracy (95.47%) was extremely close to 700k (96.99%), with nearly half the training time.

The LXMERT hyperparameters used are summarized in Tab. 2.

Tensor-NMN is trained from scratch following the process used by de Vries et al. (2019).Following their work, image features are extracted from the conv4 layer of a frozen ResNet101 (He et al., 2016). Tensor-NMN is trained in a 3 stage process—initially the program generator and execution engine are trained in a supervised manner, following which they are trained together using REINFORCE. The default hyperparameters for CLEVR from de Vries et al. (2019) are used.

The Tensor-NMN model contains 42 million trainable parameters, in addition to the frozen ResNet101 image feature extractor (27 million frozen parameters – less than the full ResNet101 as only the conv4 features are used).

Models were trained on a mixture of 16GB Nvidia Tesla T4 GPUs, and 8GB Nvidia GeForce RTX 2070 GPUs. Each run was trained on a single GPU, with the experiments spread over approximately 44 GPUs. We upper bound the number of GPU hours of compute used at approximately 24k, 32k, and 66k for the LXMERT-p, LXMERT-s and Tensor-NMN experiments respectively.

D LXMERT Detailed Results

LXMERT performance on minimal-OOD test can be found in Fig. 8. Performance on minimal-IID test can be found in Fig. 9. All plots mark 75%—this baseline performance is achieved on the minimal-OOD test split by always predicting false (i.e., the most common class). Always predicting false on minimal-IID test yield a baseline performance between 66% and 75%, depending on the HOP.

LXMERT performance on complex-OOD test can be found in Fig. 10. Performance on complex-IID test can be found in Fig. 11.

For LXMERT trained on the largest train sets (560k), we plot the complex and minimal model accuracies, averaged by the attribute types of the HOPs, in Fig. 12.

The exact average accuracies and standard deviations over 3 runs are in Tables 3 through 10.

E Tensor-NMN Detailed Results

As Tensor-NMN was only evaluated on the first 6 HOPs, we include the subset of LXMERT models trained on the same HOPs for comparison.

Model performance on minimal-OOD test can be found in Fig. 13. Performance on minimal-IID test can be found in Fig. 14. All plots mark 75%—this baseline performance is achieved on the minimal-OOD test split by always predicting false (i.e., the most common class). Always predicting false on minimal-IID test yield a baseline performance between 66% and 75%, depending on the HOP.

Model performance on complex-OOD test can be found in Fig. 15. Performance on complex-IID test can be found in Fig. 16.

For Tensor-NMN trained on the largest train sets (560k), we plot the complex and minimal model accuracies, averaged by the attribute types of the HOPs. The results are visualized in Fig. 17. Again, we include the corresponding subset of LXMERT models for comparison.

The exact average accuracies and standard deviations over 3 runs are in Tables 11 through 14.

F Systematicity Gap

As outlined in Section 4.2, we find that, on all models, averaged over HOPs, the gap between performance on complex questions involving IID vs. OOD attribute combinations does not trend to zero. Instead, it plateaus (see Figures 18 and 20). In comparison, the performance gap on minimal questions plateaus or decreases gently (see Figures 19 and 21).

In Fig. 23 we visualize the systematicity gap by attribute-types in the pair on both LXMERT and Tensor-NMN. It can be seen that the systematicity gaps are still sorted by the diversity of the attribute pairs (i.e., we see lighter colours in the top left, and darker colours in the bottom right).

The exact average systematicity gaps and standard deviations over 3 runs are in Tables 15 through 20.

The systematicity gaps for each individual HOP can be found in Figures 24 and 25 for the complex and minimal splits respectively.

F.1 Detailed Tensor-NMN Systematicity Gap

Averaging the systematicity gap in Tensor-NMN by diversity, we again find stratification by diversity for complex examples (see Fig. 26). The trend on minimal examples is noisier, but ultimately converges to the expected ordering (see Fig. 27). Note that, as is to be expected, when limited to the first six HOPs the LXMERT trend is also noisier. It is therefore reasonable to expect the Tensor-NMN trend would be cleaner with additional HOPs.

G Summary Statistics

The exact LXMERT-p and LXMERT-s average accuracies and standard deviations (averaged over 3 runs) are in Tables 3 through 10.

The exact Tensor-NMN average accuracies and standard deviations (averaged over 3 runs) are in Tables 11 through 14.

The exact average systematicity gaps and standard deviations (averaged over all runs for HOPs with the diversity in question) are in Tables 15 through 20.

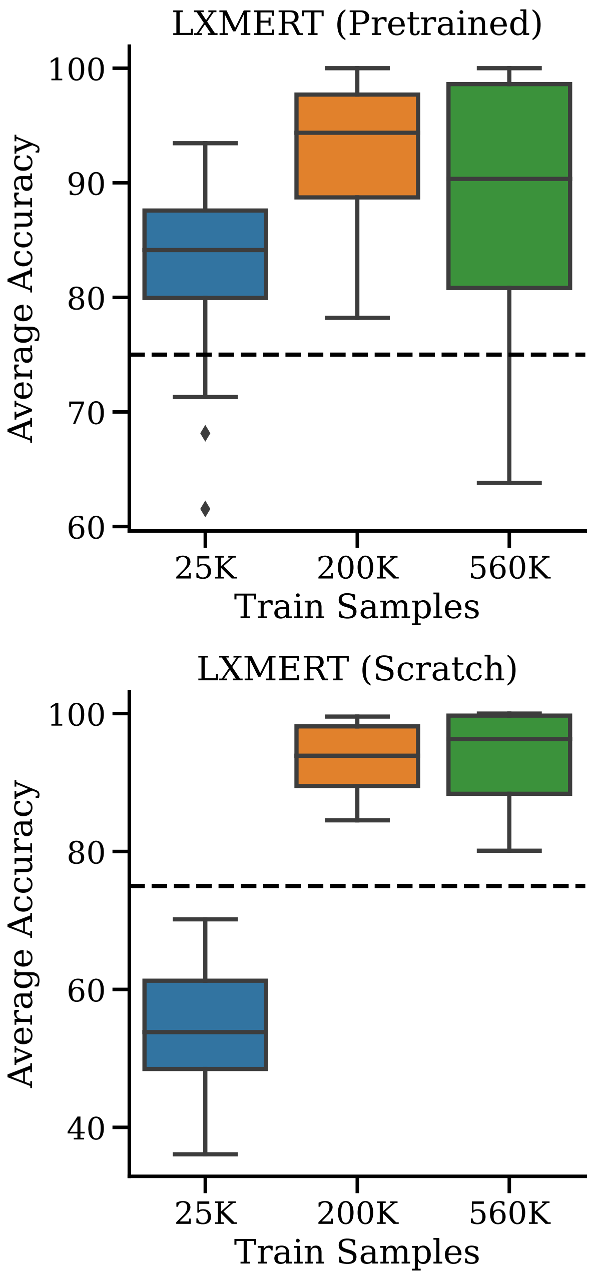

Figure 8: Box plot of **minimal-OOD test** set performance on all 29 HOPs. The average performance for each HOP is produced by averaging over 3 trials. The variation captured by this boxplot is from the difference in average performance between HOPs, rather than from the variation within the 3 trials.

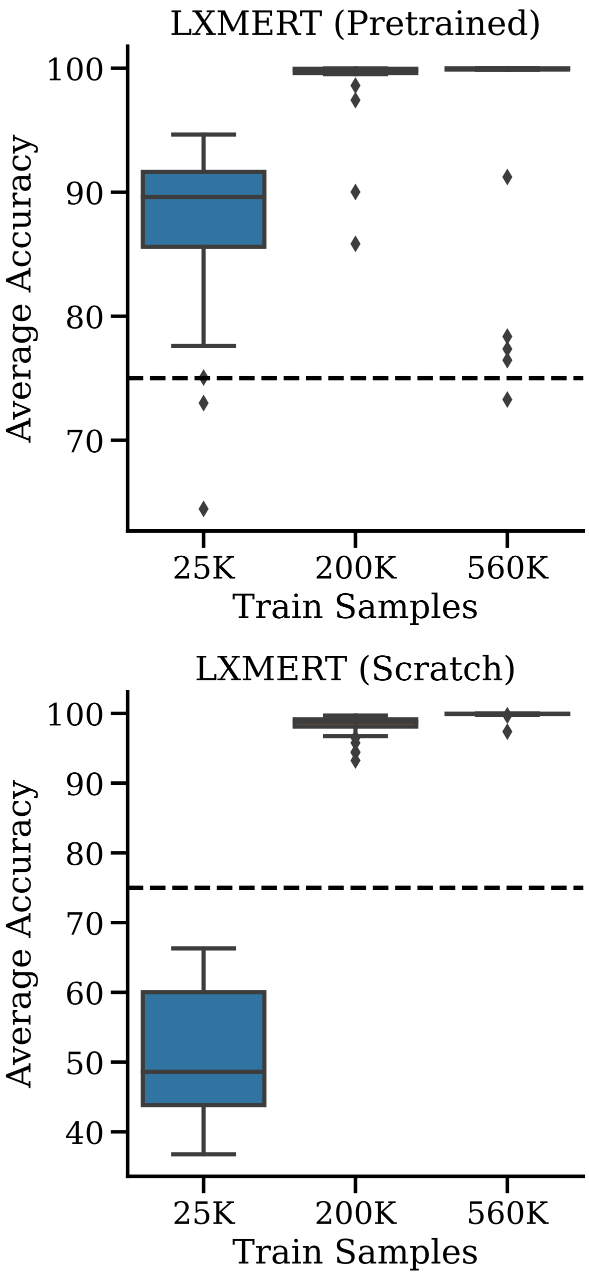

Figure 9: Box plot of **minimal-IID test** set performance on all 29 HOPs. The average performance for each HOP is produced by averaging over 3 trials. The variation captured by this boxplot is from the difference in average performance between HOPs, rather than from the variation within the 3 trials.

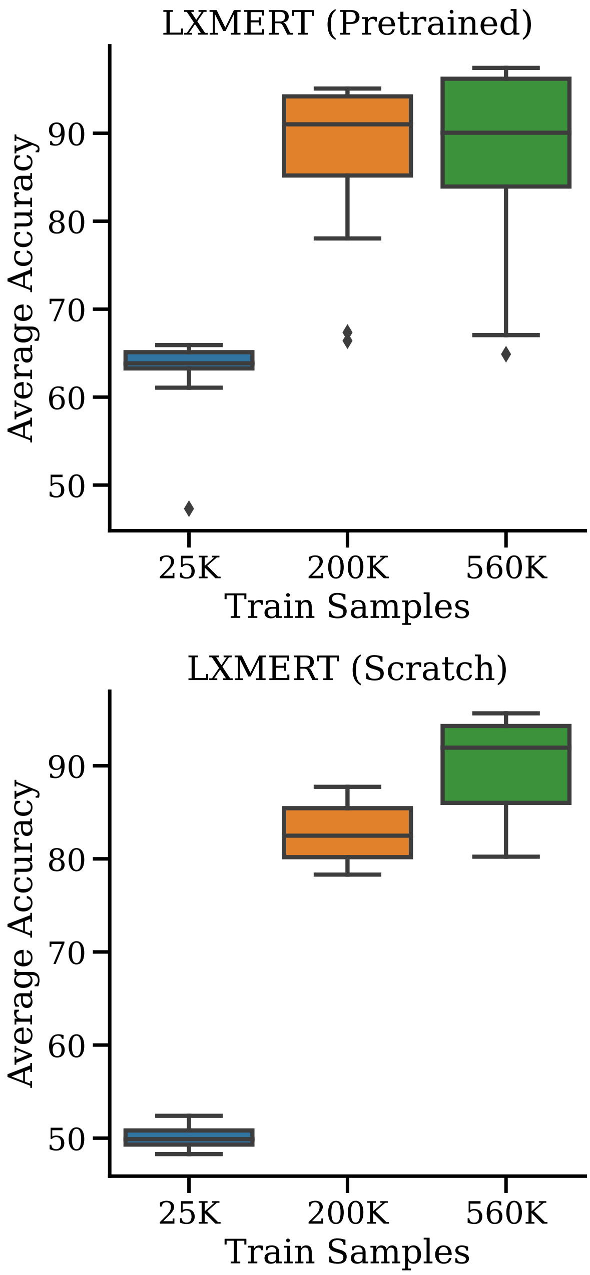

Figure 10: Box plot of **complex-OOD test** set performance on all 29 HOPs. The average performance for each HOP is produced by averaging over 3 trials. The variation captured by this boxplot is from the difference in average performance between HOPs, rather than from the variation within the 3 trials.

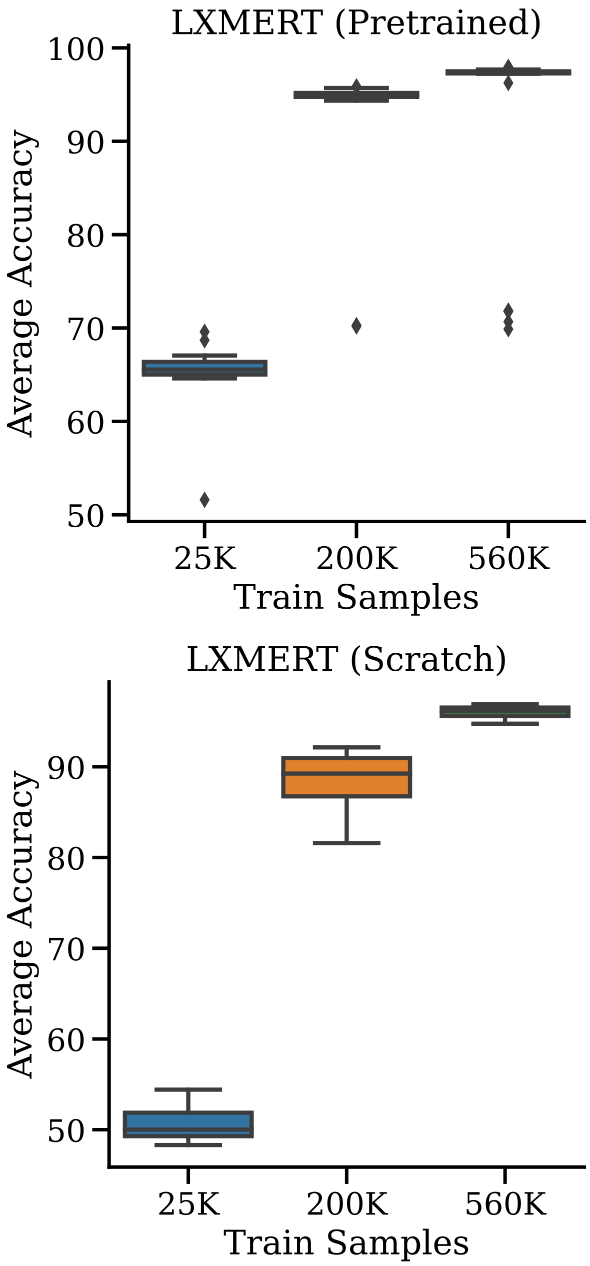

Figure 11: Box plot of **complex-IID test** set performance on all 29 HOPs. The average performance for each HOP is produced by averaging over 3 trials. The variation captured by this boxplot is from the difference in average performance between HOPs, rather than from the variation within the 3 trials.

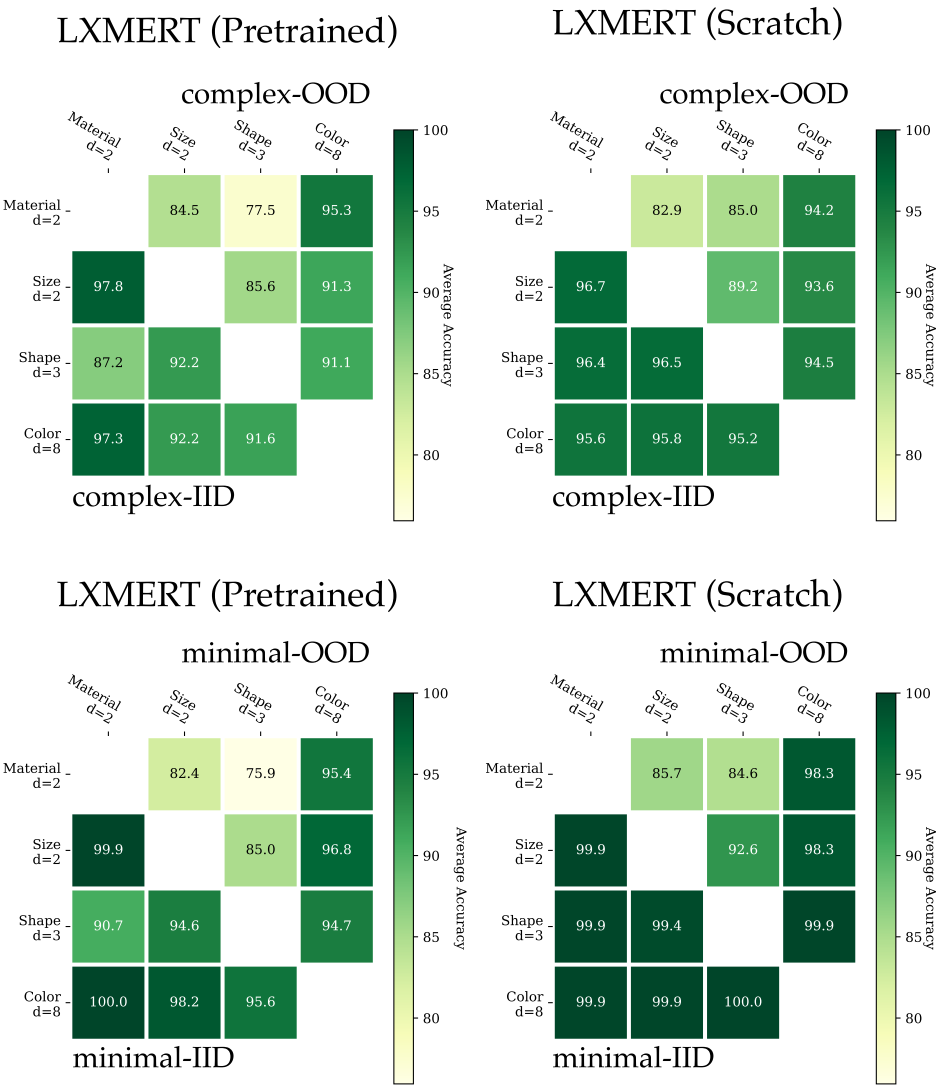

Figure 12: Model accuracies for HOP-0 through 28. Note that the LXMERT models often struggle on both IID and OOD questions when MATERIAL-SHAPE combinations are held out at train time.

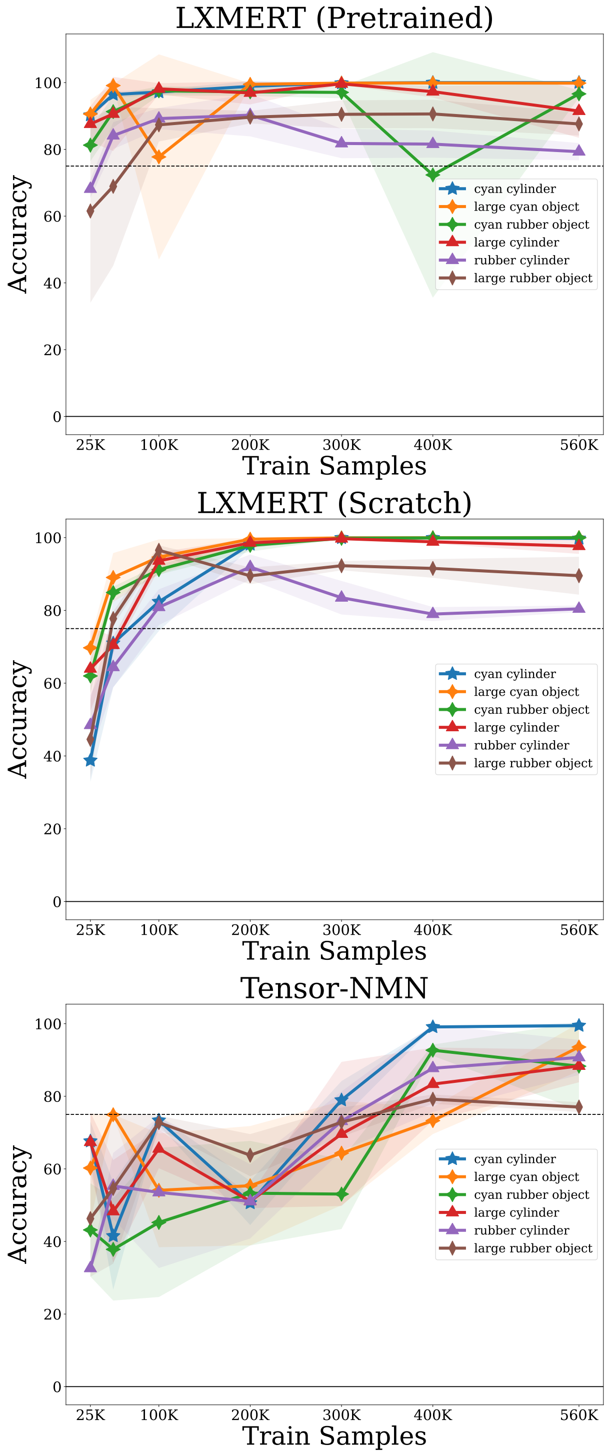

Figure 13: Average **minimal-OOD test** set Tensor-NMN performance for the first 6 HOPs over 3 trials. For comparison, we also plot the average LXMERT model performances (i.e., Fig. 8), but restricted to only the first 6 HOPs. An area corresponding to 1 standard deviation is shaded.

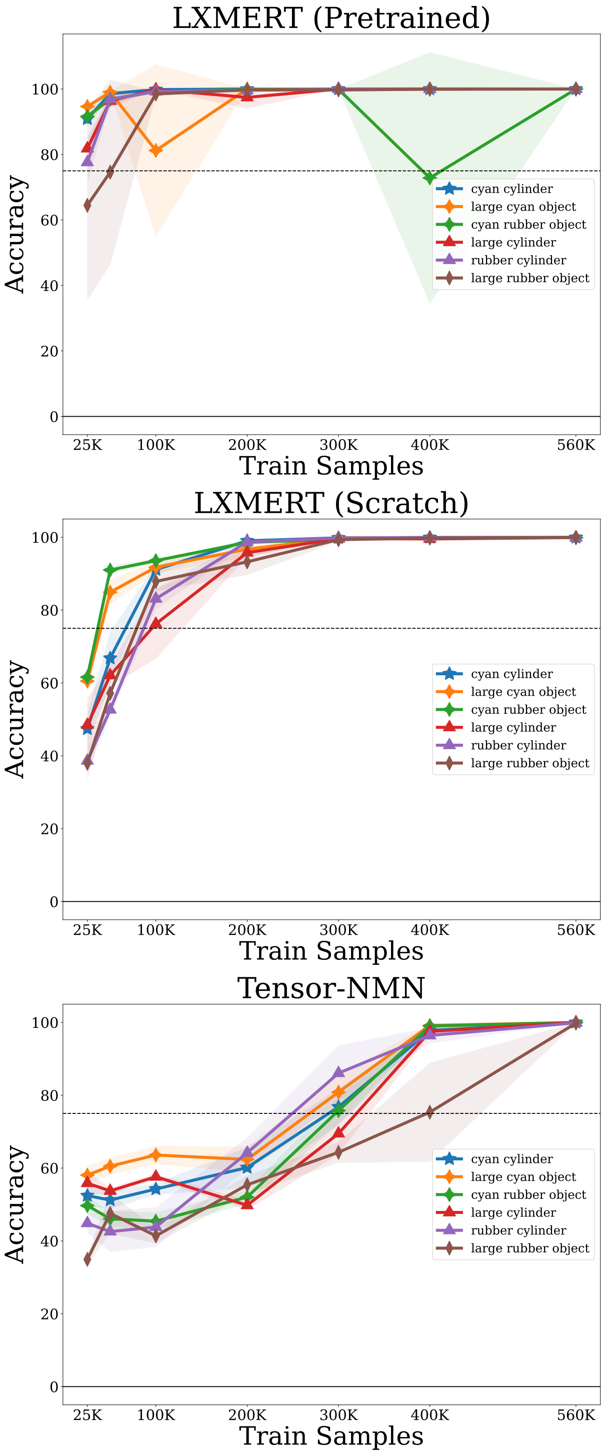

Figure 14: Average **minimal-IID test** set Tensor-NMN performance for the first 6 HOPs over 3 trials. For comparison, we also plot the average LXMERT model performances (i.e., Fig. 9), but restricted to only the first 6 HOPs. An area corresponding to 1 standard deviation is shaded.

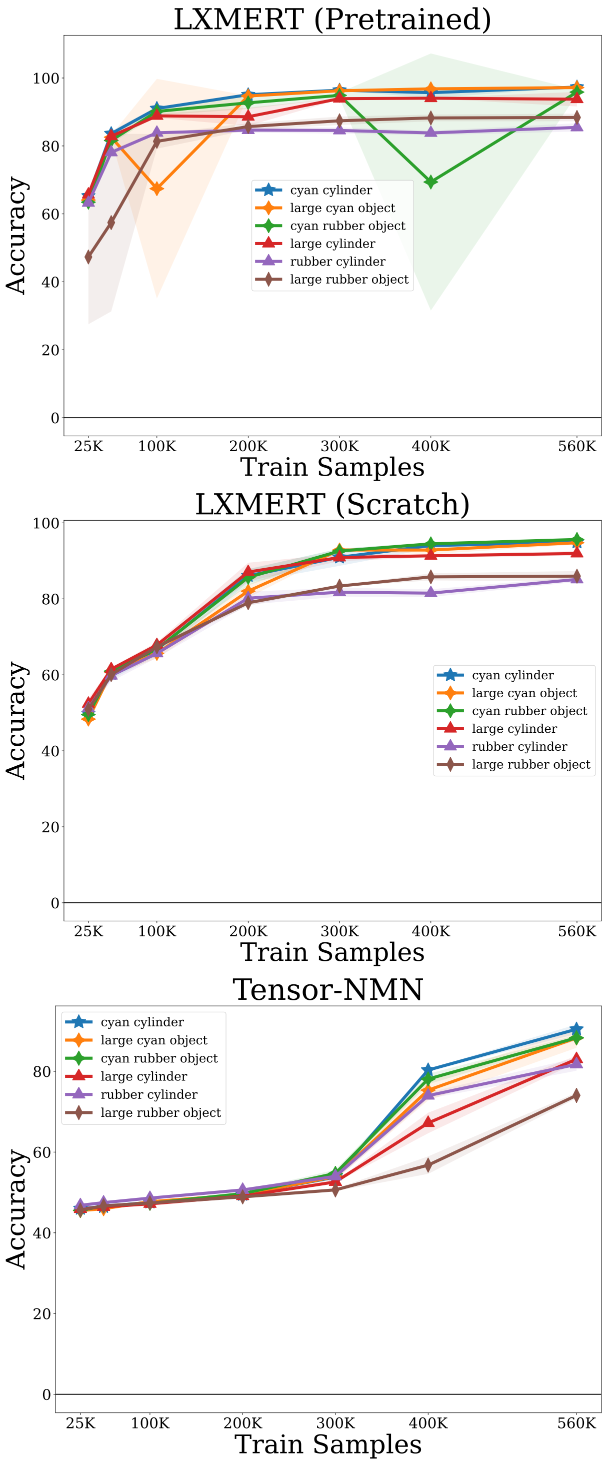

Figure 15: Average **complex-OOD test** set Tensor-NMN performance for the first 6 HOPs over 3 trials. For comparison, we also plot the average LXMERT model performances (i.e., Fig. 10), but restricted to only the first 6 HOPs. An area corresponding to 1 standard deviation is shaded.

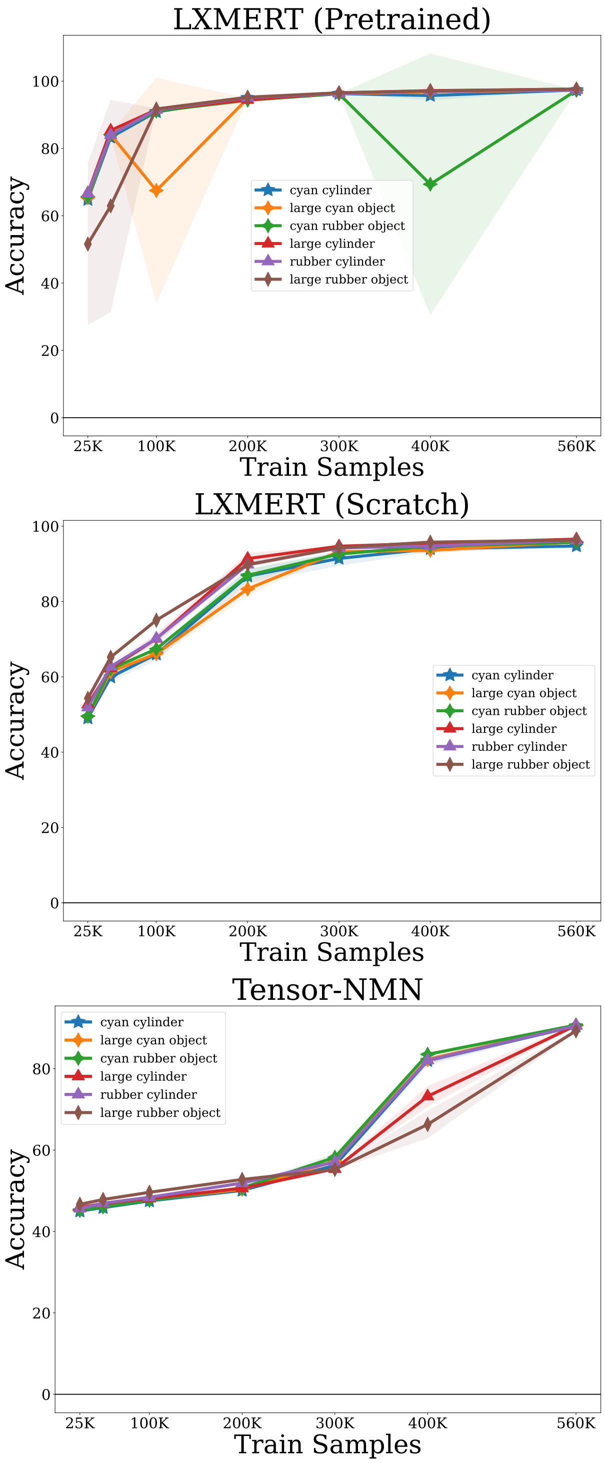

Figure 16: Average **complex-IID test** set Tensor-NMN performance for the first 6 HOPs over 3 trials. For comparison, we also plot the average LXMERT model performances (i.e., Fig. 11), but restricted to only the first 6 HOPs. An area corresponding to 1 standard deviation is shaded.

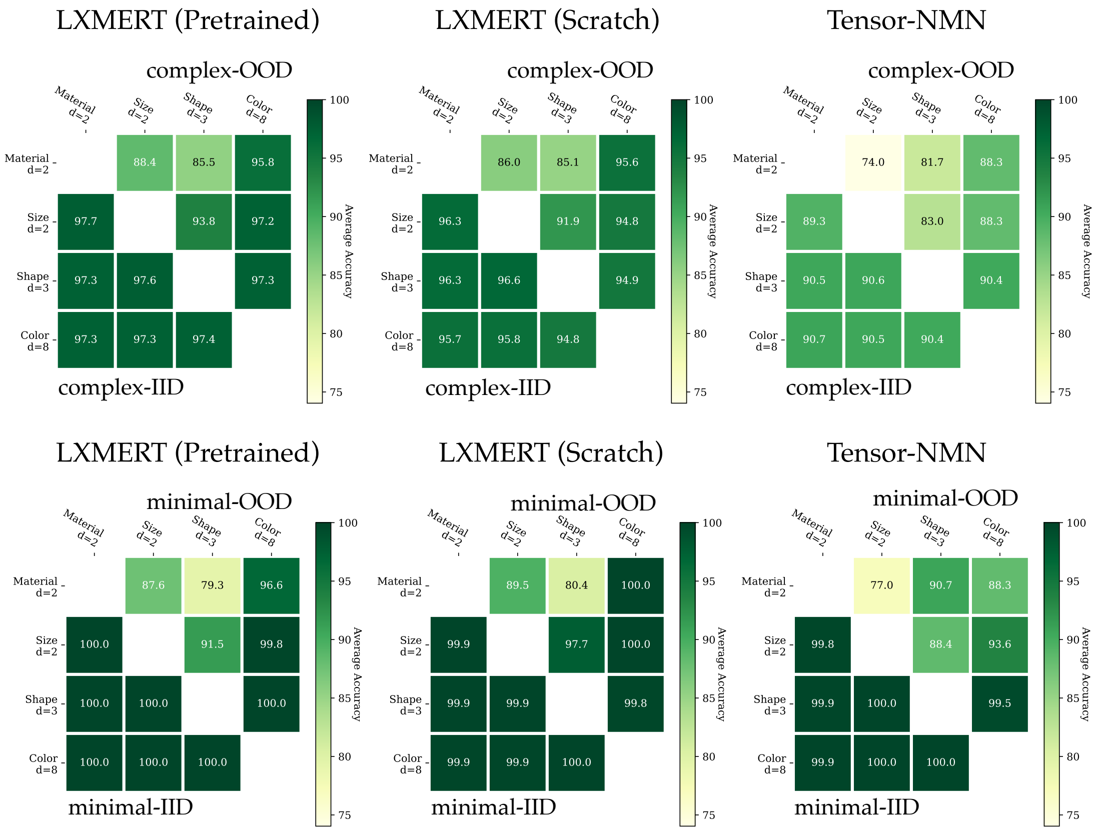

Figure 17: Model accuracies for *only* the first 6 HOPs. Note that while the LXMERT models struggle with MATERIAL-SHAPE combinations on OOD questions, Tensor-NMN does not.

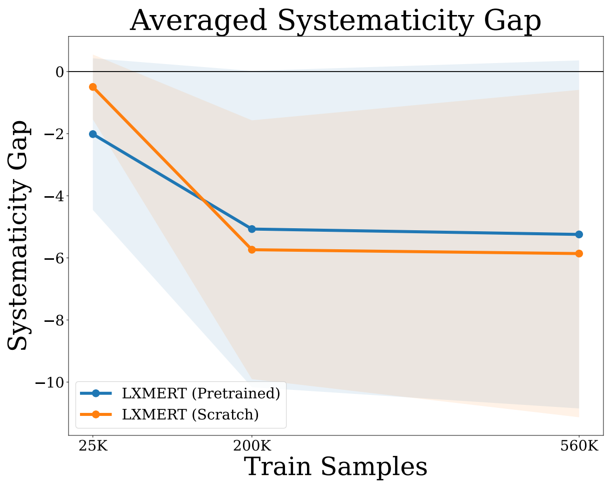

Figure 18: Average systematicity gap on **complex** examples (i.e., complex-OOD test accuracy minus complex-IID test accuracy) with 1 standard deviation; averaged over 3 runs on each of the 29 HOPs. Note that the systematicity gap plateaus, suggesting that the performance drop when generalizing to unseen combinations does not improve with additional training data.

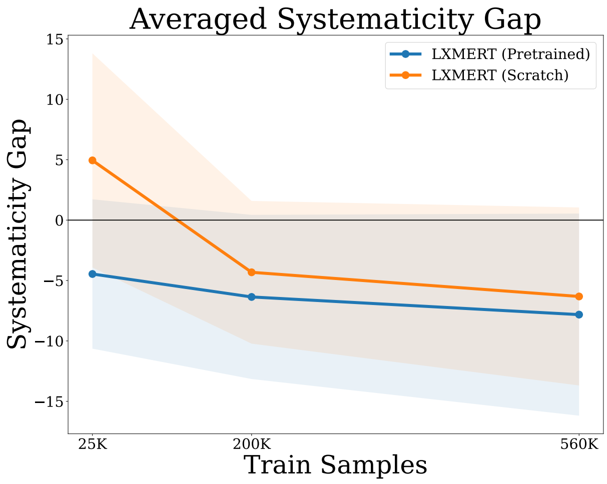

Figure 19: Average systematicity gap on **minimal** examples (i.e., minimal-OOD test accuracy minus minimal-IID test accuracy) with 1 standard deviation; averaged over 3 runs on each of the 29 HOPs.

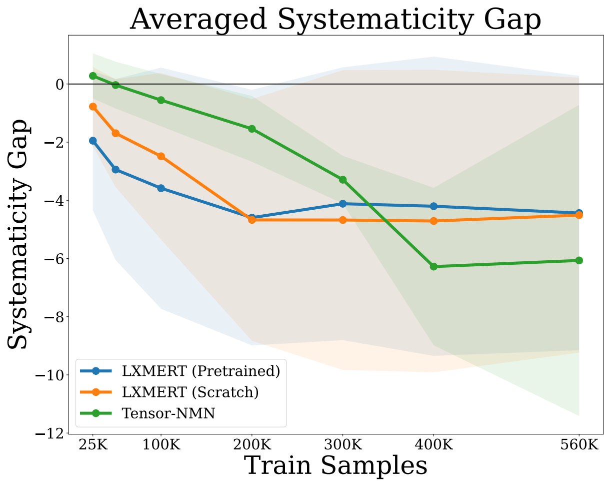

Figure 20: Average systematicity gap on **complex** examples (i.e., complex-OOD test accuracy minus complex-IID test accuracy) with 1 standard deviation; averaged over 3 runs on *only* the first 6 HOPs. Note that the systematicity gap plateaus, suggesting that the performance drop when generalizing to unseen combinations does not improve with additional training data.

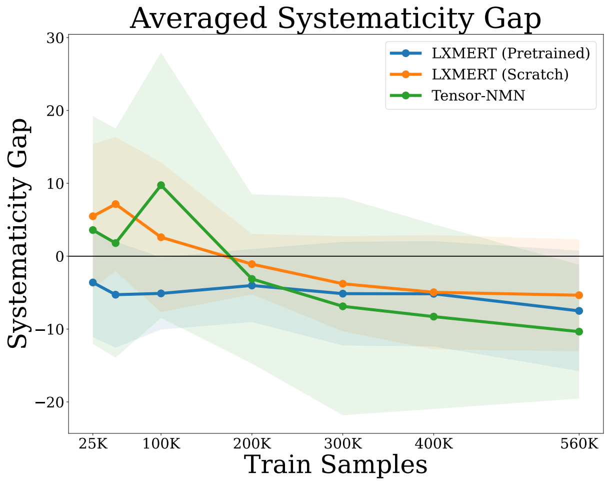

Figure 21: Average systematicity gap on **minimal** examples (i.e., minimal-OOD test accuracy minus minimal-IID test accuracy) with 1 standard deviation; averaged over 3 runs on *only* the first 6 HOPs.

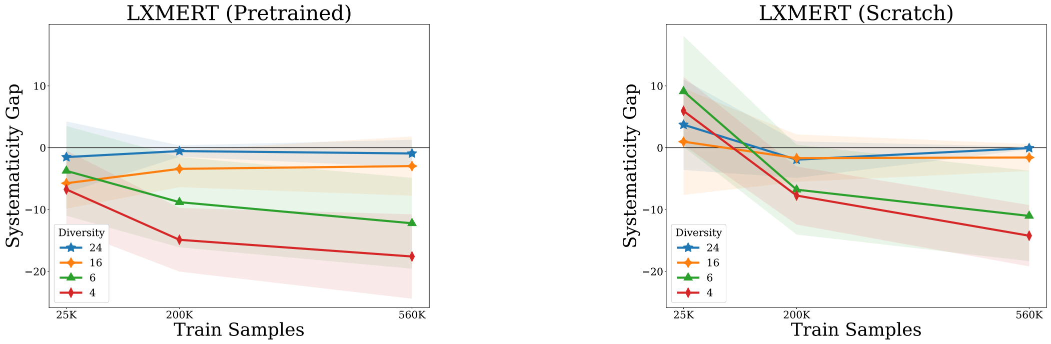

Figure 22: Systematicity gap (difference between OOD and IID model accuracy) on the **minimal** split, averaged by held-out pair (HOP) diversity over 29 HOPs, each with 3 runs.

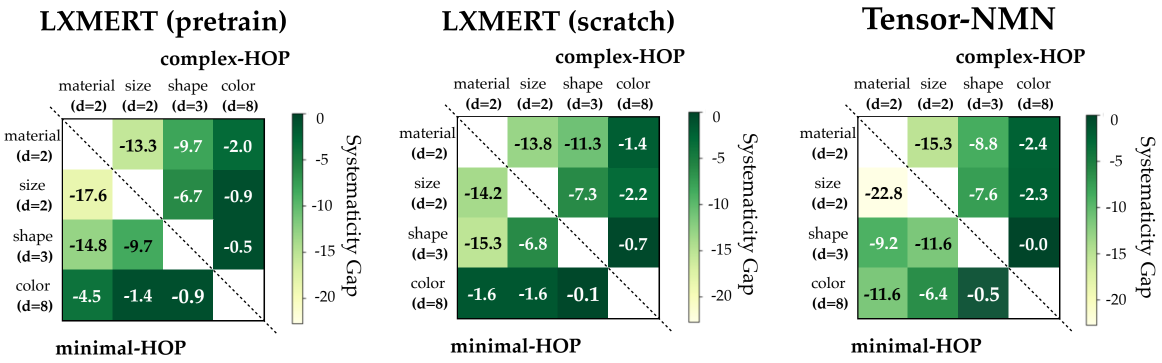

Figure 23: Systematicity gap on the complex splits (top corner) and minimal splits (bottom corner) for all models trained on 560k training examples. The systematicity gap is averaged according to the attribute types of the HOPs, all 29 HOPs for LXMERT, HOPs 0-5 for Tensor-NMN—attributes are sorted by increasing diversity on the axes (e.g., SHAPE has 2 possible values, COLOR has 8 possible values). As expected, we see a worse systematicity gap (i.e. lighter colors) in the top left (low-diversity combinations), and better systematicity gap (i.e., darker colors) in the bottom right (high-diversity combinations).

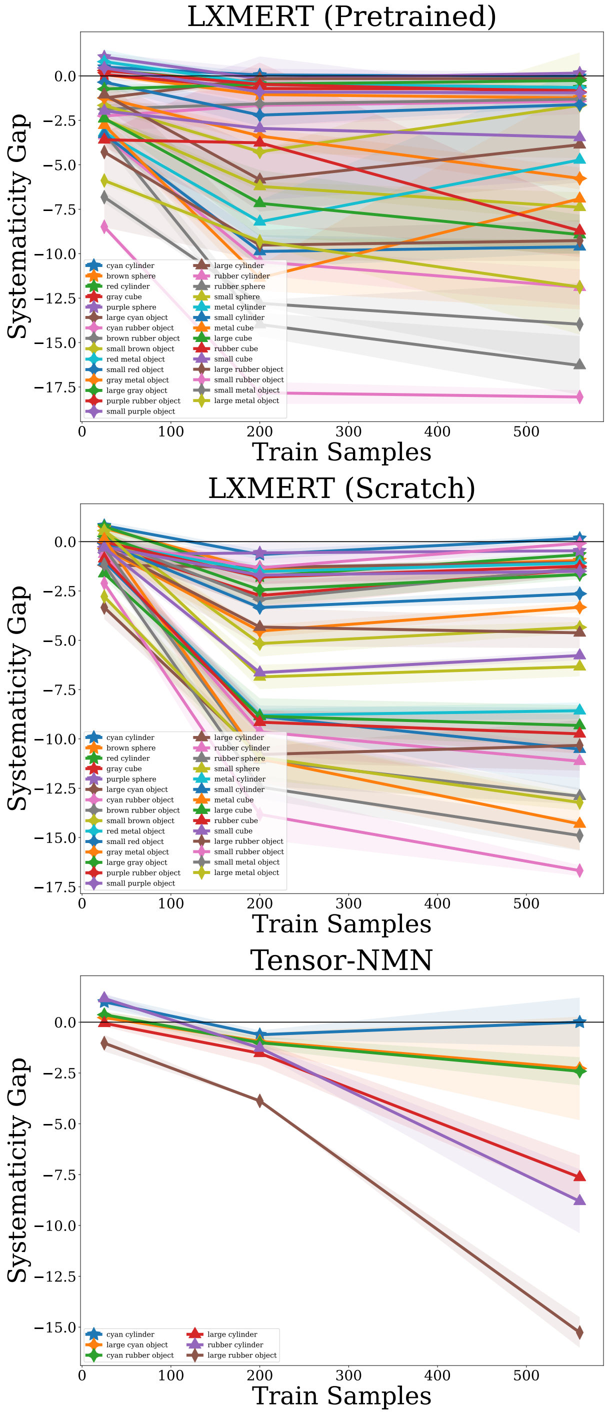

Figure 24: Systematicity gap (i.e. difference between OOD and IID model performance) for **complex** examples, averaged over 3 runs, for each HOP.

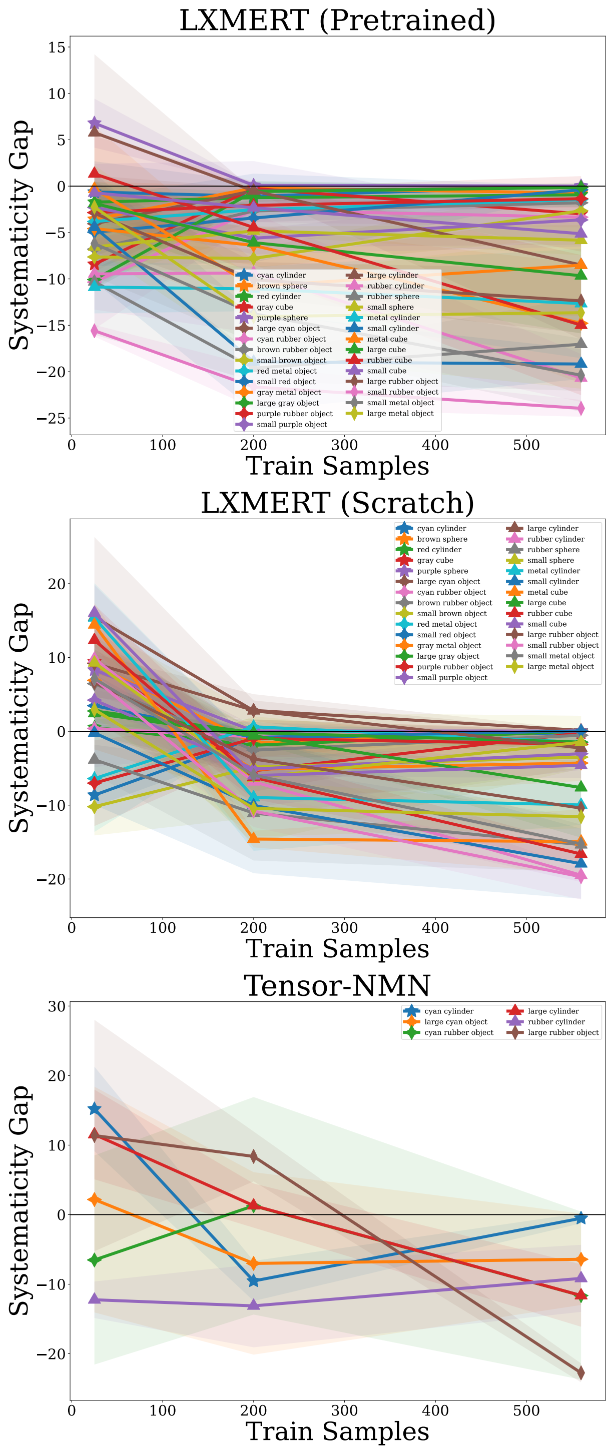

Figure 25: Systematicity gap (i.e. difference between OOD and IID model performance) for **minimal** examples, averaged over 3 runs, for each HOP.

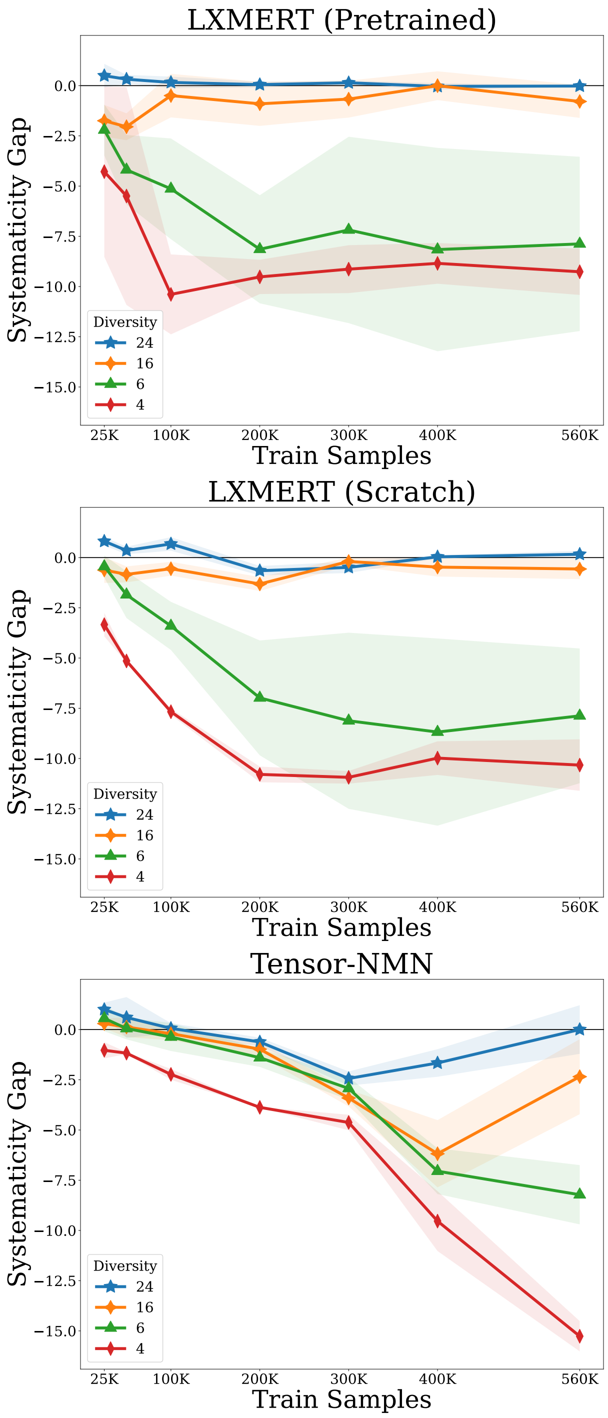

Figure 26: Systematicity gap (i.e. difference between OOD and IID model performance) for **complex** examples, averaged by HOP diversity over for the first 6 held-out attribute pairs *only*, each with 3 runs.

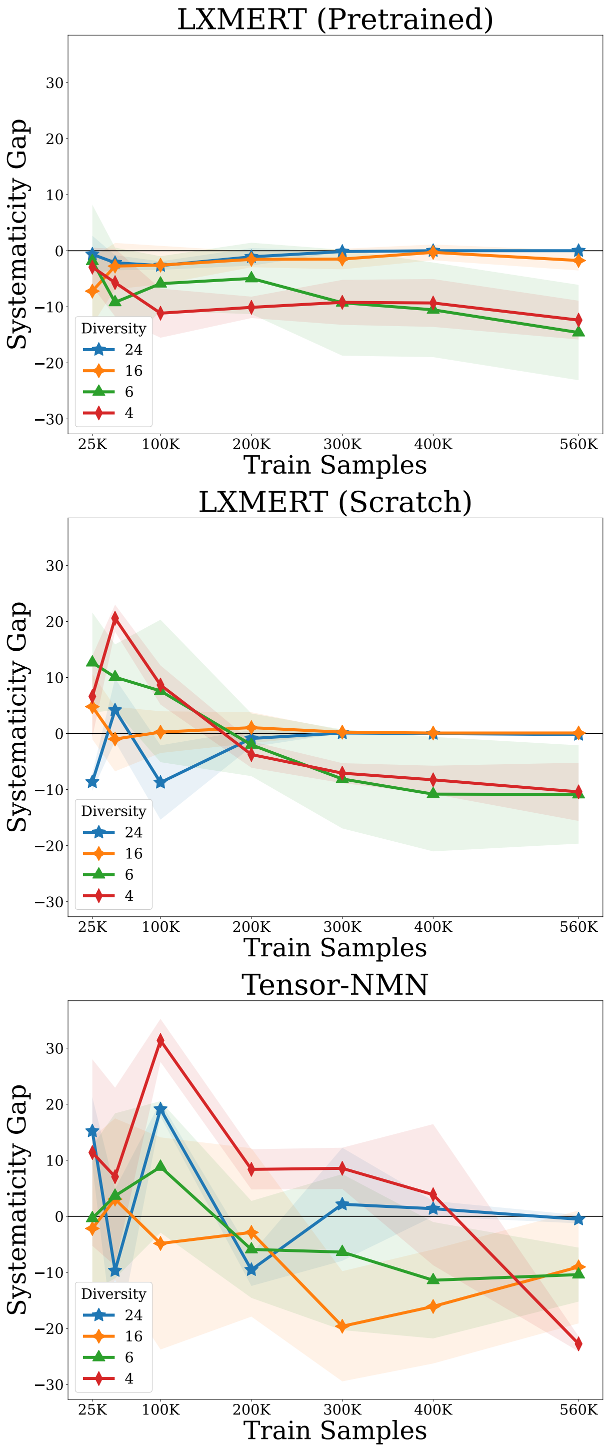

Figure 27: Systematicity gap (i.e. difference between OOD and IID model performance) for **minimal** examples, averaged by HOP diversity over for the first 6 held-out attribute pairs *only*, each with 3 runs.

HOP	Diversity	25k	200k	560k
cyan cylinder	24	64.80 ± 0.13%	95.03 ± 0.05%	97.36 ± 0.05%
brown sphere	24	65.02 ± 0.15%	95.09 ± 0.01%	97.43 ± 0.02%
red cylinder	24	65.02 ± 0.23%	95.07 ± 0.04%	96.25 ± 0.97%
gray cube	24	65.53 ± 0.23%	94.90 ± 0.13%	69.88 ± 38.90%
purple sphere	24	64.85 ± 0.52%	94.71 ± 0.03%	97.27 ± 0.12%
large cyan object	16	65.32 ± 0.22%	94.86 ± 0.11%	97.34 ± 0.05%
cyan rubber object	16	65.70 ± 0.21%	94.35 ± 0.69%	97.27 ± 0.09%
brown rubber object	16	65.55 ± 0.15%	94.88 ± 0.10%	97.33 ± 0.05%
small brown object	16	65.23 ± 0.04%	95.28 ± 0.16%	71.86 ± 36.14%
red metal object	16	64.92 ± 0.14%	95.00 ± 0.08%	97.48 ± 0.03%
small red object	16	65.19 ± 0.15%	94.71 ± 0.50%	97.33 ± 0.02%
gray metal object	16	65.31 ± 0.28%	94.75 ± 0.11%	97.29 ± 0.04%
large gray object	16	64.98 ± 0.05%	94.83 ± 0.24%	97.22 ± 0.24%
purple rubber object	16	65.14 ± 0.06%	94.85 ± 0.07%	97.31 ± 0.07%
small purple object	16	64.60 ± 0.17%	94.58 ± 0.31%	97.37 ± 0.07%
large cylinder	6	66.75 ± 0.08%	94.44 ± 0.93%	97.64 ± 0.03%
rubber cylinder	6	66.62 ± 0.20%	95.11 ± 0.08%	97.35 ± 0.22%
rubber sphere	6	66.38 ± 0.21%	95.13 ± 0.14%	97.45 ± 0.07%
small sphere	6	65.65 ± 0.28%	95.14 ± 0.16%	97.44 ± 0.04%
metal cylinder	6	66.38 ± 0.31%	95.17 ± 0.24%	71.77 ± 36.57%
small cylinder	6	67.06 ± 0.21%	95.07 ± 0.31%	97.62 ± 0.19%
metal cube	6	66.04 ± 0.41%	95.18 ± 0.10%	71.79 ± 36.61%
large cube	6	66.24 ± 0.13%	95.49 ± 0.08%	97.88 ± 0.02%
rubber cube	6	66.93 ± 0.36%	70.18 ± 35.34%	97.49 ± 0.32%
small cube	6	65.95 ± 0.07%	70.30 ± 35.03%	70.67 ± 38.16%
large rubber object	4	51.60 ± 24.05%	95.23 ± 0.15%	97.65 ± 0.05%
small rubber object	4	69.59 ± 0.18%	95.87 ± 0.08%	97.69 ± 0.27%
small metal object	4	68.69 ± 0.31%	95.84 ± 0.12%	97.91 ± 0.13%
large metal object	4	66.96 ± 0.52%	95.70 ± 0.13%	97.95 ± 0.05%

Table 3: LXMERT (Pretrained) **complex-IID** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	65.29 ± 0.48%	95.08 ± 0.15%	97.34 ± 0.08%
brown sphere	24	65.11 ± 0.08%	94.04 ± 0.40%	96.20 ± 0.22%
red cylinder	24	65.36 ± 0.11%	94.63 ± 0.08%	95.59 ± 1.32%
gray cube	24	65.60 ± 0.50%	94.19 ± 0.19%	69.15 ± 38.40%
purple sphere	24	65.92 ± 0.69%	94.55 ± 0.57%	97.43 ± 0.09%
large cyan object	16	64.08 ± 0.30%	94.70 ± 0.09%	97.19 ± 0.08%
cyan rubber object	16	63.44 ± 0.70%	92.69 ± 1.82%	95.85 ± 0.73%
brown rubber object	16	63.69 ± 0.20%	93.31 ± 0.09%	96.02 ± 0.14%
small brown object	16	63.57 ± 0.31%	91.02 ± 0.17%	70.20 ± 33.16%
red metal object	16	65.72 ± 0.68%	94.56 ± 0.26%	96.82 ± 0.26%
small red object	16	64.84 ± 0.45%	92.50 ± 1.09%	95.72 ± 0.11%
gray metal object	16	64.08 ± 0.31%	91.37 ± 0.37%	91.53 ± 0.58%
large gray object	16	64.24 ± 0.17%	94.37 ± 0.36%	96.96 ± 0.28%
purple rubber object	16	65.45 ± 0.22%	94.37 ± 0.20%	96.41 ± 0.38%
small purple object	16	65.05 ± 0.62%	93.67 ± 0.34%	96.42 ± 0.33%
large cylinder	6	65.69 ± 0.74%	88.60 ± 2.68%	93.76 ± 2.15%
rubber cylinder	6	63.26 ± 0.15%	84.66 ± 0.79%	85.46 ± 1.23%
rubber sphere	6	63.17 ± 0.57%	81.14 ± 0.77%	81.17 ± 1.60%
small sphere	6	63.23 ± 0.33%	88.92 ± 0.41%	90.06 ± 0.84%
metal cylinder	6	63.20 ± 0.64%	86.97 ± 1.39%	67.05 ± 31.47%
small cylinder	6	63.78 ± 0.21%	85.20 ± 0.91%	88.01 ± 0.20%
metal cube	6	63.27 ± 0.78%	83.82 ± 0.68%	64.88 ± 30.50%
large cube	6	63.84 ± 0.09%	88.33 ± 1.78%	88.95 ± 1.11%
rubber cube	6	63.34 ± 0.07%	66.41 ± 30.84%	88.78 ± 1.65%
small cube	6	63.98 ± 0.26%	67.35 ± 30.99%	67.21 ± 35.83%
large rubber object	4	47.32 ± 19.82%	85.71 ± 1.01%	88.39 ± 1.12%
small rubber object	4	61.10 ± 0.32%	78.04 ± 0.55%	79.62 ± 0.56%
small metal object	4	61.87 ± 0.54%	83.05 ± 0.08%	83.94 ± 2.44%
large metal object	4	61.07 ± 0.59%	86.40 ± 0.13%	86.08 ± 2.69%

Table 4: LXMERT (Pretrained) **complex-OOD** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	90.89 ± 2.49%	99.97 ± 0.02%	100.00 ± 0.00%
brown sphere	24	92.33 ± 1.49%	99.98 ± 0.01%	100.00 ± 0.00%
red cylinder	24	92.05 ± 1.98%	99.99 ± 0.00%	99.89 ± 0.16%
gray cube	24	92.57 ± 1.40%	99.95 ± 0.03%	78.36 ± 30.60%
purple sphere	24	86.66 ± 4.47%	99.91 ± 0.07%	99.99 ± 0.01%
large cyan object	16	94.65 ± 1.06%	99.97 ± 0.01%	99.98 ± 0.01%
cyan rubber object	16	91.62 ± 1.09%	99.81 ± 0.05%	99.97 ± 0.01%
brown rubber object	16	91.63 ± 1.05%	99.58 ± 0.08%	99.93 ± 0.01%
small brown object	16	90.81 ± 1.49%	99.93 ± 0.03%	91.23 ± 12.40%
red metal object	16	91.15 ± 1.33%	99.72 ± 0.02%	99.97 ± 0.01%
small red object	16	92.06 ± 0.66%	98.60 ± 1.89%	99.99 ± 0.01%
gray metal object	16	90.09 ± 1.86%	99.52 ± 0.53%	99.98 ± 0.01%
large gray object	16	94.20 ± 1.19%	99.84 ± 0.11%	99.98 ± 0.02%
purple rubber object	16	88.69 ± 2.03%	99.77 ± 0.05%	99.96 ± 0.02%
small purple object	16	93.05 ± 0.41%	99.97 ± 0.02%	99.99 ± 0.01%
large cylinder	6	81.81 ± 3.51%	97.42 ± 3.37%	99.97 ± 0.01%
rubber cylinder	6	77.60 ± 6.47%	99.61 ± 0.15%	99.99 ± 0.00%
rubber sphere	6	81.61 ± 3.88%	99.75 ± 0.07%	99.87 ± 0.02%
small sphere	6	90.59 ± 1.41%	99.93 ± 0.04%	99.93 ± 0.03%
metal cylinder	6	85.59 ± 5.81%	99.84 ± 0.10%	76.46 ± 33.26%
small cylinder	6	86.79 ± 2.68%	99.95 ± 0.03%	99.99 ± 0.01%
metal cube	6	75.06 ± 7.55%	99.53 ± 0.35%	77.36 ± 31.95%
large cube	6	89.61 ± 1.98%	99.98 ± 0.02%	100.00 ± 0.00%
rubber cube	6	73.00 ± 1.91%	85.84 ± 19.75%	99.94 ± 0.06%
small cube	6	81.08 ± 2.96%	90.02 ± 13.74%	73.28 ± 37.77%
large rubber object	4	64.46 ± 28.99%	99.74 ± 0.03%	99.98 ± 0.01%
small rubber object	4	89.38 ± 1.37%	99.85 ± 0.09%	99.99 ± 0.01%
small metal object	4	86.15 ± 2.22%	99.90 ± 0.08%	99.89 ± 0.06%
large metal object	4	85.80 ± 2.25%	99.92 ± 0.03%	99.91 ± 0.01%

Table 5: LXMERT (Pretrained) **minimal-IID** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	90.25 ± 0.82%	98.88 ± 1.58%	100.00 ± 0.00%
brown sphere	24	88.76 ± 3.74%	99.78 ± 0.18%	99.26 ± 0.46%
red cylinder	24	90.33 ± 1.04%	98.74 ± 0.64%	98.96 ± 1.47%
gray cube	24	84.15 ± 1.28%	99.70 ± 0.11%	75.37 ± 34.67%
purple sphere	24	93.45 ± 6.74%	100.00 ± 0.00%	100.00 ± 0.00%
large cyan object	16	90.60 ± 4.23%	99.48 ± 0.31%	99.84 ± 0.06%
cyan rubber object	16	81.27 ± 4.82%	97.22 ± 1.23%	96.63 ± 1.12%
brown rubber object	16	84.84 ± 2.14%	96.90 ± 1.17%	98.13 ± 1.08%
small brown object	16	83.17 ± 3.10%	92.14 ± 0.99%	88.57 ± 9.60%
red metal object	16	87.34 ± 4.08%	97.18 ± 0.62%	98.53 ± 0.76%
small red object	16	87.10 ± 3.48%	95.16 ± 6.68%	99.60 ± 0.40%
gray metal object	16	85.52 ± 1.83%	93.13 ± 2.58%	85.20 ± 6.46%
large gray object	16	84.13 ± 2.25%	99.25 ± 1.07%	99.84 ± 0.15%
purple rubber object	16	85.83 ± 4.27%	97.70 ± 0.62%	98.61 ± 0.95%
small purple object	16	90.75 ± 1.31%	94.37 ± 0.98%	96.35 ± 2.66%
large cylinder	6	87.58 ± 5.31%	96.91 ± 3.45%	91.47 ± 8.00%
rubber cylinder	6	68.14 ± 2.73%	90.25 ± 6.35%	79.31 ± 2.58%
rubber sphere	6	71.30 ± 8.29%	80.13 ± 1.34%	82.83 ± 5.02%
small sphere	6	84.04 ± 3.47%	95.10 ± 0.49%	94.10 ± 1.60%
metal cylinder	6	74.71 ± 6.63%	88.76 ± 2.50%	63.80 ± 27.45%
small cylinder	6	82.37 ± 4.19%	81.02 ± 5.24%	80.82 ± 1.63%
metal cube	6	74.75 ± 5.48%	88.72 ± 2.93%	68.84 ± 28.41%
large cube	6	87.75 ± 3.30%	93.89 ± 4.10%	90.34 ± 6.22%
rubber cube	6	74.32 ± 4.40%	81.38 ± 15.25%	84.96 ± 7.58%
small cube	6	80.35 ± 0.17%	87.70 ± 9.12%	68.15 ± 39.08%
large rubber object	4	61.54 ± 27.48%	89.64 ± 1.88%	87.61 ± 3.48%
small rubber object	4	73.79 ± 1.94%	78.21 ± 2.26%	76.04 ± 0.91%
small metal object	4	79.95 ± 3.57%	86.15 ± 3.16%	79.51 ± 3.97%
large metal object	4	83.54 ± 4.87%	85.86 ± 4.14%	86.27 ± 8.20%

Table 6: LXMERT (Pretrained) **minimal-OOD** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	49.05 ± 0.41%	86.74 ± 1.90%	94.75 ± 0.54%
brown sphere	24	48.77 ± 0.23%	88.69 ± 0.25%	95.60 ± 0.23%
red cylinder	24	49.44 ± 0.29%	85.45 ± 1.99%	95.56 ± 0.29%
gray cube	24	49.41 ± 0.64%	81.59 ± 1.95%	95.02 ± 0.42%
purple sphere	24	49.60 ± 0.94%	86.01 ± 5.30%	95.13 ± 0.42%
large cyan object	16	49.54 ± 0.74%	83.34 ± 1.77%	95.83 ± 0.30%
cyan rubber object	16	49.59 ± 0.70%	86.97 ± 1.66%	95.71 ± 0.33%
brown rubber object	16	49.16 ± 0.36%	88.87 ± 1.06%	95.52 ± 0.65%
small brown object	16	49.22 ± 0.34%	87.78 ± 2.15%	96.21 ± 0.17%
red metal object	16	49.29 ± 0.27%	89.25 ± 1.86%	95.70 ± 0.14%
small red object	16	49.13 ± 0.47%	87.76 ± 1.07%	95.53 ± 0.26%
gray metal object	16	48.95 ± 0.53%	85.17 ± 2.57%	95.88 ± 0.27%
large gray object	16	50.06 ± 0.92%	82.79 ± 4.83%	95.77 ± 0.07%
purple rubber object	16	48.31 ± 0.08%	86.51 ± 0.25%	95.31 ± 0.14%
small purple object	16	49.59 ± 0.49%	88.13 ± 1.41%	95.77 ± 0.09%
large cylinder	6	52.66 ± 1.68%	91.39 ± 1.48%	96.56 ± 0.15%
rubber cylinder	6	51.87 ± 0.88%	89.82 ± 0.64%	96.25 ± 0.25%
rubber sphere	6	50.21 ± 0.71%	90.07 ± 0.69%	96.24 ± 0.08%
small sphere	6	50.01 ± 0.58%	91.56 ± 0.89%	96.12 ± 0.07%
metal cylinder	6	51.87 ± 0.78%	90.57 ± 1.05%	96.58 ± 0.08%
small cylinder	6	52.01 ± 1.18%	91.29 ± 1.87%	96.53 ± 0.06%
metal cube	6	50.34 ± 0.33%	90.57 ± 1.09%	96.29 ± 0.15%
large cube	6	52.44 ± 0.90%	91.34 ± 0.92%	96.72 ± 0.13%
rubber cube	6	50.38 ± 0.76%	91.13 ± 0.85%	96.45 ± 0.15%
small cube	6	50.69 ± 0.58%	91.75 ± 0.47%	96.68 ± 0.17%
large rubber object	4	54.28 ± 0.47%	89.77 ± 0.72%	96.31 ± 0.20%
small rubber object	4	53.33 ± 0.90%	92.14 ± 0.65%	96.91 ± 0.24%
small metal object	4	51.94 ± 0.49%	90.97 ± 0.69%	96.84 ± 0.28%
large metal object	4	54.42 ± 0.66%	89.87 ± 2.50%	96.77 ± 0.17%

Table 7: LXMERT (Scratch) **complex-IID** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	49.86 ± 0.31%	86.08 ± 1.80%	94.92 ± 0.68%
brown sphere	24	49.46 ± 0.04%	87.24 ± 0.40%	94.64 ± 0.38%
red cylinder	24	50.20 ± 0.41%	83.70 ± 2.24%	94.90 ± 0.19%
gray cube	24	49.23 ± 0.32%	78.86 ± 1.74%	93.60 ± 0.55%
purple sphere	24	48.94 ± 0.80%	85.44 ± 5.63%	94.67 ± 0.58%
large cyan object	16	48.35 ± 0.43%	82.03 ± 1.74%	94.78 ± 0.48%
cyan rubber object	16	49.54 ± 0.47%	85.65 ± 2.12%	95.63 ± 0.24%
brown rubber object	16	49.31 ± 0.49%	85.95 ± 1.48%	94.17 ± 1.01%
small brown object	16	49.78 ± 0.26%	82.61 ± 2.81%	91.87 ± 0.37%
red metal object	16	49.21 ± 0.37%	87.74 ± 2.22%	94.61 ± 0.09%
small red object	16	49.04 ± 0.09%	84.42 ± 1.02%	92.90 ± 0.67%
gray metal object	16	48.60 ± 0.35%	80.64 ± 2.27%	92.56 ± 0.15%
large gray object	16	50.33 ± 0.75%	80.34 ± 4.06%	94.11 ± 0.30%
purple rubber object	16	48.29 ± 0.38%	84.71 ± 0.59%	94.06 ± 0.31%
small purple object	16	49.33 ± 0.53%	86.43 ± 1.87%	94.27 ± 0.10%
large cylinder	6	52.40 ± 1.33%	87.06 ± 2.46%	91.94 ± 0.63%
rubber cylinder	6	51.24 ± 0.48%	80.18 ± 1.71%	85.12 ± 0.67%
rubber sphere	6	49.89 ± 0.55%	78.99 ± 1.82%	83.34 ± 0.48%
small sphere	6	50.54 ± 0.41%	84.70 ± 1.24%	89.78 ± 0.56%
metal cylinder	6	50.87 ± 0.72%	81.76 ± 0.75%	88.00 ± 0.49%
small cylinder	6	51.01 ± 1.12%	82.43 ± 2.17%	86.01 ± 2.06%
metal cube	6	50.47 ± 0.52%	79.56 ± 1.94%	81.98 ± 1.20%
large cube	6	50.83 ± 0.72%	82.49 ± 1.37%	87.40 ± 1.06%
rubber cube	6	49.52 ± 0.33%	81.98 ± 1.08%	86.71 ± 0.90%
small cube	6	50.39 ± 0.89%	85.11 ± 0.51%	90.91 ± 0.09%
large rubber object	4	50.94 ± 0.22%	78.98 ± 0.83%	85.98 ± 1.34%
small rubber object	4	51.22 ± 0.87%	78.31 ± 1.62%	80.23 ± 0.39%
small metal object	4	50.78 ± 0.36%	78.53 ± 0.93%	81.94 ± 0.49%
large metal object	4	51.63 ± 0.47%	78.94 ± 2.44%	83.54 ± 0.42%

Table 8: LXMERT (Scratch) **complex-OOD** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	47.40 ± 4.40%	99.02 ± 0.54%	99.96 ± 0.01%
brown sphere	24	48.37 ± 2.65%	98.74 ± 0.45%	99.97 ± 0.03%
red cylinder	24	60.03 ± 5.11%	98.32 ± 1.54%	99.95 ± 0.03%
gray cube	24	60.73 ± 3.48%	98.72 ± 0.48%	99.93 ± 0.03%
purple sphere	24	49.28 ± 5.04%	99.44 ± 0.21%	99.96 ± 0.03%
large cyan object	16	60.52 ± 3.26%	96.72 ± 2.22%	99.87 ± 0.10%
cyan rubber object	16	61.60 ± 1.37%	98.60 ± 0.32%	99.89 ± 0.06%
brown rubber object	16	62.04 ± 5.68%	99.53 ± 0.04%	99.70 ± 0.17%
small brown object	16	55.37 ± 3.64%	98.73 ± 0.74%	99.80 ± 0.16%
red metal object	16	60.21 ± 3.89%	98.31 ± 0.29%	99.95 ± 0.03%
small red object	16	66.29 ± 2.51%	99.23 ± 0.34%	99.82 ± 0.22%
gray metal object	16	53.61 ± 0.64%	98.51 ± 0.47%	99.97 ± 0.02%
large gray object	16	49.47 ± 3.32%	99.36 ± 0.11%	99.95 ± 0.00%
purple rubber object	16	57.13 ± 5.64%	98.22 ± 0.77%	99.92 ± 0.04%
small purple object	16	62.36 ± 4.10%	99.35 ± 0.44%	99.97 ± 0.03%
large cylinder	6	48.47 ± 7.39%	95.77 ± 1.04%	99.92 ± 0.07%
rubber cylinder	6	38.64 ± 3.31%	98.71 ± 0.73%	99.90 ± 0.04%
rubber sphere	6	39.95 ± 6.05%	98.12 ± 0.59%	99.72 ± 0.05%
small sphere	6	48.61 ± 3.31%	99.13 ± 0.52%	97.38 ± 2.38%
metal cylinder	6	38.36 ± 1.81%	94.38 ± 2.12%	99.96 ± 0.00%
small cylinder	6	39.51 ± 5.54%	96.51 ± 1.99%	99.97 ± 0.01%
metal cube	6	40.55 ± 4.83%	99.11 ± 0.27%	99.92 ± 0.02%
large cube	6	43.91 ± 4.48%	99.24 ± 0.95%	99.97 ± 0.01%
rubber cube	6	48.91 ± 0.93%	98.90 ± 0.60%	99.91 ± 0.08%
small cube	6	36.78 ± 1.94%	99.68 ± 0.37%	99.88 ± 0.15%
large rubber object	4	37.95 ± 4.58%	93.24 ± 3.61%	99.93 ± 0.02%
small rubber object	4	44.15 ± 1.84%	96.51 ± 1.36%	99.81 ± 0.10%
small metal object	4	43.83 ± 1.89%	94.47 ± 1.34%	99.94 ± 0.08%
large metal object	4	44.12 ± 4.62%	99.05 ± 0.76%	99.93 ± 0.03%

Table 9: LXMERT (Scratch) **minimal-IID** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	200k	560k
cyan cylinder	24	38.76 ± 5.93%	98.14 ± 1.00%	99.78 ± 0.32%
brown sphere	24	57.37 ± 4.03%	97.17 ± 2.20%	100.00 ± 0.00%
red cylinder	24	60.57 ± 5.86%	96.43 ± 3.01%	100.00 ± 0.00%
gray cube	24	70.16 ± 2.97%	93.38 ± 3.11%	99.70 ± 0.28%
purple sphere	24	57.59 ± 8.41%	99.48 ± 0.74%	100.00 ± 0.00%
large cyan object	16	69.72 ± 1.99%	99.56 ± 0.30%	100.00 ± 0.00%
cyan rubber object	16	61.98 ± 4.66%	97.86 ± 1.69%	99.96 ± 0.06%
brown rubber object	16	68.49 ± 4.69%	96.98 ± 1.46%	99.17 ± 0.70%
small brown object	16	45.16 ± 7.22%	93.89 ± 6.27%	96.31 ± 1.35%
red metal object	16	53.81 ± 7.51%	98.93 ± 0.93%	98.45 ± 0.83%
small red object	16	69.76 ± 4.76%	98.41 ± 0.66%	99.88 ± 0.10%
gray metal object	16	60.52 ± 9.88%	93.37 ± 4.08%	95.67 ± 2.43%
large gray object	16	52.22 ± 6.42%	99.17 ± 0.59%	98.49 ± 0.99%
purple rubber object	16	50.12 ± 6.01%	97.26 ± 2.13%	98.17 ± 2.00%
small purple object	16	66.59 ± 6.63%	94.25 ± 0.62%	96.94 ± 3.29%
large cylinder	6	63.96 ± 11.55%	98.57 ± 1.29%	97.66 ± 2.14%
rubber cylinder	6	48.46 ± 8.28%	91.89 ± 3.47%	80.42 ± 0.91%
rubber sphere	6	36.09 ± 5.65%	87.04 ± 6.71%	84.36 ± 3.71%
small sphere	6	57.90 ± 5.78%	92.84 ± 8.36%	95.91 ± 1.82%
metal cylinder	6	53.75 ± 6.33%	85.42 ± 5.21%	89.99 ± 2.71%
small cylinder	6	39.30 ± 13.90%	86.47 ± 7.28%	82.04 ± 4.75%
metal cube	6	54.99 ± 5.96%	84.52 ± 1.40%	84.97 ± 4.35%
large cube	6	46.34 ± 3.32%	98.65 ± 0.49%	92.34 ± 6.77%
rubber cube	6	61.26 ± 5.22%	92.67 ± 2.57%	83.29 ± 1.18%
small cube	6	52.83 ± 5.33%	93.68 ± 4.05%	95.27 ± 0.69%
large rubber object	4	44.57 ± 10.97%	89.50 ± 2.31%	89.53 ± 5.20%
small rubber object	4	51.26 ± 4.02%	85.70 ± 0.79%	80.11 ± 3.01%
small metal object	4	50.97 ± 6.85%	88.53 ± 4.26%	84.69 ± 2.55%
large metal object	4	47.07 ± 3.88%	88.61 ± 3.05%	88.37 ± 1.54%

Table 10: LXMERT (Scratch) **minimal-OOD** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	50k	100k	200k	300k	400k	560k
cyan cylinder	24	45.01 ± 0.06%	45.87 ± 0.20%	47.51 ± 0.10%	50.08 ± 0.11%	56.24 ± 0.51%	81.98 ± 0.88%	90.43 ± 0.14%
large cyan object	16	45.28 ± 0.08%	46.26 ± 0.13%	47.75 ± 0.22%	50.38 ± 0.21%	57.26 ± 0.70%	82.27 ± 1.07%	90.54 ± 0.22%
cyan rubber object	16	45.21 ± 0.07%	46.09 ± 0.07%	47.67 ± 0.15%	50.71 ± 0.30%	58.21 ± 1.11%	83.52 ± 0.39%	90.72 ± 0.15%
large cylinder	6	45.96 ± 0.09%	46.86 ± 0.11%	48.11 ± 0.22%	50.62 ± 0.33%	55.39 ± 0.25%	73.21 ± 2.52%	90.64 ± 0.23%
rubber cylinder	6	45.64 ± 0.07%	46.93 ± 0.20%	48.40 ± 0.13%	51.89 ± 0.09%	57.07 ± 1.20%	82.07 ± 1.39%	90.54 ± 0.03%
large rubber object	4	46.68 ± 0.14%	47.84 ± 0.13%	49.62 ± 0.18%	52.78 ± 0.12%	55.26 ± 0.24%	66.30 ± 3.40%	89.30 ± 0.29%

Table 11: Tensor-NMN **complex-IID** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	50k	100k	200k	300k	400k	560k
cyan cylinder	24	46.01 ± 0.40%	46.47 ± 0.82%	47.57 ± 0.22%	49.46 ± 0.32%	53.81 ± 0.16%	80.31 ± 0.26%	90.43 ± 1.18%
large cyan object	16	45.50 ± 0.21%	45.98 ± 0.46%	47.71 ± 0.14%	49.44 ± 0.51%	53.94 ± 0.81%	75.33 ± 2.55%	88.26 ± 2.44%
cyan rubber object	16	45.57 ± 0.15%	46.62 ± 0.03%	47.30 ± 0.26%	49.70 ± 0.56%	54.72 ± 0.91%	78.09 ± 1.29%	88.30 ± 0.68%
large cylinder	6	45.90 ± 0.15%	46.42 ± 0.13%	47.18 ± 0.53%	49.09 ± 0.49%	52.61 ± 0.21%	67.20 ± 2.55%	83.00 ± 0.86%
rubber cylinder	6	46.81 ± 0.17%	47.48 ± 0.21%	48.60 ± 0.37%	50.62 ± 0.20%	53.99 ± 0.91%	73.98 ± 0.79%	81.73 ± 1.59%
large rubber object	4	45.65 ± 0.35%	46.67 ± 0.09%	47.39 ± 0.21%	48.90 ± 0.12%	50.63 ± 0.30%	56.76 ± 2.17%	74.04 ± 0.71%

Table 12: Tensor-NMN **complex-OOD** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	50k	100k	200k	300k	400k	560k
cyan cylinder	24	52.43 ± 2.95%	51.22 ± 2.45%	54.24 ± 1.57%	60.14 ± 5.56%	76.80 ± 4.99%	97.75 ± 1.27%	100.00 ± 0.00%
large cyan object	16	58.06 ± 2.50%	60.52 ± 2.34%	63.58 ± 2.52%	62.37 ± 3.52%	80.83 ± 4.52%	99.15 ± 0.49%	99.99 ± 0.01%
cyan rubber object	16	49.69 ± 2.29%	46.07 ± 2.12%	45.42 ± 2.43%	52.02 ± 1.85%	75.78 ± 3.72%	99.01 ± 0.72%	99.93 ± 0.01%
large cylinder	6	55.84 ± 3.04%	53.73 ± 0.89%	57.62 ± 0.88%	49.78 ± 1.33%	69.48 ± 5.76%	97.50 ± 1.03%	99.99 ± 0.00%
rubber cylinder	6	44.85 ± 2.73%	42.56 ± 5.64%	43.80 ± 5.43%	64.17 ± 4.29%	86.06 ± 7.60%	96.40 ± 2.19%	99.87 ± 0.07%
large rubber object	4	34.92 ± 1.40%	47.56 ± 5.53%	41.39 ± 2.08%	55.39 ± 2.52%	64.32 ± 2.90%	75.34 ± 13.59%	99.78 ± 0.14%

Table 13: Tensor-NMN **minimal-IID** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

HOP	Diversity	25k	50k	100k	200k	300k	400k	560k
cyan cylinder	24	67.63 ± 5.43%	41.52 ± 14.81%	73.36 ± 1.21%	50.60 ± 6.08%	78.94 ± 5.26%	99.11 ± 0.18%	99.48 ± 0.74%
large cyan object	16	60.24 ± 15.01%	74.92 ± 0.06%	54.09 ± 15.68%	55.36 ± 16.45%	64.33 ± 14.35%	73.29 ± 3.71%	93.57 ± 6.58%
cyan rubber object	16	43.17 ± 12.85%	37.82 ± 14.10%	45.24 ± 20.55%	53.29 ± 14.42%	53.06 ± 9.63%	92.70 ± 1.56%	88.29 ± 12.08%
large cylinder	6	67.37 ± 8.26%	48.28 ± 14.17%	65.53 ± 5.37%	51.08 ± 1.88%	69.62 ± 19.86%	83.36 ± 10.07%	88.36 ± 4.50%
rubber cylinder	6	32.60 ± 0.65%	55.30 ± 9.10%	53.52 ± 20.84%	51.05 ± 10.23%	73.15 ± 8.19%	87.73 ± 12.04%	90.70 ± 4.84%
large rubber object	4	46.30 ± 16.02%	54.62 ± 20.75%	72.76 ± 1.91%	63.77 ± 5.81%	72.88 ± 4.02%	79.20 ± 1.28%	77.03 ± 1.35%

Table 14: Tensor-NMN **minimal-OOD** average accuracy and standard deviation over 3 runs with different random seeds. Average accuracies are reported for each HOP (row) and each training set size (column).

Diversity	25k	200k	560k
24	$0.41 \pm 0.48\%$	$-0.46 \pm 0.51\%$	$-0.50 \pm 0.59\%$
16	$-0.78 \pm 1.08\%$	$-1.55 \pm 1.39\%$	$-1.47 \pm 1.82\%$
6	$-2.72 \pm 0.84\%$	$-7.98 \pm 3.95\%$	$-8.18 \pm 4.69\%$
4	$-6.37 \pm 2.62\%$	$-12.36 \pm 3.49\%$	$-13.29 \pm 3.72\%$

Table 15: LXMERT (Pretrained) **complex** systematicity gap (complex-OOD accuracy minus complex-IID accuracy). Average systematicity gap and standard deviation are on the differences, over all 3 runs (with different random seeds) of all HOPs with the stated diversity. Average accuracies are reported for each diversity (row) and each training set size (column).

Diversity	25k	200k	560k
24	$-1.51 \pm 5.76\%$	$-0.54 \pm 0.93\%$	$-0.93 \pm 2.22\%$
16	$-5.74 \pm 4.13\%$	$-3.42 \pm 2.96\%$	$-2.97 \pm 4.80\%$
6	$-3.74 \pm 7.26\%$	$-8.80 \pm 7.31\%$	$-12.22 \pm 7.35\%$
4	$-6.74 \pm 6.30\%$	$-14.89 \pm 5.15\%$	$-17.59 \pm 6.83\%$

Table 16: LXMERT (Pretrained) **minimal** systematicity gap (minimal-OOD accuracy minus minimal-IID accuracy). Average systematicity gap and standard deviation are on the differences, over all 3 runs (with different random seeds) of all HOPs with the stated diversity. Average accuracies are reported for each diversity (row) and each training set size (column).

Diversity	25k	200k	560k
24	$0.28 \pm 0.63\%$	$-1.43 \pm 0.90\%$	$-0.67 \pm 0.57\%$
16	$-0.11 \pm 0.55\%$	$-2.60 \pm 1.38\%$	$-1.83 \pm 1.22\%$
6	$-0.53 \pm 0.71\%$	$-8.52 \pm 2.14\%$	$-9.32 \pm 3.11\%$
4	$-2.35 \pm 0.91\%$	$-12.00 \pm 1.54\%$	$-13.78 \pm 2.47\%$

Table 17: LXMERT (Scratch) **complex** systematicity gap (complex-OOD accuracy minus complex-IID accuracy). Average systematicity gap and standard deviation are on the differences, over all 3 runs (with different random seeds) of all HOPs with the stated diversity. Average accuracies are reported for each diversity (row) and each training set size (column).

Diversity	25k	200k	560k
24	$3.72 \pm 7.32\%$	$-1.93 \pm 2.96\%$	$-0.06 \pm 0.22\%$
16	$0.98 \pm 8.62\%$	$-1.69 \pm 3.84\%$	$-1.58 \pm 2.18\%$
6	$9.12 \pm 8.97\%$	$-6.78 \pm 7.26\%$	$-11.03 \pm 7.32\%$
4	$5.95 \pm 5.51\%$	$-7.73 \pm 4.70\%$	$-14.23 \pm 4.96\%$

Table 18: LXMERT (Scratch) **minimal** systematicity gap (minimal-OOD accuracy minus minimal-IID accuracy). Average systematicity gap and standard deviation are on the differences, over all 3 runs (with different random seeds) of all HOPs with the stated diversity. Average accuracies are reported for each diversity (row) and each training set size (column).

Diversity	25k	50k	100k	200k	300k	400k	560k
24	$1.00 \pm 0.34\%$	$0.60 \pm 1.01\%$	$0.06 \pm 0.26\%$	$-0.62 \pm 0.21\%$	$-2.44 \pm 0.35\%$	$-1.66 \pm 0.69\%$	$0.00 \pm 1.21\%$
16	$0.30 \pm 0.24\%$	$0.12 \pm 0.50\%$	$-0.21 \pm 0.33\%$	$-0.98 \pm 0.36\%$	$-3.41 \pm 0.42\%$	$-6.18 \pm 1.67\%$	$-2.35 \pm 1.87\%$
6	$0.56 \pm 0.64\%$	$0.05 \pm 0.54\%$	$-0.37 \pm 0.70\%$	$-1.40 \pm 0.46\%$	$-2.93 \pm 0.56\%$	$-7.05 \pm 1.14\%$	$-8.22 \pm 1.48\%$
4	$-1.03 \pm 0.40\%$	$-1.17 \pm 0.05\%$	$-2.23 \pm 0.27\%$	$-3.87 \pm 0.02\%$	$-4.63 \pm 0.39\%$	$-9.54 \pm 1.49\%$	$-15.27 \pm 0.75\%$

Table 19: Tensor-NMN **complex** systematicity gap (complex-OOD accuracy minus complex-IID accuracy). Average systematicity gap and standard deviation are on the differences, over all 3 runs (with different random seeds) of all HOPs with the stated diversity. Average accuracies are reported for each diversity (row) and each training set size (column).

Diversity	25k	50k	100k	200k	300k	400k	560k
24	$15.20 \pm 6.08\%$	$-9.70 \pm 14.59\%$	$19.12 \pm 1.31\%$	$-9.54 \pm 2.88\%$	$2.15 \pm 10.09\%$	$1.36 \pm 1.39\%$	$-0.52 \pm 0.74\%$
16	$-2.17 \pm 16.24\%$	$3.07 \pm 14.40\%$	$-4.84 \pm 18.92\%$	$-2.87 \pm 15.02\%$	$-19.61 \pm 9.83\%$	$-16.08 \pm 10.18\%$	$-9.03 \pm 10.07\%$
6	$-0.36 \pm 12.87\%$	$3.64 \pm 14.73\%$	$8.82 \pm 11.73\%$	$-5.91 \pm 8.65\%$	$-6.38 \pm 13.92\%$	$-11.40 \pm 10.38\%$	$-10.40 \pm 4.82\%$
4	$11.38 \pm 16.64\%$	$7.06 \pm 15.89\%$	$31.37 \pm 3.85\%$	$8.37 \pm 3.59\%$	$8.56 \pm 3.68\%$	$3.86 \pm 12.58\%$	$-22.75 \pm 1.36\%$

Table 20: Tensor-NMN **minimal** systematicity gap (minimal-OOD accuracy minus minimal-IID accuracy). Average systematicity gap and standard deviation are on the differences, over all 3 runs (with different random seeds) of all HOPs with the stated diversity. Average accuracies are reported for each diversity (row) and each training set size (column).

H CLEVR-HOPE Dataset Datasheet

Motivation for Dataset Creation

Why was the dataset created? (e.g., were there specific tasks in mind, or a specific gap that needed to be filled?)

The CLEVR-HOPE diagnostic dataset was created to study systematicity with respect to held-out pairs of attribute values in a controlled setting. These held-out pairs include various color-shape, color-material, color-size, size-shape, size-material, and shape-material pairs; each of the 29 pairs has a dedicated train set and four dedicated test sets. The specific task is visual question answering (VQA), in the form of 28-way classification.

To the best of the author's knowledge, this was a specific gap that needed to be filled. The closest prior work is the CLEVR-CoGenT dataset: Johnson et al. (2017a) created a train-test CLEVR split where at train time cubes and cylinders are restricted to limited color palettes, that are reversed at test time. Unlike CLEVR-HOPE, CLEVR-CoGenT does not change the question distribution at train time — held-out combinations can leak by appearing in text at train time. Furthermore, CLEVR-CoGenT has only a single train set with held-out COLOR-SHAPE combinations — whereas CLEVR-HOPE expands the set of held-out combinations to 29 train sets, covering all possible pairs of attribute types. CLEVR-HOPE also independently assesses each HOP, including in a minimal setting. In combination, these improvements allows the use of CLEVR-HOPE to study the impact of train-time diversity on systematicity.

What (other) tasks could the dataset be used for? Are there obvious tasks for which it should not be used?

CLEVR-HOPE can also be useful for studying model transfer from another domain (e.g., natural images) to the synthetic CLEVR domain. CLEVR-HOPE is a diagnostic dataset only, it is not intended as a thorough evaluation of a model's systematicity.

Has the dataset been used for any tasks already? If so, where are the results so others can compare (e.g., links to published papers)?

CLEVR-HOPE has only been used in this paper. A GitHub repo for recording works using this dataset will be provided. It is redacted at present to preserve anonymity.

Who funded the creation of the dataset? If there is an associated grant, provide the grant number.

Resources used in preparing this research were provided, in part, by the Department of Computer Science at the University of Toronto, the Province of Ontario, the Government of Canada through CIFAR, companies sponsoring the Vector Institute (`www.vectorinstitute.ai/partnerships/current-partners/`), the Hyundai Motor Company (under the project Uncertainty in Neural Sequence Modeling), the Samsung Advanced Institute of Technology (under the project Next Generation Deep Learning: From Pattern Recognition to AI), and by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

Ian Berlot-Attwell is funded by a Natural Sciences and Engineering Research Council of Canada Postgraduate Scholarship-Doctoral, and a Vector Institute Research Grant. A. Michael Carrell is funded in part by a Microsoft Research scholarship. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Yash Sharma.

Any other comments? N/A

Dataset Composition

What are the instances? (that is, examples; e.g., documents, images, people, countries) Are there multiple types of instances? (e.g., movies, users, ratings; people, interactions between them; nodes, edges)

Each instance is comparable to a CLEVR instance. i.e., each instance consists of an image (a rendered blender scene of colored blocks on a plain background in the style of the CLEVR dataset), an English question, and a 1-word answer (there are 28 possible answers, exactly the same as in the original CLEVR). Scene graphs and the question's corresponding functional program (specified with the CLEVR question primitives) are also provided.

For each of the 29 held-out pairs (HOPs) in CLEVR-HOPE, train instances are of comparable complexity to CLEVR and do not contain the HOP in the image, or the question.

Of the four test sets: The complex-IID test and complex-OOD test sets have images and questions of comparable complexity to CLEVR. The minimal-OOD test and minimal-IID test sets contain minimal examples; the images are of only a single object, and the questions ask whether there is an object in the scene matching a specific pair of attribute values – e.g., "Are there any rubber cylinders?". Of these four test sets, the IID sets are like the train set in that the images and questions do not contain the HOP. The OOD test sets contain the HOP in both the question, and in at least one object in the image.

For more details see Sections 2 and B. Example images and questions are visualized in Fig. 1.

Are relationships between instances made explicit in the data (e.g., social network links, user/movie ratings, etc.)?

The only relationships between instances are that some instances re-use images (see Appendix B for further details), and some instances use questions generated from the same base template. In both cases, these relationships are available in the data. Instances reusing images refer to the same image index, and each question records its question family, as in CLEVR.

How many instances of each type are there?

For each of the 29 held-out pairs (HOPs) in CLEVR-HOPE, the approximate size of the corresponding splits is outlined below:

- train set: 62k images, and 560k image-question pairs
- complex-IID test set: 13k images, 120k image-question pairs
- complex-OOD test set: 15k images, 15k image-question pairs
- minimal-IID test set: 2576-3200 images, 8640-11970 image-question pairs (depending on HOP)
- minimal-OOD test set: 448-3840 images, 448-3840 image-question pairs (depending on HOP)

In general, for every HOP, each image in the train, and complex-IID test has 9 matching questions. Each image in complex-OOD test has 1 corresponding question.

The number of questions per image for minimal-IID test and minimal-OOD test varies depending on the HOP – see Section B for details on the construction of the minimal-IID test and minimal-OOD test datasets.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images)? Features/attributes? Is there a label/target associated with instances? If the instances are related to people, are subpopulations identified (e.g., by age, gender, etc.) and what is their distribution?

For every instance, the image is a 320 × 480 pixels. Images are aggregated over all HOPs in three HDF5 files (corresponding to train, IID test sets, and OOD test sets, respectively), which can be easily converted back to individual images in the PNG format.

The scene graphs are represented as .json files, following the CLEVR specification.

Questions, programs, and answer labels are provided in HDF5 files. Functional programs are encoded as a sequence of integers, the vocabulary mapping these integers to their English equivalents is provided in a JSON file. Questions are similarly encoded. Questions have undergone minimal tokenization, and the raw English questions are available in a separate JSON file. The only tokenization performed is the treating of "," and ";" as separate tokens, the removal of "." and "?" characters, and separation by white space. Answers are encoded as a single integer; the mapping to English is again in the JSON vocab file.

Instances are not related to people.

Is everything included or does the data rely on external resources? (e.g., websites, tweets, datasets) If external resources, a) are there guarantees that they will exist, and remain constant, over time; b) is there an official archival version. Are there licenses, fees or rights associated with any of the data?

CLEVR-HOPE does not rely on external resources.

Are there recommended data splits or evaluation measures? (e.g., training, development, testing; accuracy/AUC)

The dataset comes with recommended train/test splits that ensure no images are shared between the train and test splits, and the held-out pair only occurs in given test sets. It is recommended that hyperparameter tuning be done on the original CLEVR dataset. The intended evaluation is to report accuracy.

What experiments were initially run on this dataset? Have a summary of those results and, if available, provide the link to a paper with more information here.

Initial experiments were the fitting of LXMERT (both finetuned, and from scratch) on each of the 29 held-out pairs. Tensor-NMN was also fit to the first 6 HOPs. Models were trained using the full training set (560k image-question pairs), as well as subsets of size 25k and 200k.

In all cases, models exhibited some degree of systematicity, but performance degraded on OOD test sets. Furthermore, studying the systematicity gap (the difference between OOD and IID test performance) it was clear that the systematicity gap narrrowed as the train-time diversity of the HOP (i.e., the number of pairs of the same attribute types but different values) increased. See Sections 4.1 and 4.2 for details.

Data Collection Process

How was the data collected? (e.g., hardware apparatus/sensor, manual human curation, software program, software interface/API; how were these constructs/measures/methods validated?)

Data was generated via computer program. The code was modified from the original CLEVR codebase, and tested via code review among the authors, and manual inspection of the output.

Who was involved in the data collection process? (e.g., students, crowdworkers) How were they compensated? (e.g., how much were crowdworkers paid?)

N/A: Only the authors were involved.

Over what time-frame was the data collected? Does the collection time-frame match the creation time-frame? How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part of speech tags; model-based guesses for age or language)? If the latter two, were they validated/verified and if so how? Does the dataset contain all possible instances? Or is it, for instance, a sample (not necessarily random) from a larger set of instances?

N/A: The data was generated by python program, and the images rendered with Blender 2.7.

If the dataset is a sample, then what is the population? What was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)? Is the sample representative of the larger set (e.g., geographic coverage)? If not, why not (e.g., to cover a more diverse range of instances)? How does this affect possible uses?

For each of the 29 HOPs:

For the train, and complex-IID test the full population of images is the space of all valid CLEVR images such that no object matches the HOP (e.g., if the HOP is rubber cylinder, then there must be no rubber cylinders in the scene). The complex-OOD test population of images is valid CLEVR images such that at least one object matches the HOP. The minimal-OOD test and minimal-IID test are similar to complex-IID test and complex-OOD test respectively, but always have exactly 1 object in the scene.

The key constraints that valid CLEVR images must meet are that at least 100 pixels of each object must be visible, and that there must be 3-10 objects in the scene.

The sampling of images was probabilistic, uniformly at random.

The space of questions is the space of all instantiations of the CLEVR templates that produce well-formed questions (the key constraint being that questions are unambiguously answerable from the scenegraph and the functional form of the question). The sampling method was probabilistic in all cases. Following CLEVR, question templates were sampled randomly, and instantiations found via depth first search with randomized ordering of possibilities. Following CLEVR, sampling probabilities shift over time to encourage distribution balance with respect to question templates.

Is there information missing from the dataset and why? (this does not include intentionally dropped instances; it might include, e.g., redacted text, withheld documents) Is this data missing because it was unavailable?

No.

Are there any known errors, sources of noise, or redundancies in the data?

No.

Data Preprocessing

What preprocessing/cleaning was done? (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values, etc.)

The English questions were tokenized. The only tokenization performed is the treating of "," and ";" as separate tokens, the removal of "." and "?" characters, and separation by white space. Capitalization was not changed.

Was the "raw" data saved in addition to the preprocessed/cleaned data? (e.g., to support unanticipated future uses)

Yes.

Is the preprocessing software available?

Yes, the same tokenization as (Johnson et al., 2017b) was used.

Does this dataset collection/processing procedure achieve the motivation for creating the dataset stated in the first section of this datasheet?

Yes, for each of the 29 held-out pairs, we have a train set that does not contain the HOP, and test sets of minimal and comparable complexity that do or do not contain the HOP. Thus we can asses the systematicity of a model, as well as how the systematicity is affected by the exact HOP, the amount of training data, and the complexity of test data.

Dataset Distribution

How is the dataset distributed? (e.g., website, API, etc.; does the data have a DOI; is it archived redundantly?)

Distribution details are on the paper's official repository: `https://github.com/ikb-a/systematicity-gap-in-vqa`. The data is not archived redundantly.

When will the dataset be released/first distributed? (Is there a canonical paper/reference for this dataset?)

CLEVR-HOPE will be released with the publication of this paper.

What license (if any) is it distributed under? Are there any copyrights on the data?

CLEVR-HOPE is shared under a Creative Commons CC BY 4.0 license.

Note that CLEVR-HOPE contains images from the original CLEVR dataset (Johnson et al., 2017a) which is also shared under a CC BY 4.0 license, and CLEVR-HOPE was created using a modified version of the CLEVR generation code which was shared under a BSD license.

Are there any fees or access/export restrictions?

No.

Dataset Maintenance

Who is supporting/hosting/maintaining the dataset?

Hosting TBD, the details will be on the paper's official repository `https://github.com/ikb-a/systematicity-gap-in-vqa`. The lead author is maintaining the dataset.

How does one contact the owner/curator/manager of the dataset (e.g. email address, or other contact info)?

Contact the lead author via email at ianberlot@cs.toronto.edu.

Will the dataset be updated? How often and by whom? How will updates/revisions be documented and communicated (e.g., mailing list, GitHub)? Is there an erratum?

There are no plans for the dataset to be updated. If needed, it will be updated by the lead author, and changes documented via GitHub.

If the dataset becomes obsolete how will this be communicated?

The GitHub page will be updated to reflect this.

Is there a repository to link to any/all papers/systems that use this dataset?

Works using this work can be linked to on this page in the repository: `https://github.com/ikb-a/systematicity-gap-in-vqa/blob/main/FOLLOWUP.md`.

If others want to extend/augment/build on this dataset, is there a mechanism for them to do so? If so, is there a process for tracking/assessing the quality of those contributions. What is the process for communicating/distributing these contributions to users?

There is no provided mechanism, but they are free to do so under the license, and enouraged to do so by the authors.

Any other comments?

Due to the size of the dataset, we are currently exploring hosting options.

Legal & Ethical Considerations

If the dataset relates to people (e.g., their attributes) or was generated by people, were they informed about the data collection? (e.g., datasets that collect writing, photos, interactions, transactions, etc.)

N/A

If it relates to other ethically protected subjects, have appropriate obligations been met? (e.g., medical data might include information collected from animals) If it relates to people, were there any ethical review applications/reviews/approvals? (e.g. Institutional Review Board applications)

N/A

If it relates to people, were they told what the dataset would be used for and did they consent? What community norms exist for data collected from human communications? If consent was obtained, how? Were the people provided with any mechanism to revoke their consent in the future or for certain uses?

N/A

If it relates to people, could this dataset expose people to harm or legal action? (e.g., financial social or otherwise) What was done to mitigate or reduce the potential for harm?

N/A

If it relates to people, does it unfairly advantage or disadvantage a particular social group? In what ways? How was this mitigated?

N/A

If it relates to people, were they provided with privacy guarantees? If so, what guarantees and how are these ensured?

N/A

Does the dataset comply with the EU General Data Protection Regulation (GDPR)? Does it comply with any other standards, such as the US Equal Employment Opportunity Act?

N/A

Does the dataset contain information that might be considered sensitive or confidential? (e.g., personally identifying information)

No.

Does the dataset contain information that might be considered inappropriate or offensive?

No.