Limited Linguistic Diversity in Embodied AI Datasets

Anonymous ACL submission

Abstract

Language is an important component of Vision-Language-Action (VLA) models, but the linguistic quality of training and test data remains underexplored. We analyze language in several VLA datasets and find that it is highly repetitive and structurally simple. These findings highlight the need for more diverse and linguistically rich data to support robust language understanding in embodied settings.

1 Introduction

With advances in large language models (LLMs) and multimodal learning, language is increasingly used as an input modality across research fields, enabling practical, real-world systems. In robotics, this trend is reflected in the growing focus on VLA models such as OpenVLA (Kim et al., 2025), RT-X (Collaboration et al., 2024), and π0.5 (Intelligence et al., 2025). Much of this progress has been driven by datasets like Open X-Embodiment (OXE) (Collaboration et al., 2024), which are significantly larger and more diverse than earlier robotics datasets. Together with advances in language modeling, these developments have enabled a shift toward generalist robotic systems that use language to specify tasks.

Despite this progress, language—while a core modality in VLA systems—is often overlooked in dataset documentation and model evaluations. Most datasets emphasize diversity across objects, scenes, and embodiments, while evaluations focus on task success without testing robustness to language variation. Although some works raise concerns about limited generalization (AgiBot-World-Contributors et al., 2025) and insufficient evaluation, including sensitivity to paraphrases (Wang et al., 2024), the linguistic characteristics of these datasets—and their impact—remain largely unexamined. As a result, it is often unclear what kind of language these models are trained on. This limits our ability to assess model robustness, safety, and real-world applicability.

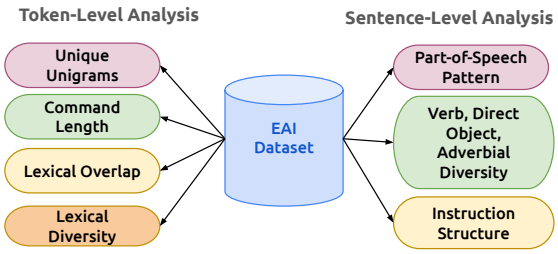


Figure 1: We perform linguistic diversity analysis on EAI datasets across two main categories: Token-Level for granular, lexical features and Sentence-Level for higher-level, syntactic patterns.

To address this, we analyze the language in several VLA datasets from OXE and compare them to other datasets from robotics and natural language understanding benchmarks. Using standard NLP tools and metrics, we evaluate linguistic diversity at both the token and sentence level. Our analysis uncovers systemic linguistic limitations in current VLA datasets that hinder model robustness and generalization. The datasets contain few unique commands and exhibit limited lexical diversity when compared to other robotics and natural language understanding datasets. The language used tends to follow repetitive syntactic patterns, with minimal variation in structure and vocabulary. Complex linguistic constructs such as negations, conditionals, and cycles are largely absent.

Although generalist language-guided robots are gaining traction, the language used to train VLA models remains limited in quality and diversity. Enhancing this language—either by collecting richer data or generating more varied synthetic inputs—could substantially improve natural language understanding in current VLA systems.

Dataset	Citations	Focus	Language Style
ALFRED (Shridhar et al., 2020)	738+	Household task instruction following	Step-by-step, high-level
SCOUT (Lukin et al., 2024)	0	Two-way, task-oriented dialogue	Unconstrained, interactive
Open X-Embodiment (Collaboration et al., 2024)	459+	Collection of datasets	Varied, not always included
RT-1 (Brohan et al., 2023)	1013+	Kitchen instruction following	Concise, imperative, templated
BRIDGE (Walke et al., 2023)	204+	Skill generalization across domains	Diverse, step-by-step
TacoPlay (Rosete-Beas et al., 2022)	81+	Task-agnostic "play" behaviors	Simple, low-variety, templated
Language Table (Lynch et al., 2023)	214+	Open-vocab spatial manipulation	Natural, open-ended
LIBERO (Liu et al., 2023)	114+	Knowledge transfer in robot learning	Natural[1]

Table 1: Overview of the datasets explored in this work. We include citation counts for each dataset; note that some of the referenced works focus primarily on dataset creation, while others introduce new methods alongside the dataset.

2 Datasets

In total, we examined seven robotics datasets (see Table 1) that cover a range of language types—from rigid, templated instructions to natural, open-ended, and interactive dialogue.

We include four well-known datasets from the OXE collection: RT-1 (Brohan et al., 2023), BRIDGE (Walke et al., 2023), TacoPlay (Rosete-Beas et al., 2022), and Language Table (Lynch et al., 2023). RT-1 and BRIDGE both target generalization across diverse tasks but differ in scope: RT-1 features imperative, templated commands, while BRIDGE offers richer linguistic and cultural variation, supporting tool use and nuanced object interactions. TacoPlay adopts a task-agnostic "play" paradigm, learning general-purpose behavior from unstructured interaction. In contrast, Language Table is designed for open-vocabulary spatial manipulation in controlled tabletop settings. We also include LIBERO (Liu et al., 2023), which is not part of the OXE collection but serves as a simulation benchmark focused on knowledge transfer. It has recently been used to fine-tune and evaluate models such as OpenVLA (Kim et al., 2024).

Additionally, we include two robotics datasets that are more focused on language interaction than directly training VLA models. ALFRED (Shridhar et al., 2020) emphasizes natural language through fine-grained, step-by-step action alignment, making it particularly suited for studying task decomposition. SCOUT (Lukin et al., 2024) contains the most naturalistic language among the datasets we consider. It captures unconstrained human-robot dialogues during navigation tasks, enabling more adaptive, context-aware interaction beyond static commands. Notably, it includes transcriptions from real robot commanders, and its accompanying publication provides detailed statistics on language use.

To contextualize the language complexity of modern robotics datasets, we include GLUE (Wang et al., 2018) and combine the training splits from each GLUE task into one GLUE dataset. Our goal is not to evaluate GLUE task performance but to use its examples as a reference for linguistic richness in comparison to robotics commands.

3 Results

This section presents a portion of our framework for analyzing language commands, focusing on token-level and syntax-level characteristics. Collectively, these analyses provide insight into the linguistic limitations of current EAI datasets. Methodological details can be found in the Appendices.

3.1 Token-Level Analysis

In this section, we provide token-level analysis to evaluate language through more interpretable lexical features, in contrast to the LLM-based representation analysis used in Section 3.2. For implementation details, see Appendix C.

Unique Commands and Unigrams serve as a simple metric to assess the diversity of each dataset and its vocabulary. This analysis (see Table 2) reveals a notable disparity: in most OXE datasets, fewer than 2% of language instructions contain unique wording. This is largely due to the same command being paired with multiple action trajectories via multiple trials. Compared to ALFRED and SCOUT, the other robotics datasets, except Bridge, contain relatively few unique unigrams. This is especially notable due to the difference in unique commands between SCOUT and LanguageTable. Compared to GLUE, all the other datasets have few unique unigrams, even considering the difference in unique commands, which can be explained by the task-focused nature of others.

Dataset	# Commands	% Unique Commands	# Unique Commands	# Unique Unigrams
ALFRED (Shridhar et al., 2020)	162K+	79.9%	126,005	2,627
SCOUT (Lukin et al., 2024)	23K+	39.4%	8,795	1,631
Open X-Embodiment (Collaboration et al., 2024)	-	-	-	-
RT-1 (Brohan et al., 2023)	3.7M+	0.02%	577	49
Bridge (Walke et al., 2023)	864K+	1.4%	11,693	1,189
TacoPlay (Rosete-Beas et al., 2022)	214K	0.2%	403	74
LanguageTable (Lynch et al., 2023)	7.0M+	1.81%	127,370	928
LIBERO (Liu et al., 2023)	6.5K	1.72%	112	79
GLUE (Wang et al., 2018)	1.0M+	73.1%	748,729	193,713

Table 2: Summary unique commands and unigrams of EAI datasets reviewed in this work.

Dataset	CR ↓	Levenshtein ↑	Jaccard ↓	BLEU-4 ↓	ROUGE-L ↓	Tree Kernel ↓	BERTScore ↓
ALFRED (Shridhar et al., 2020)	5.912	**46.695 ± 0.883**	0.128 ± 0.004	0.003 ± 0.000	0.214 ± 0.002	5.705 ± 0.140 %	0.638 ± 0.002
SCOUT (Lukin et al., 2024)	**4.851**	24.512 ± 0.946	**0.052 ± 0.002**	**0.002 ± 0.001**	**0.072 ± 0.004**	**1.892 ± 0.219** %	**0.493 ± 0.003**
RT-1 (Brohan et al., 2023)	118.195	28.143 ± 0.413	0.138 ± 0.001	0.026 ± 0.006	0.190 ± 0.007	5.090 ± 0.202 %	0.636 ± 0.005
BRIDGE (Walke et al., 2023)	64.904	35.139 ± 0.180	0.088 ± 0.004	0.003 ± 0.000	0.149 ± 0.002	3.680 ± 0.120 %	0.600 ± 0.002
TacoPlay (Rosete-Beas et al., 2022)	158.858	27.705 ± 0.137	0.188 ± 0.003	0.020 ± 0.001	0.304 ± 0.005	8.863 ± 0.132 %	0.683 ± 0.002
Language Table (Lynch et al., 2023)	56.643	32.206 ± 0.171	0.198 ± 0.002	0.010 ± 0.001	0.288 ± 0.004	-	0.697 ± 0.001
LIBERO (Liu et al., 2023)	134.862	34.269 ± 0.188	0.248 ± 0.006	0.064 ± 0.003	0.378 ± 0.003	12.222 ± 0.285	0.714 ± 0.001
GLUE (Wang et al., 2018)	**2.605**	**66.013 ± 1.480**	**0.039 ± 0.001**	**0.001 ± 0.001**	**0.069 ± 0.003**	**1.603 ± 0.029** %	**0.487 ± 0.001**

Table 3: Text similarity measures across robotics datasets. Each measure is computed by sampling 1,000 commands from each dataset, repeated three times for robustness. Arrows indicate increasing linguistic diversity. CR stands for Compression Ratio. The Tree Kernel method is from Moschitti (2006).

The **Command Length** distribution across seven datasets reveals a preference for short commands that fall within the range of 3 to 15 words (see Figure 4.) This highlights the dominance of concise phrasing, which may limit exposure to more complex linguistic structures, e.g., multi-clause, multi-step instructions.

Lexical Overlap. We analyze how much vocabulary is shared across datasets along the following POS categories: verbs, nouns, and adverbs. As shown in the heatmap in Figure 5, TacoPlay and RT-1, which have smaller vocabularies overall, share significantly fewer words with other datasets. Nouns are the most widely shared category, likely because many robotic tasks involve similar objects (e.g., boxes, cans, drawers). Verbs are also shared, though to a lesser extent, likely constrained by the specific capabilities of each robot embodiment.

Lexical Diversity Metrics. We present text similarity statistics in Table 3, which closely align with the unigram diversity patterns observed in Table 2. GLUE, SCOUT, and ALFRED consistently exhibit the highest levels of diversity, maintaining this ranking across all evaluated metrics. Notably, the low compression ratios for RT-1 and TacoPlay suggest that their language commands are highly structured and repetitive.

3.2 Intrinsic Dimensionality Analysis

We analyze the intrinsic dimensionality of language data by applying PCA to embeddings generated by standard LLM encoders. We approximate intrinsic dimensionality as the minimum number of principal components required to explain 95% of a dataset's cumulative variance (Fan et al., 2010; Verleysen and Lee, 2013); we justify our approach in Appendix A. We can infer a dataset's information density by determining how many principal components are necessary to reach this threshold. To mitigate model-specific biases, we evaluate embeddings from four distinct models: USE (512D) (Cer et al., 2018), SBERT (768D) (Reimers et al., 2019), CLIP (512D, multimodal) (Radford et al., 2021), and SONAR (1024D, multimodal) (Duquenne et al., 2023). Table 4 presents our results. We note that sample size does not trivially determine our results (see Figure 3) (Oates and Jensen, 1997).

3.3 Sentence-Level Analysis

In this section, we examine sentence-level structure, focusing on syntactic patterns, verb and direct object coverage, and uncover tendencies in instruction style. Refer to Appendix D for greater detail.

In particular, ALFRED and SCOUT are more

Dataset	# SBERT ↑	# USE ↑	# SONAR ↑	# CLIP ↑
ALFRED (Shridhar et al., 2020)	165	**159**	**406**	**198**
SCOUT (Lukin et al., 2024)	**194**	148	295	181
RT-1 (Brohan et al., 2023)	27	33	42	35
BRIDGE (Walke et al., 2023)	115	125	239	149
TacoPlay (Rosete-Beas et al., 2022)	31	42	41	36
Language Table (Lynch et al., 2023)	57	86	108	71
LIBERO (Liu et al., 2023)	32	34	44	33
GLUE (Wang et al., 2018)	**393**	**262**	**752**	**383**

Table 4: The Minimum Number of PCA Components to Explain 95% Variance for each EAI Dataset. A greater number of components represents stronger diversity.

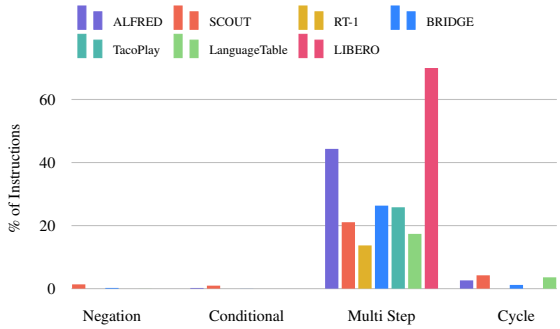


Figure 2: Percentage of instructions exhibiting four structural phenomena: negation, conditionality, multi-step sequencing, and cyclic repetition.

comparable to GLUE, while RT-1 and TacoPlay show much lower dimensionality, suggesting that their language is more limited in scope.

Part-of-Speech (POS) Pattern analysis examines the grammatical structure of commands, specifically how words are arranged using POS patterns. We use an LLM to extract these structures. As shown in the histograms in Figure 14, TacoPlay, SCOUT, RT-1, and LIBERO-10 exhibit long-tailed distributions, where just one or two syntactic templates dominate. This reliance on repetitive sentence structures may make it harder for models to generalize to more complex instructions. Refer to Figures 7a and 7b for qualitative examples of dominant patterns. Figure 13 offers an aggregated view across datasets.

Verb, Direct Object, Adverbial Diversity analysis explores how diverse the actions and modifiers are in language instructions. We measure how many unique verbs are associated with each object for manipulation datasets. Figures 16 and 15 show that most objects co-occur with fewer than ten verbs (fewer than five in LIBERO-10 and RT-1), indicating limited task diversity. However, ALFRED and Language Table exhibit more balanced and varied distributions. While some constraints stem from limitations in manipulation capabilities, others appear artificial; for example, TacoPlay's stacked blocks could support richer interactions (e.g., "observe" or "tip"). For navigation datasets like SCOUT, we examine the diversity of adverbials, which modify actions in ways that convey nuance in direction (north, forward), location (inside, around), manner (slowly, precisely), time (now, again), and conversational fillers (please, okay) (see Figure 11.)

Instruction Structure Analysis examines how instructions are logically composed, beyond just their vocabulary, by identifying four structural patterns: negation, conditionality, multi-step sequencing, and cyclical or loop-like patterns. Figure 2 visualizes their distribution, and Table 6 provides representative examples. See Appendix F for details.

We find that multi-step instructions are the most prevalent across all datasets, reflecting a strong bias toward procedural, linear task decomposition, particularly in LIBERO-10. Datasets like RT-1 and SCOUT contain fewer multi-step commands and favor shorter, atomic actions. Negation and conditional structures occur in less than 2% of cases. Their absence suggests that many benchmarks do not adequately capture logical disjunctions, exception handling, or constraint-driven behaviors essential for safe and flexible deployment. Cyclical or loop-like structures, common in real-world tasks, are similarly underrepresented, with only SCOUT and ALFRED showing a modest signal. This points to a structural bias in current datasets toward flat, step-by-step formulations, with limited support for more complex task logic.

4 Conclusion

In this work, we analyzed the linguistic properties of VLA datasets and showed that the language they contain is highly repetitive and structurally limited compared to language-focused robotics datasets and benchmarks like GLUE. The ALFRED and SCOUT datasets, with more focus on language, show significantly more diversity than those used for VLA training. These findings highlight that language remains an underemphasized modality in current VLA systems. Collecting more diverse language instructions or incorporating synthetic and augmented language data could substantially improve the limited language understanding of existing VLA models.

Limitations

Parts of our analysis rely heavily on automated annotations generated by LLMs. While we took steps to assess annotation quality for dependency parsing, occasional errors were observed and, due to dataset scale, could not be corrected exhaustively. A more rigorous study would include a structured quality assurance process and measure inter-annotator agreement even for manually reviewed generations, e.g., Section D.2. Additionally, while we analyzed seven datasets, which we believe capture dominant trends in the field, our findings may not fully generalize to all EAI instruction-following datasets.

References

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, and 33 others. 2025. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*.

Emily M. Bender. 2019. The benderrule: On naming the languages we study and why it matters. *The Gradient*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Anthony Brohan and 1 others. 2023. Rt-1: Robotics transformer for real-world control at scale. *Preprint*, arXiv:2212.06817.

Daniel Cer and 1 others. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Embodiment Collaboration and 1 others. 2024. Open x-embodiment: Robotic learning datasets and rt-x models. *Preprint*, arXiv:2310.08864.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Mathieu Dehouck and Carlos Gómez-Rodríguez. 2020. Data augmentation via subtree swapping for dependency parsing of low-resource languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3818–3830, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Paul-Ambroise Duquenne and 1 others. 2023. SONAR: sentence-level multimodal and language-agnostic representations. *arXiv preprint*.

Mingyu Fan, Nannan Gu, Hong Qiao, and Bo Zhang. 2010. Intrinsic dimension estimation of data by principal component analysis. *Preprint*, arXiv:1002.2050.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012.

Matthew Honnibal and 1 others. 2020. spacy: Industrial-strength natural language processing in python.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, and 17 others. 2025. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *Preprint*, arXiv:2504.16054.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. Openvla: An open-source vision-language-action model. *Preprint*, arXiv:2406.09246.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2025. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023. Libero:

Benchmarking knowledge transfer for lifelong robot learning. *Preprint*, arXiv:2306.03310.

Stephanie M. Lukin and 1 others. 2024. SCOUT: A situated and multi-modal human-robot dialogue corpus. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14445–14458, Torino, Italia. ELRA and ICCL.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. 2023. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, pages 1–8.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–120, Trento, Italy. Association for Computational Linguistics.

Tim Oates and David Jensen. 1997. The effects of training set size on decision tree complexity. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, volume R1 of *Proceedings of Machine Learning Research*, pages 379–390. PMLR. Reissued by PMLR on 30 March 2021.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Kishore Papineni and 1 others. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint*.

Nils Reimers and 1 others. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Erick Rosete-Beas and 1 others. 2022. Latent plans for task agnostic offline reinforcement learning.

Haoyue Shi, Karen Livescu, and Kevin Gimpel. 2021. Substructure substitution: Structured data augmentation for nlp. *Preprint*, arXiv:2101.00411.

Mohit Shridhar and 1 others. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Michel Verleysen and John A. Lee. 2013. Nonlinear dimensionality reduction for visualization. In *Neural Information Processing*, pages 617–622, Berlin, Heidelberg. Springer Berlin Heidelberg.

Homer Walke and 1 others. 2023. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. 2024. Ladev: A language-driven testing and evaluation platform for vision-language-action models in robotic manipulation. *Preprint*, arXiv:2410.05191.

Yubo Zhang and 1 others. 2020. Diagnosing the environment bias in vision-and-language navigation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 890–897. International Joint Conferences on Artificial Intelligence Organization. Main track.

A Intrinsic Dimensionality Analysis

A notable limitation of our methodology is using linear dimensionality reduction techniques, specifically PCA, to assess data that may lie on a nonlinear manifold, as is often the case with LLM-encoded datasets. While PCA assumes linearity, this limitation does not significantly undermine our analysis. In fact, it likely results in an *overestimation* of the intrinsic dimensionality, since PCA cannot exploit underlying nonlinear relationships in the data (Verleysen and Lee, 2013). For our purposes, this effect only further underscores the discrepancy between the structure of robotics datasets and the more diverse language representations found in natural language understanding (NLU) research.

Although the conclusions of this analysis are reinforced by our more interpretable feature-based methods (see Section 3.1); in future work, we would like to strengthen this effort.

B Qualitative Features of EAI datasets

We conducted an informal qualitative review of the examined datasets and highlighted interesting attributes, summarized in Table 5.

On Conversational Strengths. The SCOUT dataset exhibits a distinct dialogue structure that differentiates it from traditional instruction-following datasets. Rather than adhering to a rigid, directive style, its dialogues often involve an exploratory or inquiry-based approach, as seen in exchanges like "move west uh zero point five meters" and "...and then the last question here anything that indicates the environment was recently occupied". This interactive nature may offer advantages for EAI by allowing more adaptive responses. For example, in cases where instructions involve complex spatial reasoning (e.g., placing an object in a specific but ambiguous location), the dataset's conversational format could aid in disambiguation.

On Cultural Knowledge. One of the more striking aspects of the BRIDGE dataset is its incorporation of multicultural culinary terminology, despite being primarily monolingual (English). Unlike many Western-centric datasets, BRIDGE includes references to diverse cooking utensils and ingredients, such as purkoli (broccoli), brinjal (eggplant), brezzela (eggplant), capsicum (bell pepper), quince fruit, nigiri, wok, and kadai. This linguistic diversity suggests a broader representation of cultural knowledge, making incremental progress toward addressing concerns raised in prior work on dataset biases (Bender et al., 2021; Bender, 2019). Specifically, it challenges the tendency for data collection to reflect primarily Western, white, and wealthy audiences. Additionally, BRIDGE captures subtle social characteristics of human perception, such as humor, evidenced by an annotation that describes a mushroom toy as a "phallic looking item."

On "Common Sense" Reasoning. A recurring challenge across real-world datasets is the disconnect between world knowledge, common-sense reasoning, and practical instruction execution. While BRIDGE and ALFRED aim to ground tasks in realistic environments, many instructions contain fundamental inconsistencies or implausible directives. In ALFRED, for example, commands such as "open refrigerator, place potato to the right of tomato on second shelf of refrigerator, close refrigerator, open refrigerator, pick up potato from refrigerator, close refrigerator" expose rigid, mechanical assumptions about human behavior. Additionally, one must ask what has been accomplished by storing a potato in a refrigerator and then removing said potato in a matter of seconds. Another example from ALFRED includes, "Put an egg in a pan in the fridge." More concerning, and at times, unintentionally amusing, are instances of potentially unsafe or property-damaging instructions, such as "place a heated slice of tomato on a counter and **store a knife in a microwave**" or "**stab the tip of the knife into the wooden table**, in front of the gray plate closest to the lettuce." While a robot damaging a kitchen table may be preferable to microwaving a knife, these examples highlight inconsistencies in world knowledge modeling within these datasets. Similar anomalies appear in BRIDGE, where commands such as "take sushi out of the pan," "put sushi in pot...," and "put spatula in pan" suggest an oversimplified understanding of object affordances, human behavior, and broader world and cultural knowledge. If the broader EAI community sees embodiment as a necessary step toward elevating the representational learning of single-modality models, e.g., LLMs, we ought to discourage dataset collectors from building illogical "common-sense" associations.

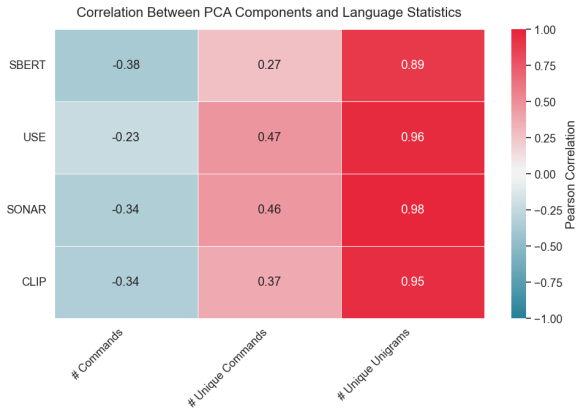


Figure 3: Correlation between the number of PCA components required to explain 95% variance and language statistics across EAI datasets. PCA components derived from SBERT, USE, SONAR, and CLIP embeddings are compared against the number of commands, unique commands, and unique unigrams in each dataset. Strong positive correlations are observed between unique unigrams and all embedding models, particularly SONAR and USE. In contrast, the total number of commands shows weak or negative correlation with embedding diversity

Theme	Example Instruction(s)
Cultural Terms (BRIDGE)	“put the kadai on the stove”, “grab the brinjal from the drawer”
Unsafe Action (ALFRED)	“store a knife in a microwave”, “stab the tip of the knife into the table”
Commonsense Violation (ALFRED)	“Put an egg in a pan in the fridge"
Commonsense Violation (BRIDGE)	“take sushi out of the pan”

Table 5: Selected examples illustrating conversational structure, cultural variation, and commonsense inconsistencies across EAI datasets.

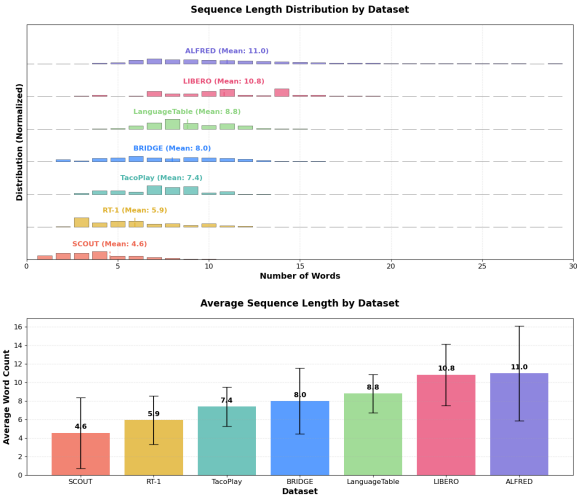


Figure 4: Distribution of command lengths across six examined EAI datasets. The majority of commands contain fewer than ten words. Command lengths are capped at a maximum of 30 words for analysis.

C Token-Level Analysis Methodology and Expanded Results.

C.1 Text Cleaning

All datasets were cleaned to standardize white space and remove punctuation. However, SCOUT (Lukin et al., 2024), a dialogue dataset, required further cleaning of user role tags and tags that indicate filler words, e.g., “um", silence, and noise. Due to the complexity of this data, we focus our initial analysis only on the “robot commander" dialogue, with plans to expand our analysis to all roles in the future and to incorporate filler filtering in the text cleaning pipeline. Once cleaned, we use a combination of `spacy` (Honnibal et al., 2020) and `pandas` (pandas development team, 2020) methods, e.g., `.unique()` to develop Tables 2 and Figure 4.

C.2 Lexical Overlap

To assess how much vocabulary is shared across datasets, we examine the distribution of words across three part-of-speech (POS) categories: nouns, verbs, and adverbs. We use dependency parsing (see Section 3.3) to extract tokens by their POS tags. We then construct a dataset–word matrix that records how often each word appears in more than one dataset. This allows us to visualize lexical overlap using a heatmap (Figure 5).

C.3 Token-Level Text Diversity Analysis

We use several text similarity measures in our analysis (see Table 3.) The first involves assessing syntactic diversity by comparing constituency parse trees (Moschitti, 2006). Following previous work (Zhang et al., 2020), we calculate BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) scores for candidate sentences against the remainder of their respective datasets. Additionally, we utilize Levenshtein distance as a metric as well as BERTScore. Given that these methods entail pairwise comparisons, we perform 1,000 commands to obtain these scores across 3 trials.

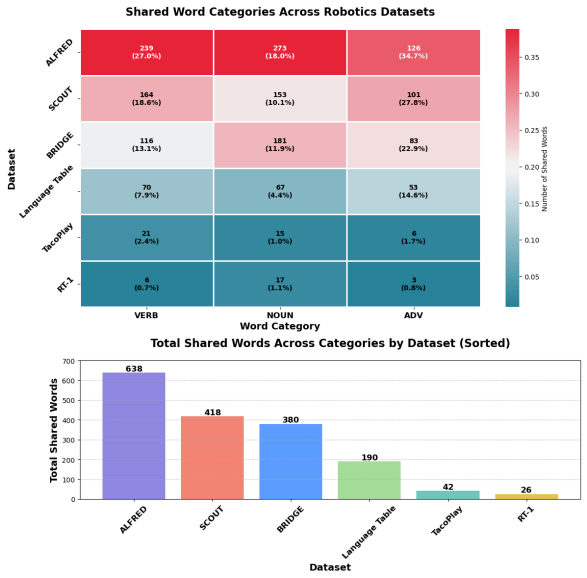


Figure 5: Shared POS categories across datasets. Using ALFRED as a pretraining dataset is advantageous because it has the greatest amount of lexical coverage across the examined EAI datasets.

```
# Function to Construct Prompts
def format_prompt(text):
    return f"""
    Perform dependency parsing on the following robotics command:

    Sentence: "{text}"

    Provide the output in a **valid JSON format** with the following structure:

    ```json
    {{
      "sentence": "PICK UP the red block",
      "tokens": [
        {{"text": "PICK", "lemma": "pick", "pos": "VERB", "head": 1, "dep": "ROOT"}},
        {{"text": "UP", "lemma": "up", "pos": "ADP", "head": 0, "dep": "prt"}},
        {{"text": "the", "lemma": "the", "pos": "DET", "head": 4, "dep": "det"}},
        {{"text": "red", "lemma": "red", "pos": "ADJ", "head": 4, "dep": "amod"}},
        {{"text": "block", "lemma": "block", "pos": "NOUN", "head": 1, "dep": "dobj"}}
      ]
    }}
    ```

    Token Fields Explanation:
    - `"text"`: The original word in the sentence.
    - `"lemma"`: The base (dictionary) form of the word.
    - `"pos"`: Part of Speech (e.g., VERB, NOUN, ADJ, etc.).
    - `"head"`: The index of the word that this token is dependent on.
    - `"dep"`: The dependency relation label (e.g., `ROOT`, `dobj`, `amod`, etc.).

    Ensure the output is in **valid JSON format** with proper nesting and data types.
    """
```

Figure 6: Prompt used in dependency parse work.

D Sentence-Level Analysis Methodology and Expanded Results

D.1 POS Patterns

We implemented a large-scale dependency parsing pipeline using an LLM to extract POS and dependency parse patterns, leveraging multi-GPU parallel processing for efficiency. Each GPU independently processed a subset of instructions using `DeepSeek-R1-Distill-Qwen-32B` (DeepSeek-AI et al., 2025), a state-of-the-art instruction-following LLM. The model was loaded in 8-bit quantized format to optimize memory usage, and batch $b = 10$ processing was employed to maximize throughput. The prompts for the model followed a structured format (see Figure 6), instructing it to perform dependency parsing and return results in valid JSON format. The output JSON included:

- The original instruction
- A tokenized breakdown, where each word was annotated with its:
  - Lemma (root form)
  - Part of speech (POS) tag
  - Syntactic head (parent word in the dependency tree)
  - Dependency label (e.g., ROOT, direct object, modifier, etc.)

For qualitative examples related to each POS pattern, please refer to Figure 7.

The **BRIDGE** dataset is heavily characterized by prepositional phrases, frequently structuring instructions that specify spatial relationships between objects and the environment. This results in a high frequency of ADP (adpositions), NOUN (nouns), and DET (determiners), forming patterns, e.g. “put the spoon on the cloth", “put the mangoes in a pan", and “Move the spatula near the egg." While this structure ensures precision in command execution, it lacks syntactic variation beyond simple prepositional constructs, potentially limiting generalization to more complex spatial reasoning tasks.

RT-1, in particular, exhibits highly repetitive syntactic patterns, as seen in commands like “place 7up can into middle drawer," “place water bottle into white bowl," and “place rxbar blueberry into bottom drawer." Similarly, TacoPlay demonstrates significant syntactic redundancy, with instructions such as “place the purple block on the table," “store the pink object in the drawer," and “slide the yellow block to the right." This lack of linguistic variability, likely due to the template-driven generation of these datasets, may limit a model’s ability to generalize to more complex instructions, particularly those involving hierarchical dependencies or compound actions.

SCOUT introduces more numerical expressions and adverbial structures, implying an instructional style where robots may be required to count, measure, or modify behaviors dynamically, e.g., “move south four feet", “turn right twenty degrees", “go forward one meter". However, its emphasis on concise command structures might underrepresent more complex multi-step directives.

The POS histograms in Figures 13 and 14 reveal a long-tailed distribution in TacoPlay, SCOUT, and RT-1, where the frequency of syntactic structures drops sharply after the first or second most common parse pattern. Such patterns indicate a reliance on repetitive syntactic templates, which may limit a model’s ability to generalize to linguistically varied instructions. Language Table shows the longest and most evenly distributed bar set among all datasets, with no single POS pattern dominating. Language Table sets the upper bound for linguistic diversity among embodied AI datasets and should be more widely used. However, for datasets like RT-1, we recommend that synthetic data augmentation could help mitigate this imbalance by introducing greater syntactic variability, such as tree-based reordering techniques, inspired by data augmentation in machine translation (Dehouck and Gómez-Rodríguez, 2020; Shi et al., 2021), could be adapted to generate syntactic variants of robotic commands while preserving their semantics.

Dataset	POS Pattern	Example Sentences
TacoPlay	VERB → DET → ADJ → NOUN → ADP → DET → NOUN	put the purple block on the table
		slide the purple block to the left
		place the yellow block on the table
	VERB → DET → ADJ → NOUN → ADP → DET → ADJ → NOUN	put the pink object inside the left cabinet
		put the yellow block inside the right cabinet
		place the purple block inside the right cabinet
	VERB → DET → ADJ → NOUN → CCONJ → VERB → PRON → ADV	take the purple block and rotate it right
		take the yellow block and turn it right
		grasp the purple block and turn it left
RT-1	VERB → NOUN → NOUN → ADP → ADJ → NOUN	place rxbar blueberry into bottom drawer
		move rxbar chocolate near orange can
		move 7up can near green can
	VERB → NOUN → NOUN → ADP → NOUN → NOUN	move water bottle near rxbar chocolate
		move coke can near water bottle
		move rxbar blueberry near water bottle
	VERB → NOUN → NOUN → ADP → ADJ → NOUN → CCONJ → VERB → ADP → NOUN	pick coke can from bottom drawer and place on counter
		pick water bottle from top drawer and place on counter
		pick rxbar blueberry from middle drawer and place on counter

(a) TacoPlay and RT1.

SCOUT	VERB → ADV → NUM → NOUN	turn left thirty degrees
		turn left ninety degrees
		move forward one foot
	VERB → ADP → DET → NOUN	move towards a shoe
		move towards the barrel
		go through the door
	VERB → NUM → NOUN → ADV	turn sixty degrees left
		move ten inches northeast
		move two feet forward
BRIDGE	VERB → DET → NOUN → ADP → DET → NOUN → PUNCT	Place the mushroom behind the spatula.
		Place the salmon in the pot.
		Move the mushroom onto the towel.
	VERB → DET → NOUN → ADP → DET → NOUN → ADP → DET → NOUN → PUNCT	Move the spatula at the edge of the table.
		Move the spoon to the left of the napkin.
		Put the cloth to the left of the spoon.
	VERB → DET → NOUN → ADP → DET → ADJ → NOUN → PUNCT	Place the strawberry in the silver pot.
		Set the pot onto the green cloth.
		Place the pot on the blue cloth.

(b) SCOUT and ALFRED.

Figure 7: Common POS Parse Patterns.

D.2 Verb, Direct Object, Adverbial Diversity.

To extract verb, direct object, and adverbial features, we implemented a large-scale annotation pipeline using two model variants: `R1-Distill-Llama-8B` and `R1-Distill-Qwen-14B` (DeepSeek-AI et al., 2025), just as in Section D.1. However, the prompts for the model followed the format shown in Figure 8. We implemented in-context learning (ICL) to enhance accuracy by retrieving sentence-specific examples using TF-IDF similarity. Despite using LLMs, all annotations were manually reviewed to ensure consistency, including lemmatizing verbs, removing duplicates, and normalizing synonymous expressions (e.g., "pick" vs. "pick up"). This hybrid method enabled the construction of high-quality annotations for downstream analysis. Results are provided in Figures 16, 15, and 11.

On Object and Adverbial Diversity. We assessed how many distinct verbs are used with each direct object for manipulation datasets. Low counts suggest limited interaction diversity, sometimes due to real-world constraints, but often due to overly templated instruction generation. Direct object structures are less relevant for navigation-focused datasets, instead how an instruction is followed, e.g., directional terms (e.g., "north," "forward"), location-based modifiers (e.g., "around," "inside"), manner descriptors (e.g., "slowly," "directly") are more relevant.

On Numeric Generalization. As VLA models are increasingly expected to interpret numerical quantities (e.g., distances, angles) in an end-to-end manner, the distribution of numerical values in navigation instructions becomes more critical. Figure 12a shows that numbers like "two," "three," and "five" are relatively common in SCOUT, while values such as "seven," "eight," or "twelve" are rare. ALFRED (see Figure 12b) appears more tailed and its numeric coverage is weaker than SCOUT; however, the overall representation of numerics is greater due to dataset size. This sparsity raises concerns about whether models trained on these datasets can interpolate or generalize to underrepresented numerical instructions. For example, can a robot correctly interpret "move seven meters" if it has never encountered that number in training? What if it has only encountered meters but is given a command in yards? What if the command contains common shortcuts, such as using 4K to refer to 4,000? Future research should investigate the impact of numeric and unit sparsity on navigation performance and explore methods for balancing numerical distributions during data collection or augmentation.

E Instruction Structure Analysis

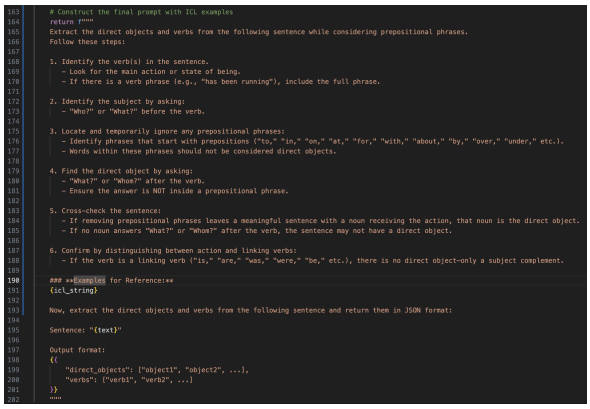
To analyze the compositional structure of language in robotics datasets, we use LLM-generated feature information (see Appendices D.1 and D.2) to construct heuristics for detecting four types of instruction-level patterns: negation, conditionality, multi-step sequencing, and cyclical structures. These patterns are identified through string-matching techniques and syntactic cues extracted from dependency parses and part-of-speech tags.

- **Negation** was detected using syntactic cues like neg dependencies and lexical markers (e.g., "not", "don't", "never").
- **Conditionality** was identified via subordinating conjunctions (e.g., "if", "unless") and dependency markers indicating conditional clauses.
- **Multi-step** sequencing was inferred from coordinating conjunctions (e.g., "and", "then"), punctuation, or imperative chaining.
- **Cyclical** patterns were identified using repeat verbs ("again", "repeat") or constructions indicating iteration or loops.

For each instruction, we annotated binary indicators for each structure type and aggregated them to compute relative frequencies across datasets. Quantitative results are presented in Figure 2, and representative examples are shown in Table 6. These results help reveal structural tendencies in instruction design; particularly, the dominance of linear, stepwise instruction formats and the underrepresentation of more complex, logic-driven patterns.

F Case Study: OpenVLA & LIBERO-10

This case study probes the linguistic robustness of OpenVLA using the LIBERO-10 dataset. Although LIBERO-10 is designated as an evaluation split in the LIBERO benchmark (Liu et al., 2023) (see Figure 10), the OpenVLA checkpoint used here (`openvla-7b-finetuned-libero-10`) was trained directly on this test set. As such, this



(a) Verb–direct object prompt example used in Section 3.3.

```
    # Format in-context learning examples
    icl_string = "\n".join(
        f"""Example {i+1}:
    Sentence: "{ex['example_sentence']}"
    Output:
    {{
        "direct_objects": {ex['direct_objects']},
        "verbs": {ex['verbs']}
    }}\n"""
        for i, ex in enumerate(icl_examples)
        if ex  # Ensuring valid examples are included
    )
```

(b) In context learning string generated by tf-idf distance k-nearest neighbors.

Figure 8: Prompts used in direct object and verb parsing tasks for instruction analysis.

experiment does not assess cross-split generalization (e.g., LIBERO-90 → LIBERO-10). But it may still yield two critical insights:

First, current evaluation practices in the robotics community often lead to confusion, as benchmark train/test splits are frequently assumed to be respected—even when they are not.

Second, and more surprisingly, despite leveraging a pretrained LLaMA 2 backbone (Touvron et al., 2023), the model—fine-tuned directly on the LIBERO-10 test split—fails when presented with simple paraphrased versions of the same instructions. This brittleness suggests that the limited linguistic diversity of the fine-tuning data alone can restrict generalization. In fact, we hypothesize that models finetuned on narrow, repetitive language may overwrite the model's generalist, linguistic capabilities encoded during pretraining. As shown in Figure 18, the average task success rate dropped from 0.66 on original instructions to 0.3168 on paraphrased variants.

Methodology. We begin by extracting linguistic features (verbs, direct objects, and syntactic patterns) from the LIBERO-10 test set (Liu et al., 2023), following the process in Section 3.3, but using GPT-3.5-turbo due to local GPU constraints. These features (see Figures 17a and 17b) inform targeted augmentations designed to probe the model's robustness, specifically by generating paraphrases that diverge from common verbs, objects, and syntactic templates. Paraphrases were generated using GPT-4o through a multifaceted process that included object substitutions (e.g., "cup" for "mug"), verb replacements (e.g., "activate" for "turn on"), and syntactic restructuring based on dependency parse patterns. Our exact prompt is provided in Figure 9. Variations included clause reordering, relative clauses, participial phrases, and passive constructions, with one strategy applied per prompt to ensure diversity while maintaining interpretability. Each prompt included the original BDDL file content to preserve semantic validity, exposing GPT-4o to the relevant object sets, affordances, and environment configurations. This context prevented implausible commands. Paraphrased instructions were then substituted into duplicated BDDL files to ensure the evaluation isolated linguistic robustness alone. For each task (original and paraphrased), we executed five trials per BDDL file, enabling a side-by-side performance comparison across language variants. Figure 17c demonstrates the efficacy of the paraphrasing pipeline.

```python
# ========== PROMPT BUILDING ==========
def build_prompt(row, variation_type, constraints):
    original = row["language_instruction"]
    verb = row["verb"]
    obj = row["direct_object"]
    dep = row["dependency_parse"]

    valid_targets = []
    for k, v in constraints.items():
        if obj in k or obj in v:
            valid_targets.extend(list(v))

    surface_hint = (
        f"The object '{obj}' can be placed or moved onto: {', '.join(valid_targets)}."
        if valid_targets
        else ""
    )

    if variation_type == "structure-complex":
        return STRUCTURE_COMPLEXITY_PROMPT.format(original=original, dep=dep)

    if variation_type == "verb":
        instruction = f"Replace the verb '{verb}' with a semantically equivalent verb while keeping the rest
of the instruction unchanged."
    elif variation_type == "object":
        instruction = f"Replace the direct object '{obj}' with a semantically equivalent noun phrase,
possibly with modifiers."
    elif variation_type == "structure":
        instruction = f"Rephrase the sentence by changing the structure (e.g., reordering clauses or using
different syntax like 'bowl that is black' instead of 'black bowl') while keeping the semantics identical."

    prompt = f"""
You are given a robot instruction:

Original command: "{original}"
Verb: {verb}
Object: {obj}
Dependency parse pattern: {dep}
{surface_hint}

Your task:
1. {instruction}
2. Return a **new instruction** that preserves meaning but makes the required variation.
3. Then, produce a **dependency parse** as a list of (word, relation, head) triplets.
4. Format your answer **exactly** as:

New instruction: <your new instruction>
Dependency parse:
[
  ("word1", "relation", "head"),
  ...
]
"""
    return prompt.strip()
```

```
STRUCTURE_COMPLEXITY_PROMPT = """
You are given a robot instruction and its dependency parse pattern.

Original command: "{original}"
Dependency parse pattern: {dep}

Your task:

1. Generate a paraphrase of the instruction by restructuring the syntax in a non-trivial way. You must alter
the syntactic structure, such as:

    - Reordering clauses
    - Introducing participial phrases (e.g., “after turning on the stove...”)
    - Using relative clauses (e.g., “the moka pot that goes on the stove...”)
    - Converting active to passive voice (e.g., “the stove should be turned on...”)
    - Fronting prepositional phrases (e.g., “on the stove, place the moka pot...”)
    - Embedding verbs or events within noun phrases or clauses

2. Ensure the new sentence preserves the original meaning, even if its structure is significantly different.

3. Pick one syntactic strategy at random — do not apply multiple at once unless natural.

Return:

New instruction: <your new instruction>
Dependency parse:
[
  ("word1", "relation", "head"),
  ...
]
""".strip()
```

Figure 9: Prompt used in paraphrase generation for test set. The parameter: `constraints` contains information from BDDL files which are then captured by `surface_hint`.

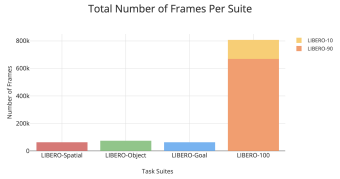


Figure 10: LIBERO task suite overview from https://libero-project.github.io/datasets.

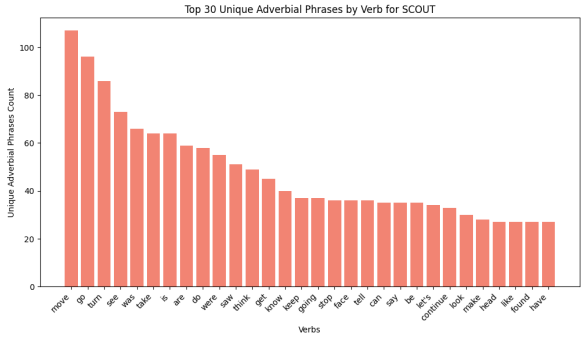
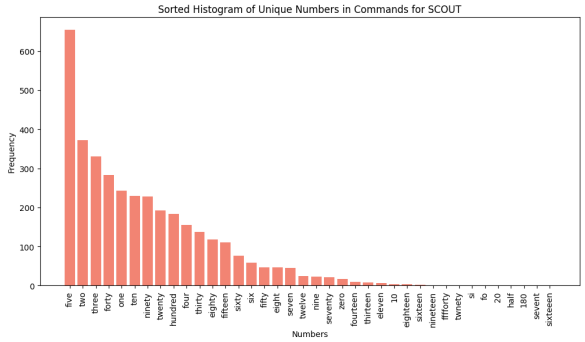
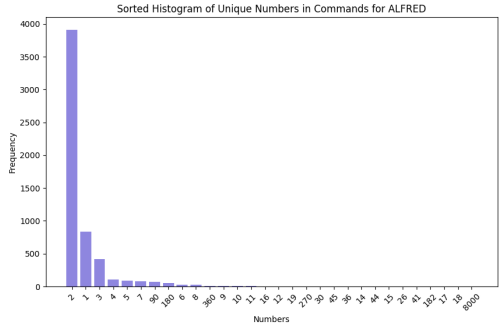


Figure 11: VLN adverbials - limited to the top 20 verbs with most unique language use



(a) SCOUT Numerics



(b) ALFRED Numerics

Figure 12: Numeric representation in navigation datasets.

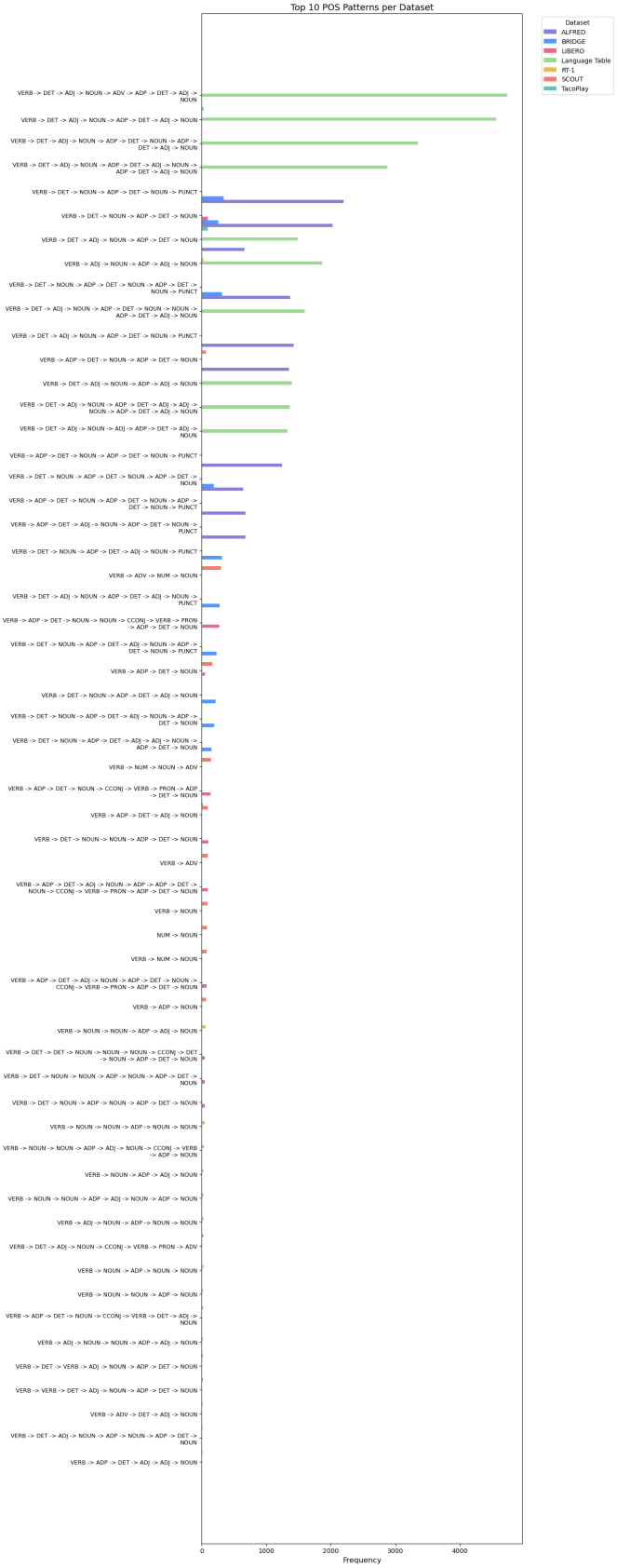


Figure 13: POS parse pattern distribution on unique commands in the datasets.

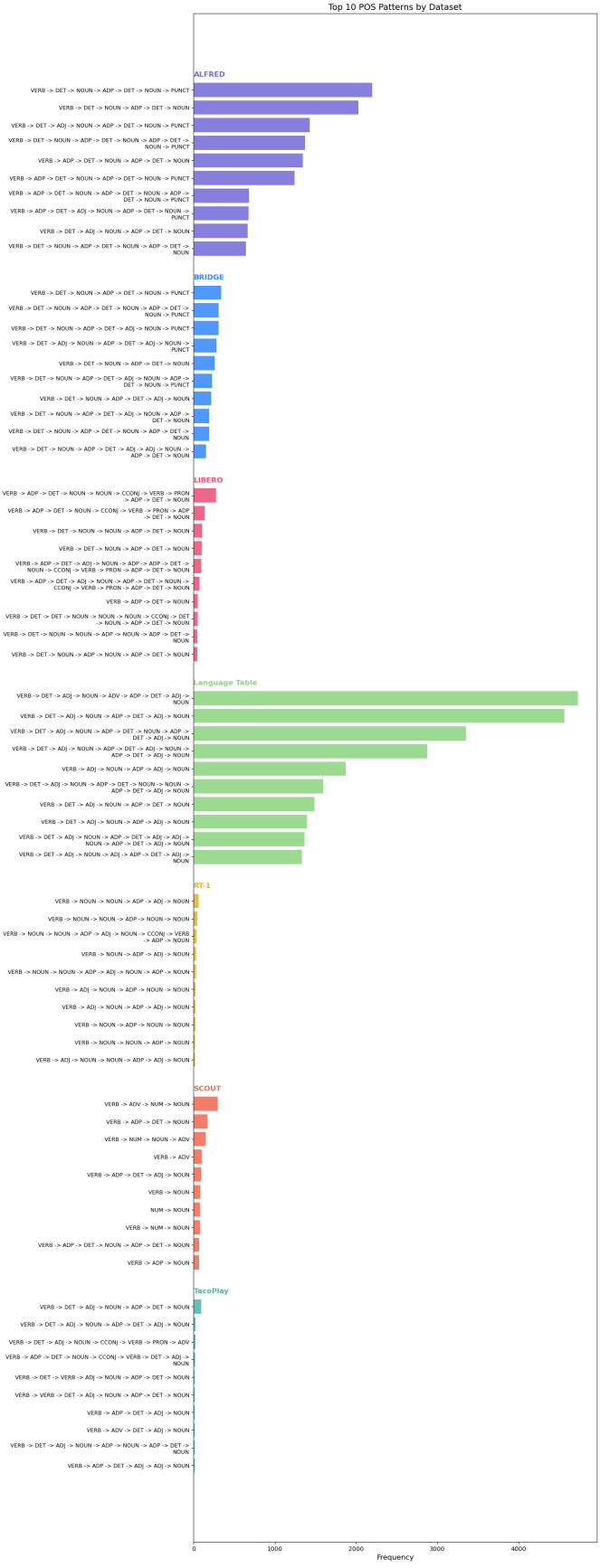


Figure 14: Grouped view of top 10 POS parse patterns on unique commands in EAI datasets.

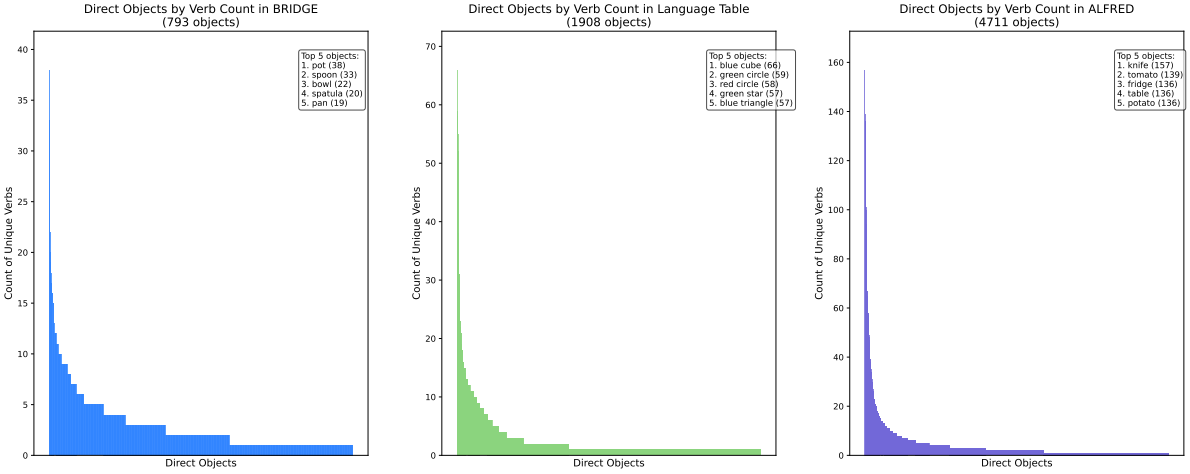


Figure 15: Frequency Plot of Unique Verbs per Direct Object for Manipulation Datasets

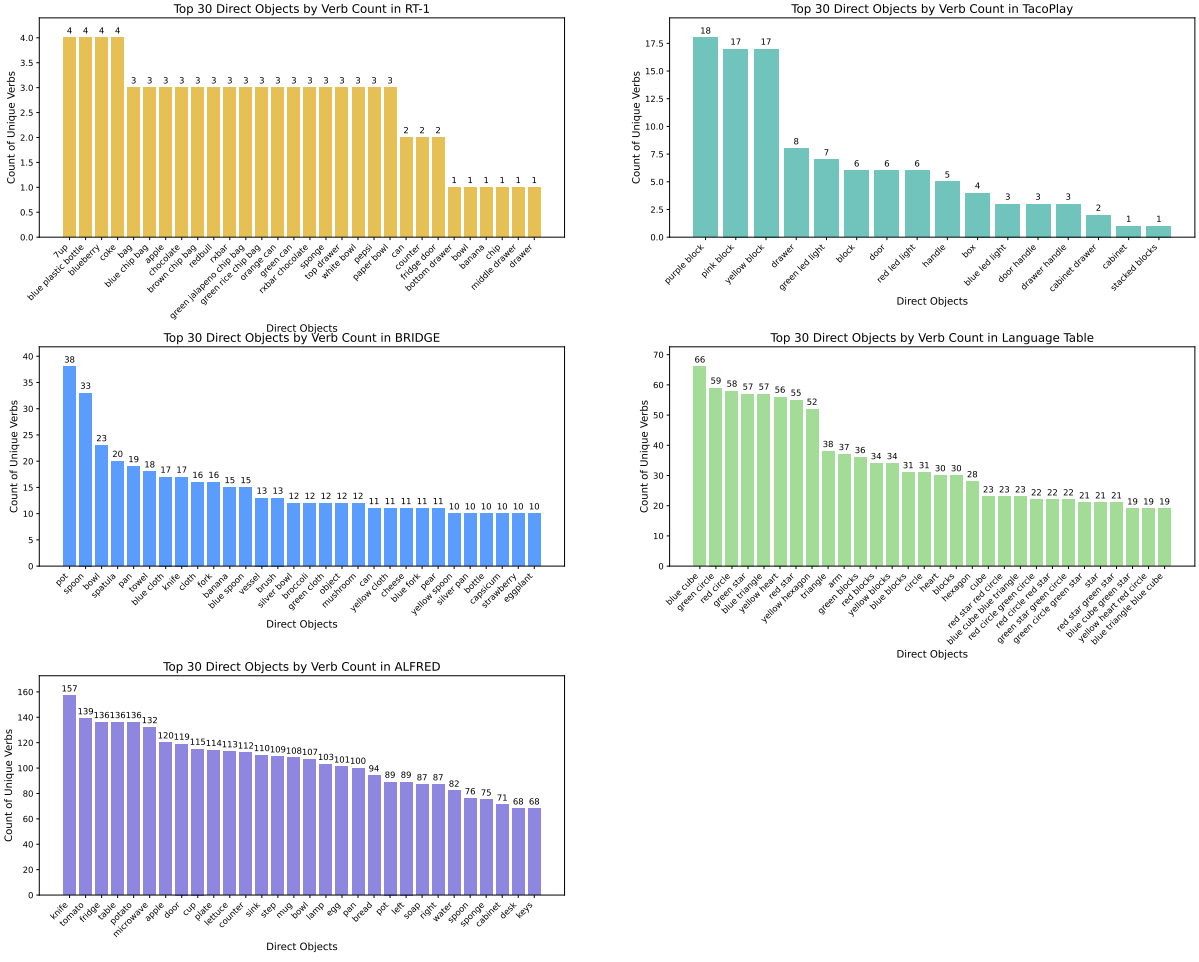
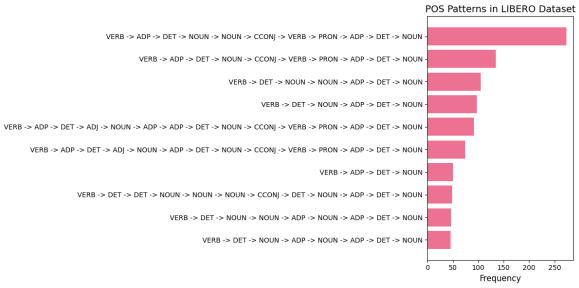


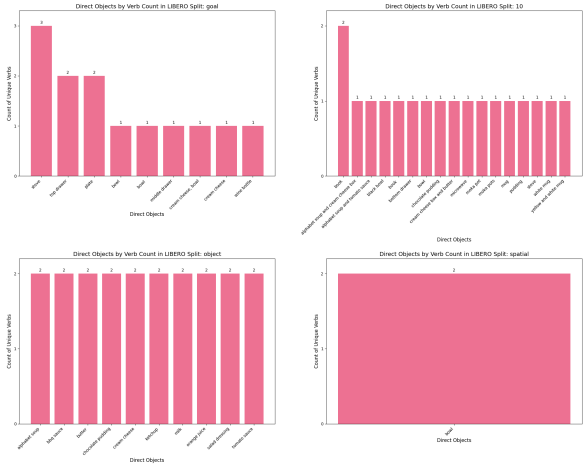
Figure 16: Frequency Plot of Unique Verbs per Direct Object for Manipulation Datasets

Category	Dataset	Examples
Negation	SCOUT	i don't know what the red thing was you are not at the total entrance no i did not see any
	BRIDGE	video frames not showing video frames or not showing Picture is not downloading, not able to view.
	ALFRED	This step does not exist. Slice the tomato on the counter but do not put down the knife. Cook the potato slice in the microwave and do not put the cooked potato slice on the counter.
Conditional	SCOUT	see if there's a doorway check and see if there's a doorway there and i'll point out when there's a doorway so we can count them
	BRIDGE	Pick the orange towel and place it on the middle if the table PLACE THE YELLOW TOPWEL SIDE IF THE TABLE
	ALFRED	Take keys from the black table, leave them on the lamp when you turn it on. Turn right and walk until you're even with the fridge on your right and when you are turn right and walk to it. Turn left and walk to the table then turn right when you get to it.
Multi-Step	LIBERO	open the top drawer and put the bowl inside
	TacoPlay	go towards the drawer and place the pink object go towards the purple block and grasp it take the purple block and rotate it right
	RT-1	pick coke can from bottom drawer and place on counter pick apple from top drawer and place on counter pick green rice chip bag from bottom drawer and place on counter
	SCOUT	and take a picture and then the last question here anything that indicates the environment was recently occupied and then take a picture
	BRIDGE	put pot or pan on stove and put egg in pot or pan Take the spatula from the vessel and place it on the table.
	ALFRED	Open the drawer. Put the cell phone in the drawer on the right side towards the back and close it. open the top right drawer of the desk, put phone inside, close the drawer Turn and move to the far end of the kitchen island, so you're facing the tomato and fork.
Cycle	SCOUT	continue moving forward follow hallway to the end of the wall uh to until you reach the wall take a photo every forty five degrees
	BRIDGE	end effector reaching knife pick orange toy from vessel and keep it on the left side of the table end effector reaching corn
	ALFRED	Move over to the right side of the desk again. Put the potato slice in the fridge and shut the door and then take the potato slice out and shut the fridge door again. Walk to your left until you see a loaf of bread on the counter top.

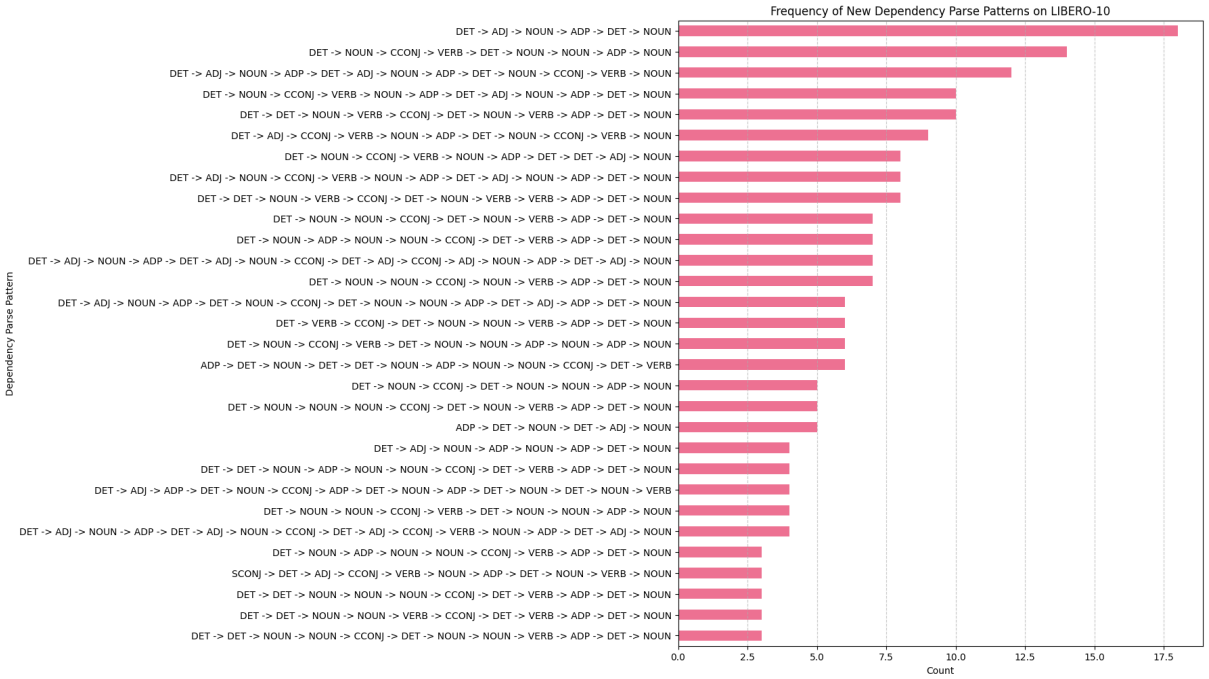
Table 6: Representative instruction examples for negation, conditional, multi-step, and cycle structures. Note that in BRIDGE and ALFRED, some examples contain noise from the original OXE metadata (e.g., typos or syntactic errors); and in many cases, this noise artificially inflate diversity scores.



(a) Dependency parse features across **all** LIBERO splits.



(b) Verb and direct object frequencies across **all** LIBERO splits.



(c) Distribution of POS patterns in the GPT-4o augmented LIBERO-10 test set.

Figure 17: Feature extraction across LIBERO datasets. Top: parse and verb–object statistics across all splits. Bottom: POS diversity from paraphrased instructions in LIBERO-10. These insights guide our augmentation pipeline (see Figure 1).

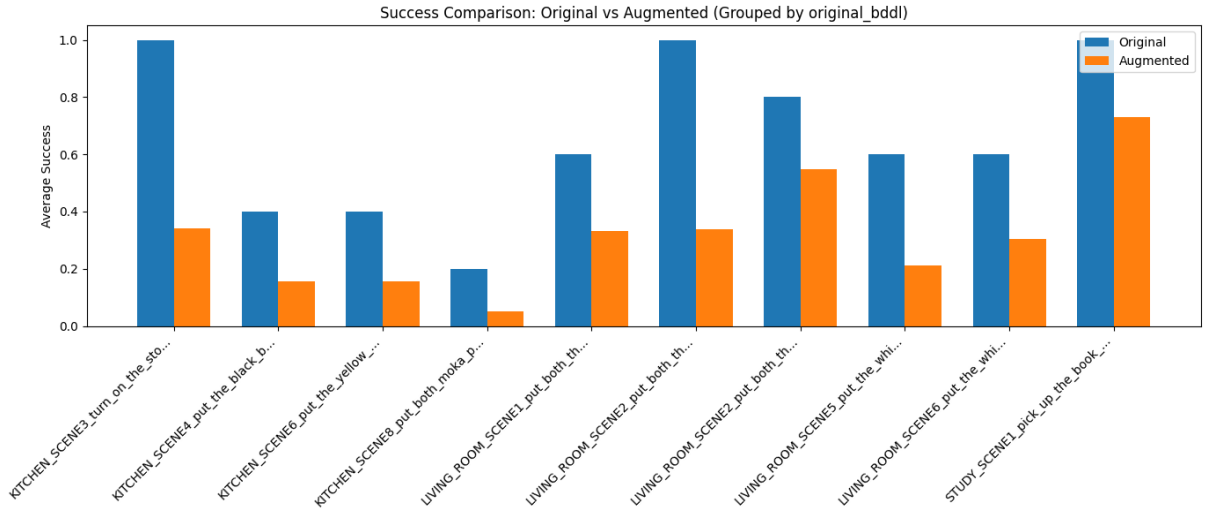


Figure 18: Average Task Success Rates Across Original and Augmented Instructions for LIBERO-10 Tasks. Each pair of bars represents the success rate of the OpenVLA model on a specific LIBERO-10 task using either the original task description (blue) or a GPT-4o-generated paraphrased version (orange). The drop in success on paraphrased instructions highlights the model's sensitivity to linguistic variation and limited robustness to novel language inputs. The OpenVLA checkpoint used was trained and tested on the LIBERO-10 split, so these results reflect a model (most likely) highly overfit to language data.