Let's Talk About Language! Investigating Linguistic Diversity in Embodied AI Datasets

Selma Wanna[1,4,*], Agnes Luhtaru[2,*], Ryan Barron[3,4], Jonathan Salfity[1],
Juston Moore[4], Cynthia Matuszek[3], and Mitch Pryor[1]

***Abstract*— The linguistic quality of Embodied AI (EAI) datasets is underexplored. We present a feature extraction pipeline that quantifies diversity across token- and sentence-level traits such as lexical variation and syntactic complexity. Applied to multiple EAI datasets, our analysis reveals a reliance on repetitive language that may hinder generalization. A feature-guided paraphrasing case study on LIBERO-10 shows that minor syntactic shifts can cut OpenVLA's success rate by over 50%, underscoring the value of fine-grained linguistic analysis for dataset design and model evaluation.**

I. INTRODUCTION

General-purpose models like large language models (LLMs) have gained widespread popularity across domains [1]–[4]. Following this trajectory, recent years have seen rapid progress in developing Vision-Language-Action (VLA) models and, more broadly, robotic foundation models, with works such as OpenVLA [5], RT-X [6], and others. These advancements have been largely driven by the emergence of datasets such as Open X-Embodiment (OXE) [6], which are significantly larger and more general-purpose than those in the past. This shift has enabled positive transfer between different robotics embodiments and platforms, some degree of generalization to unseen objects, among other promising capabilities.

Building on this progress, many recent works have also mentioned important limitations of OXE, such as limited diversity in scenes and objects [9], [10], poor generalization [9], difficulties with multiple objects [11], [12], reduced performance with novel objects [12], importance of optimizing the data mixture [13] and challenges in bimanual robot manipulation [14]. However, one aspect that remains underexplored is the role of language in these datasets.

We argue that language remains an overlooked, yet essential, component in the development and evaluation of VLA models. As a step towards studying the language in VLAs, we start by looking into the available datasets. By analyzing a subset of the OXE datasets alongside several others, we find that the language used in commonly adopted datasets lacks diversity across several dimensions, including the number of unique utterances, lexical variety (nouns and verbs), and morphological structure.

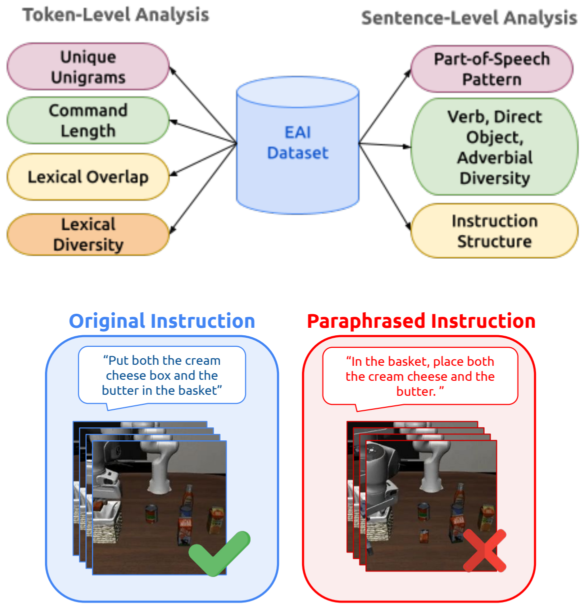


Fig. 1: (Top) We perform linguistic diversity analysis on EAI datasets across two main categories: Token-Level for granular, lexical features and Sentence-Level for higher-level, syntactic patterns. (Bottom) Using our insights from Part-of-Speech Pattern, we show that a small divergence from an original LIBERO-10 [7] instruction into a paraphrased instruction causes failures in OpenVLA [8].

We see this as a significant safety concern for the practical deployment of current systems and a promising opportunity for training better models. Motivated by this, we also show that simple paraphrasing of the test commands can significantly degrade model performance, revealing critical limitations in how current models represent and generalize language.

[*] Denotes equal contribution
[1] Nuclear and Applied Robotics Group, Department of Mechanical Engineering, The University of Texas at Austin, slwanna@utexas.edu
[2] Institute of Computer Science, University of Tartu, Estonia
[3] University of Maryland, Baltimore County. Maryland, USA
[4] Los Alamos National Laboratory. Los Alamos, USA

In summary, we show that:

- OXE datasets contain few unique commands and exhibit limited lexical diversity compared to other robotics and natural language understanding datasets;
- The language in EAI datasets follows repetitive syntactic patterns, includes few unique nouns and adverbs per verb, and rarely features complex structures such as negations, conditionals, or cycles;
- Paraphrasing language commands can lead to over 50% lower task success rates on OpenVLA pretrained on OXE and finetuned on LIBERO-90.

Based on these findings, we advocate for greater attention to language in VLA datasets — including reporting detailed language statistics, applying language augmentation for training data, and adopting more rigorous evaluation protocols to separate true language understanding from learned dataset-specific biases.

II. Datasets

The seven datasets we examine reflect a broad spectrum of Embodied AI research priorities, from low-level manipulation to high-level instruction following and dialogue (c.f. Table I.) For the purposes of our analysis, we consider all instructions (high-level goals and step-by-step directives) provided by the datasets. ALFRED emphasizes fine-grained, step-by-step action alignment with natural language in simulated indoor environments, which is ideal for studying grounded task decomposition. SCOUT uniquely captures two-way, unconstrained human-robot dialogues during navigation tasks, enabling more adaptive, context-aware interaction beyond static commands. RT-1 and BRIDGE both target generalization across diverse tasks, but differ in domain: RT-1 provides a real-world scale with short imperative commands, while BRIDGE includes richer linguistic and cultural variation, supporting tool use and nuanced object interactions. TacoPlay adopts a task-agnostic "play" paradigm to learn general-purpose behavior from unstructured interaction. Lastly, Language Table is designed for open-vocabulary spatial manipulation in controlled tabletop settings. These datasets span a continuum from rigid, templated instructions to open-ended, multimodal, and interactive language grounded in action. For additional insights, please refer to Table III, which highlights common pitfalls in the more sophisticated coverage datasets as well as a neat feature of cultural knowledge present in BRIDGE.

III. Results

This section presents a portion of our framework for analyzing language commands, focusing on token-level and syntax-level characteristics. We illustrate the relevance of these linguistic features through a case study that highlights challenges in language generalization on OpenVLA [8] and LIBERO-10 [7]. Collectively, these analyses provide insight into the linguistic limitations of current EAI datasets. Methodological details can be found in the Appendices.

A. Intrinsic Dimensionality Analysis

We perform an intrinsic dimensionality analysis of language data by encoding that data using standard LLM encoders and then performing a principal component analysis (PCA) across the entire embedded dataset. We approximate intrinsic dimensionality as the minimum number of principal components required to explain 95% of a dataset's cumulative variance [23], [24]; we justify our approach in Appendix I. 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) [25], SBERT (768D) [26], CLIP (512D, multimodal) [27], and SONAR (1024D, multimodal) [28]. Table II presents our results. We note that sample size does not trivially determine our results (see Figure 4) [29].

To contextualize the language complexity of modern robotics datasets, we include GLUE [30], a widely-used NLU benchmark suite. We 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. Despite being nearly 7 years old, GLUE exhibits higher intrinsic dimensionality than many robotics datasets. In particular, ALFRED and SCOUT are more comparable to GLUE, while RT-1 and TacoPlay show much lower dimensionality, suggesting that their language spaces are far more repetitive and limited in scope.

B. 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 III-A. For implementation details, see Appendix III.

Unique Unigrams refer to words that appear only once in a given dataset. This simple metric helps determine the diversity of each dataset's vocabulary. This analysis (see Table IV) reveals a disparity in vocabulary diversity: in most OXE datasets, fewer than 2% of language instructions contain unique wording. Even Language Table, which matches ALFRED's number of unique commands, lags. In contrast, ALFRED and SCOUT stand out with much richer vocabularies.

The **Command Length** distribution across six datasets reveals a preference for short commands that fall within the range of 3 to 15 words (see Figure

TABLE I: Overview of EAI datasets. The included datasets are covered by OXE [6] and prior research [15]. *The LIBERO-10 commands are taken from Ego4D [16] then used to develop language templates.

Dataset	**Citations**	**in OXE?**	**Focus**	**Interaction Type**	**Environment**	**Language Style**
ALFRED [17]	662	N	Household task instruction following	Egocentric nav. + manipulation	AI2-THOR simulator	Step-by-step, high-level
SCOUT [18]	1	N	Two-way, task-oriented dialogue	Collaborative navigation (dialogue turns)	Real-world (Wizard-of-Oz)	Unconstrained, interactive
RT-1 [19]	533	Y	Kitchen instruction following for scalable multi-task robot learning	Demonstration-based manipulation	Kitchen-themed setups	Concise, imperative, templated
BRIDGE [20]	124+	Y	Skill generalization across domains	Multi-task manipulation (incl. tools)	Kitchen-themed and tool shop setups	Diverse, step-by-step
TacoPlay [21]	54+	Y	Task-agnostic “play” behaviors	Unstructured, unlabeled interaction	Tabletop with toys/objects	Simple, low-variety, templated
Language Table [22]	217+	Y	Open-vocab spatial manipulation	Language-conditioned arrangement	Tabletop, fixed objects	Natural, open-ended
LIBERO [7]	102+	N	Knowledge transfer in lifelong robot learning	Demonstration-based manipulation	Procedurally generated kitchen and home environments	Natural*

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

Dataset	# SBERT ↑	# USE ↑	# SONAR ↑	# CLIP ↑
ALFRED [17]	165	**159**	**406**	**198**
SCOUT [18]	**194**	148	295	181
RT-1 [19]	27	33	-	35
BRIDGE [20]	115	125	239	149
TacoPlay [21]	31	42	41	36
Language Table [22]	57	86	-	71
GLUE [30]	**393**	**262**	-	**383**

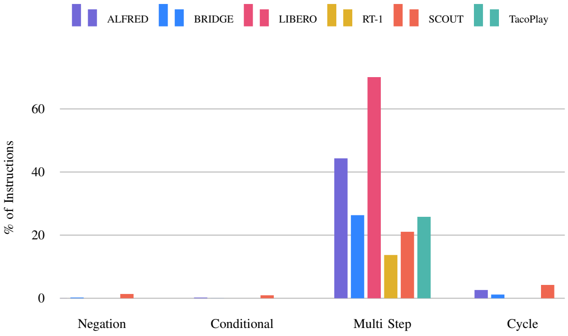


Fig. 2: Percentage of instructions exhibiting four structural phenomena: negation, conditionality, multi-step sequencing, and cyclic repetition. Multi-step constructions dominate across all datasets.

5.) 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 6, 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. Only four words appear in all datasets: move, close, open, and pick.

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

C. 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 IV for greater detail.

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 12, 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 8a and 8b for qualitative examples of dominant patterns. Our Case Study in Section III-D shows how OpenVLA struggles when deviating from these structures. To counteract this, we suggest future data collection or augmentation should focus on enriching the tail end of the syntactic distribution. Figure 11 offers an aggregated view across datasets to help guide that process.

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. As shown in Figures 17 and 14, most objects appear with fewer than ten distinct verbs (LIBERO-10 and RT-1 exhibit fewer than 5), revealing limited task diversity. 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,

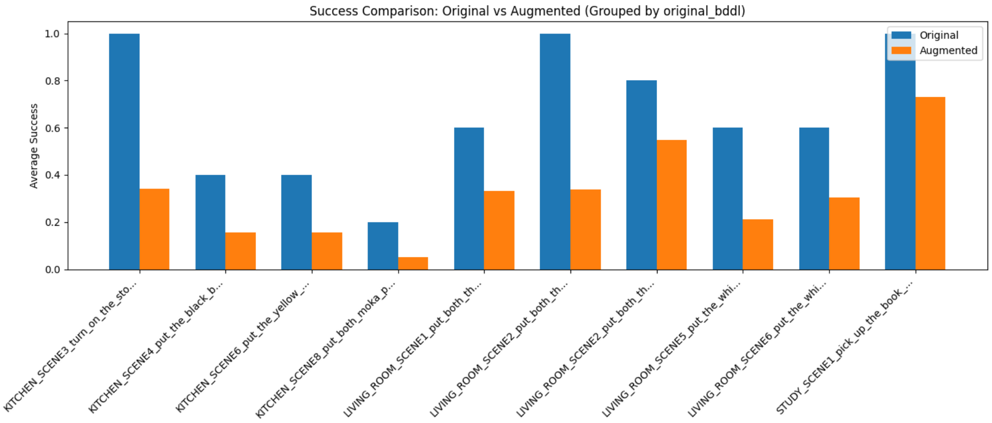


Fig. 3: Average Task Success Rates Across Original and Augmented Instructions for LIBERO-10 Tasks. Each pair of bars represents the success rate of the OpenVLA model on a specific LIBERO-10 task using either the original task description (blue) or a GPT-4o-generated paraphrased version (orange). The drop in success on paraphrased instructions highlights the model's sensitivity to linguistic variation and limited robustness to novel language inputs.

around), manner (slowly, precisely), time (now, again), and conversational fillers (please, okay) (see Figure 15.)

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 VI provides representative examples. See Appendix VI 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.

D. Case Study: OpenVLA & LIBERO-10

This case study examines the language generalization capabilities of OpenVLA (checkpoint: `openvla-7b-finetuned-libero-10`) by leveraging our prior feature analyses to create a challenging test set by targeting paraphrases that diverge from common verbs, objects, and POS patterns. See prompt and feature details in Appendix VI. The results, visualized in Figure 3, show a drop in performance on paraphrased instructions. The average success rate on original tasks was 0.66, compared to only 0.3168 for paraphrased variants. A paired t-test confirmed this was a statistically significant effect (t = -6.12, p = 0.0002), strongly suggesting that the performance degradation is not due to chance. These findings underscore a critical gap in current VLA benchmarks: models fine-tuned on a narrow band of linguistic expressions struggle to generalize to realistic, syntactically varied commands. This brittleness poses risks for real-world deployment and opens potential adversarial attack surfaces. It is imperative that future benchmarks and datasets more thoroughly account for linguistic diversity.

IV. CONCLUSION

Our data analysis, from the granular token level to the sentence level, presents linguistic attributes within existing EAI datasets for future EAI dataset developers. Our findings highlight critical limitations in the current VLA models' linguistic diversity and generalization capabilities. Even simple paraphrases can cause significant drops in performance, revealing an overreliance on surface-level language patterns. To support the development of more robust and trustworthy systems, we encourage the community to place greater emphasis on language in VLA research. In particular, we advocate for (1) reporting detailed statistics about the language data used in training, (2) incorporating synthetic paraphrases to improve generalization and robustness, and (3) investigating how language variation impacts both generalization and safety in real-world deployments. We hope this work motivates deeper integration of language-centered evaluation and augmentation in the future of embodied AI.

Appendix I
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 [24]. 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 III-B); in future work, we would like to strengthen this effort.

Appendix II
Qualitative Features of EAI datasets

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

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 [31], [32]. 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.

Appendix III
Token-Level Analysis Methodology and Expanded Results.

A. Text Cleaning

All datasets were cleaned to standardize white space and remove punctuation. However, SCOUT [18], 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` [33] and `pandas` [34] methods, e.g., `.unique()` to develop Tables IV and Figure 5.

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

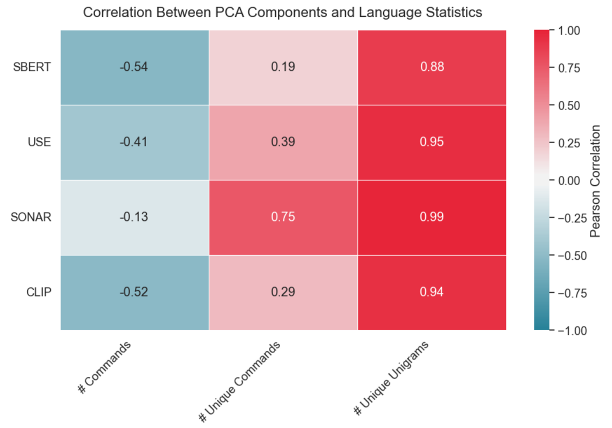


Fig. 4: Correlation between the number of PCA components required to explain 95% variance and language statistics across EAI datasets. PCA components derived from SBERT, USE, SONAR, and CLIP embeddings are compared against the number of commands, unique commands, and unique unigrams in each dataset. 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

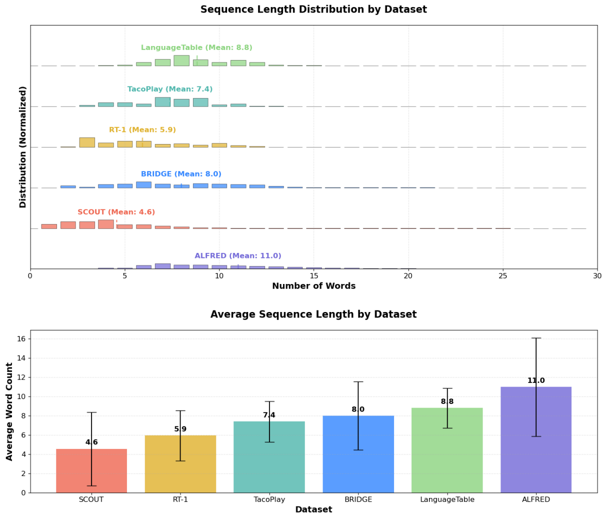


Fig. 5: Distribution of command lengths across six examined EAI datasets. The majority of commands contain fewer than ten words. Command lengths are capped at a maximum of 30 words for analysis.

B. 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 III-C) 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 6).

C. Token-Level Text Diversity Analysis

We use several text similarity measures in our analysis (see Table V.) The first involves assessing syntactic diversity by comparing constituency parse trees [35]. Following previous work [36], we calculate BLEU-4 [37] and ROUGE-L [38] 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 pair-wise comparisons, we perform 1,000 commands to obtain these scores across 3 trials.

Appendix IV
Sentence-Level Analysis Methodology and Expanded Results

A. 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` [40], 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 7), instructing it to perform dependency parsing and return results in valid JSON format. The output JSON included:

- The original instruction

TABLE IV: Summary unique commands and unigrams of EAI datasets reviewed in this work.

Dataset	# Commands	% Unique Commands	# Unique Commands	# Unique Unigrams
ALFRED [17]	162K+	79.9%	126,005	2,627
SCOUT [18]	23K+	39.4%	8,795	1,631
Open X-Embodiment [39]	-	-	-	-
RT-1 [19]	3.7M+	0.02%	577	49
Bridge [20]	864K+	1.4%	11,693	1,189
TacoPlay [21]	214K	0.2%	403	74
LanguageTable [22]	7.0M+	1.81%	127K+	928

TABLE V: Text similarity measures on robotics datasets. These measures were taken by sampling 1000 commands from each dataset, performed three times. Arrows point toward increasing diversity. Compression Ratio is CR.

Dataset	CR ↓	Levenshtein ↑	Jaccard ↓	BLEU-4 ↓	ROUGE-L ↓	Tree Kernel [35] ↓	BERTScore ↓
ALFRED [17]	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 [18]	**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 [19]	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 [20]	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 [21]	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 [22]	56.643	32.206 ± 0.171	0.198 ± 0.002	0.010 ± 0.001	0.288 ± 0.004	-	0.697 ± 0.001
GLUE [30]	**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**

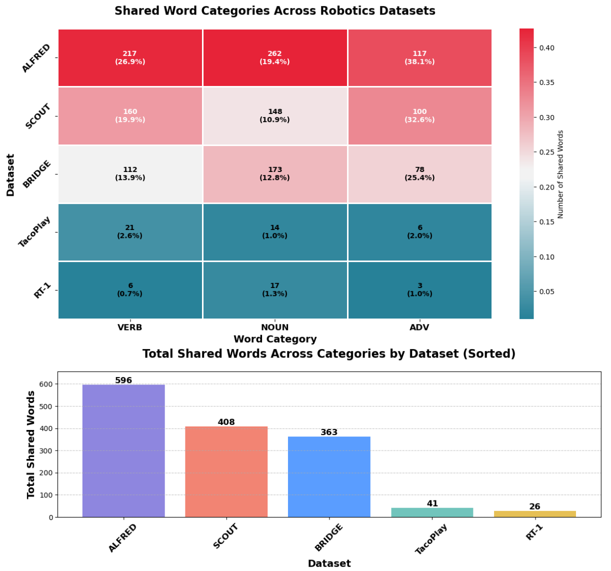
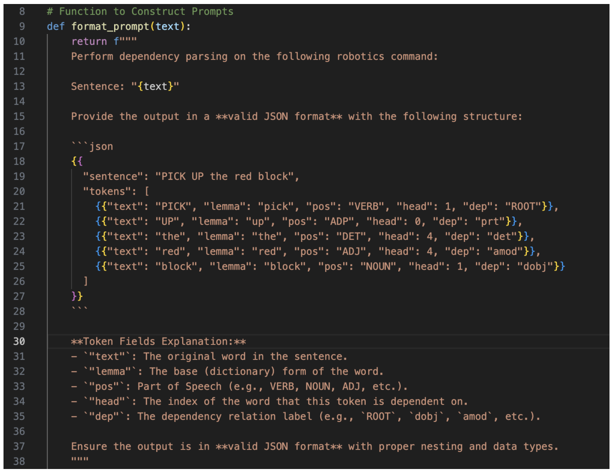


Fig. 6: Shared POS categories across datasets. Using ALFRED as a pretraining dataset is advantageous because it has the greatest amount of lexical coverage across the examined EAI datasets.

- 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)



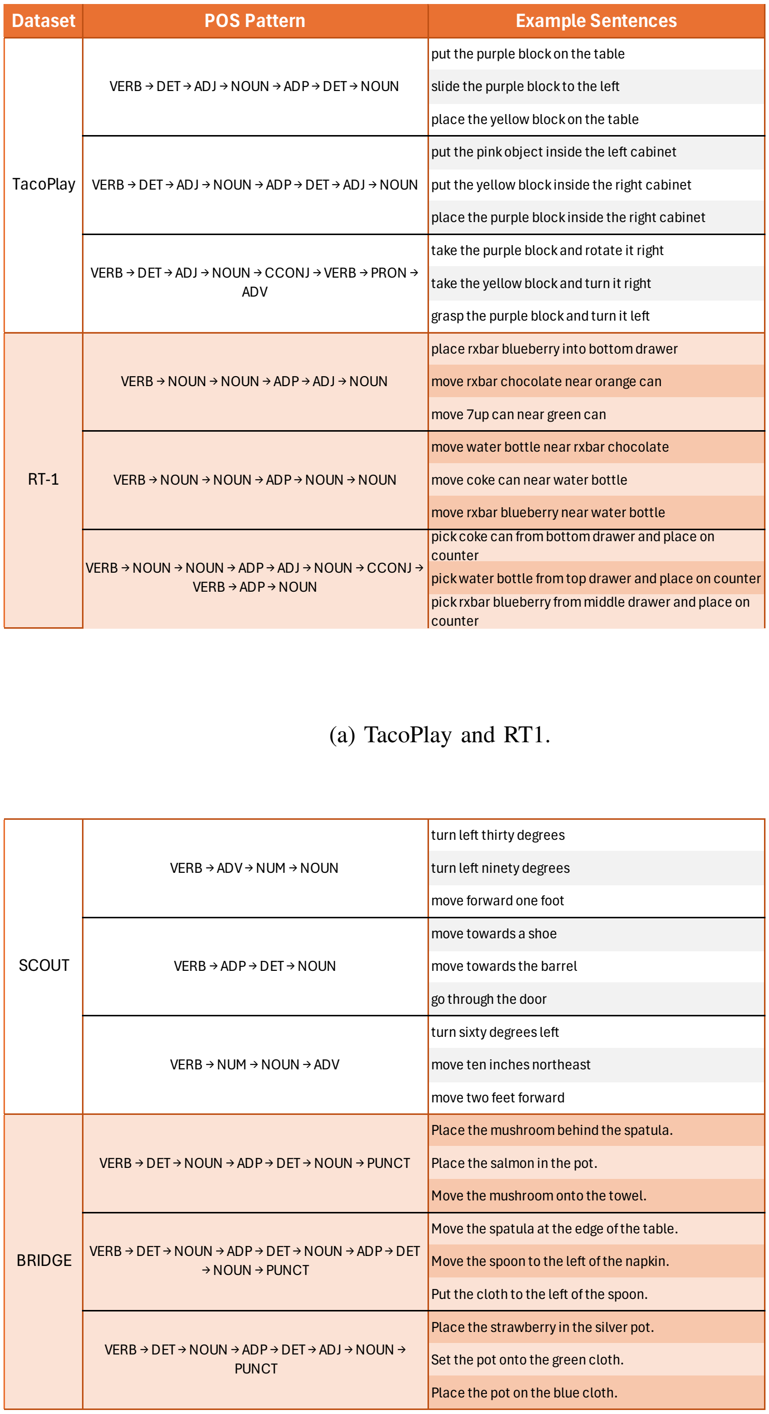
~~~
# Function to Construct Prompts
def format_prompt(text):
    return f"""
    Perform dependency parsing on the following robotics command:

    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.
    """
~~~

Fig. 7: Prompt used in dependency parse work.

  - Dependency label (e.g., ROOT, direct object, modifier, etc.)

For qualitative examples related to each POS pattern, please refer to Figure 8.

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



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.

Fig. 8: Common POS Parse Patterns.

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 11 and 12 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. 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 [41], [42], could be adapted to generate syntactic variants of robotic commands while preserving their semantics.

B. 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` [40], just as in Section IV-A. However, the prompts for the model followed the format shown in Figure 9. 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 17, 14, and 15.

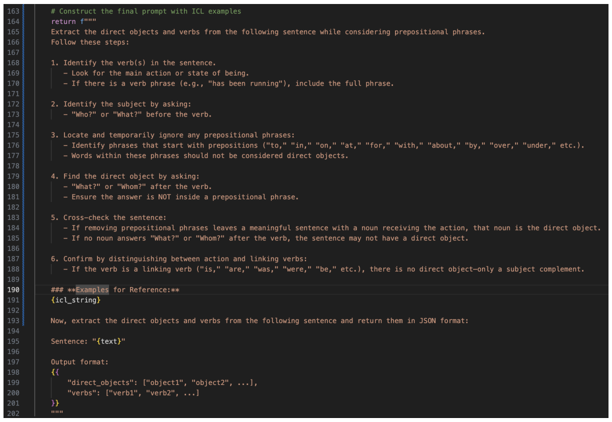
On Object and Adverbial Diversity. We assessed how many distinct verbs are used with each direct object for manipulation datasets. Low counts suggest limited interaction diversity, sometimes due to real-world constraints, but often due to overly templated instruction generation. Direct object structures 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 16 shows that numbers like “two,” “three,” and “five” are relatively common in SCOUT, while values such as “seven,” “eight,” or “twelve” are rare. 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.

Appendix V
Instruction Structure Analysis

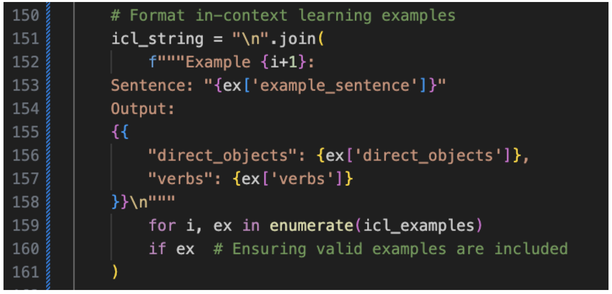
To analyze the compositional structure of language in robotics datasets, we use LLM-generated feature information (see Appendices IV-A and IV-B) to construct heuristics for detecting four types of instruction-level patterns: negation, conditionality, multi-step sequencing, and cyclical structures. These 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.



```
# Construct the final prompt with ICL examples
return f"""
Extract the direct objects and verbs from the following sentence while considering prepositional phrases.
Follow these steps:

1. Identify the verb(s) in the sentence.
   - Look for the main action or state of being.
   - If there is a verb phrase (e.g., "has been running"), include the full phrase.

2. Identify the subject by asking:
   - "Who?" or "What?" before the verb.

3. Locate and temporarily ignore any prepositional phrases:
   - Identify phrases that start with prepositions ("to," "in," "on," "at," "for," "with," "about," "by," "over," "under," etc.).
   - Words within these phrases should not be considered direct objects.

4. Find the direct object by asking:
   - "What?" or "Whom?" after the verb.
   - Ensure the answer is NOT inside a prepositional phrase.

5. Cross-check the sentence:
   - If removing prepositional phrases leaves a meaningful sentence with a noun receiving the action, that noun is the direct object.
   - If no noun answers "What?" or "Whom?" after the verb, the sentence may not have a direct object.

6. Confirm by distinguishing between action and linking verbs:
   - If the verb is a linking verb ("is," "are," "was," "were," "be," etc.), there is no direct object—only a subject complement.

### **Examples for Reference:**
{icl_string}

Now, extract the direct objects and verbs from the following sentence and return them in JSON format:

Sentence: "{text}"

Output format:
{{
    "direct_objects": ["object1", "object2", ...],
    "verbs": ["verb1", "verb2", ...]
}}
"""
```

(a) Verb–direct object prompt example used in Section III-C.



```
# 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.

Fig. 9: Prompts used in direct object and verb parsing tasks for instruction analysis.

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 VI. 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.

Appendix VI
Case Study: OpenVLA & LIBERO-10

Why LIBERO-10? LIBERO-10 is the designated evaluation suite for downstream generalization in the LIBERO benchmark. As shown in Figure 10, LIBERO-100 dominates the dataset regarding frame count, with LIBERO-90 comprising the bulk of the training data and LIBERO-10 representing only a small fraction reserved for evaluation. While LIBERO-Spatial, -Object, and -Goal are designed to isolate specific types of knowledge transfer, LIBERO-10 requires generalization over entangled knowledge domains, making it a natural testbed for stress-testing language-conditioned policies. Given its small data footprint yet high importance for lifelong learning assessment, augmenting LIBERO-10 with linguistically diverse instructions enables a more rigorous evaluation of language generalization capabilities.

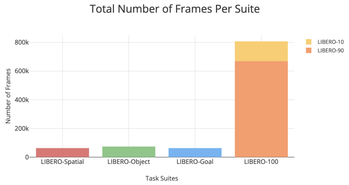


Fig. 10: LIBERO task suite overview from `https://libero-project.github.io/datasets`.

This case study examines the language generalization capabilities of OpenVLA (checkpoint: `openvla-7b-finetuned-libero-10`). We begin by extracting linguistic features (verbs, direct objects, and syntactic patterns) from the LIBERO-10 test set [7], following the process in Section III-C, but using GPT-3.5-turbo due to local GPU constraints.

These features (see Figures 18a and 18b) 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 13. 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 18c demonstrates the efficacy of the paraphrasing pipeline. The final success rate results, visualized in Figure 3, show a staggering drop in performance on paraphrased instructions.

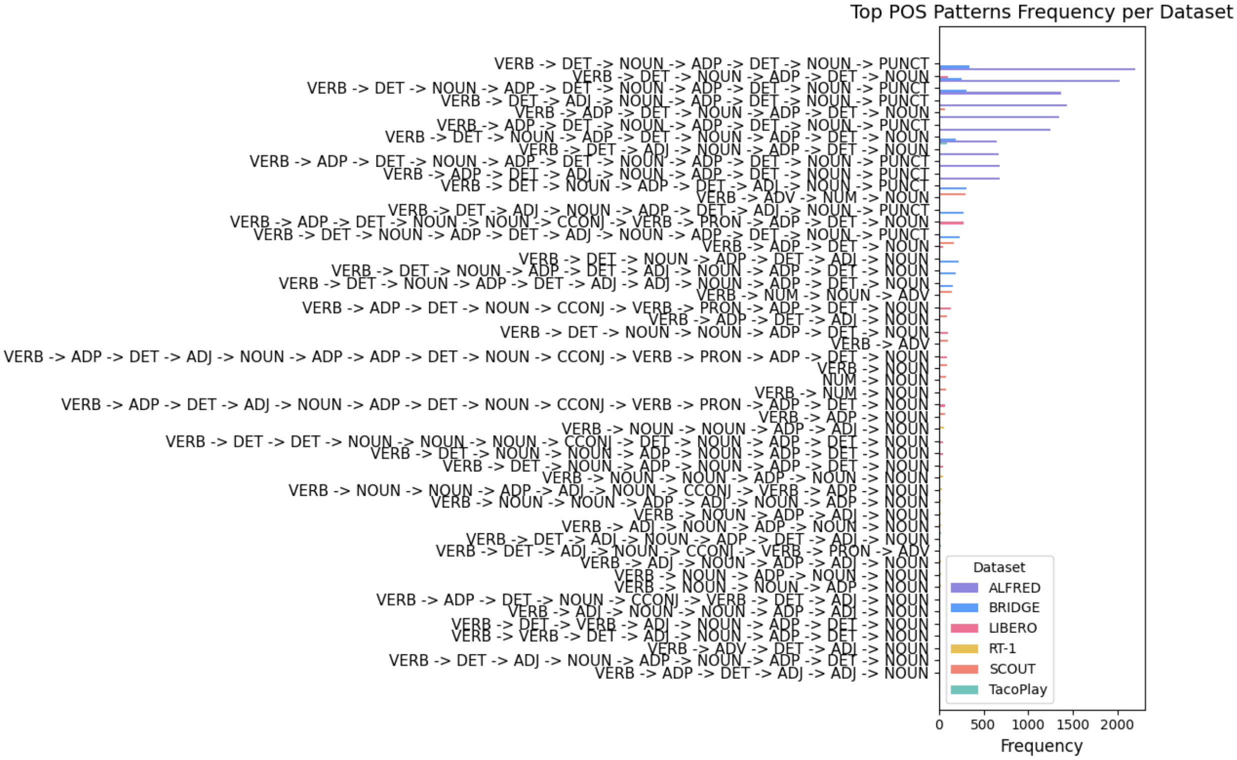


Fig. 11: POS parse pattern distribution on unique commands in the datasets.

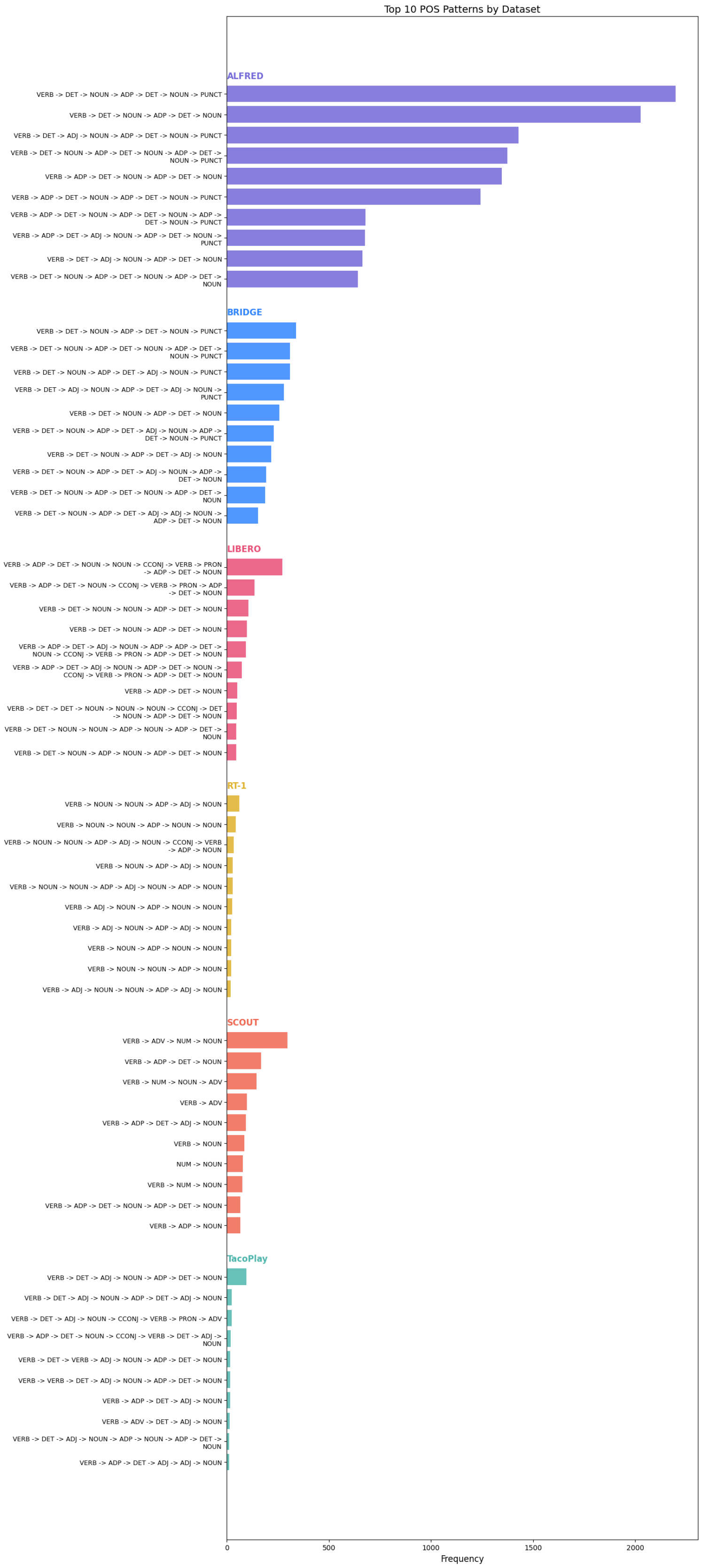
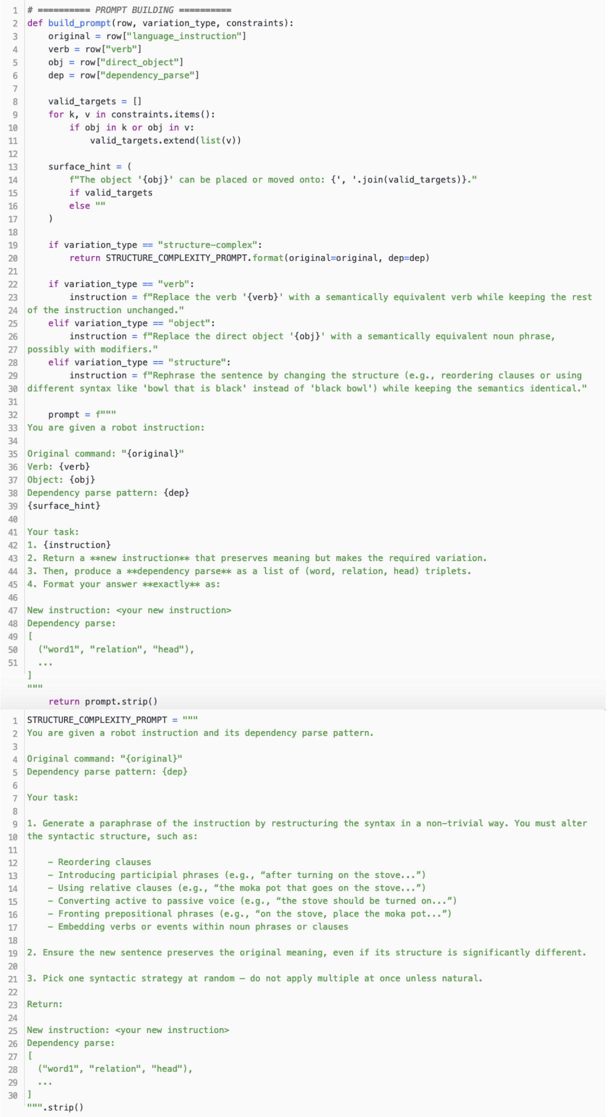


Fig. 12: Grouped view of top 10 POS parse patterns on unique commands in EAI datasets.



```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()
```

```python
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()
```

Fig. 13: Prompt used in paraphrase generation for test set. The parameter: `constraints` contains information from BDDL files which are then captured by `surface_hint`.

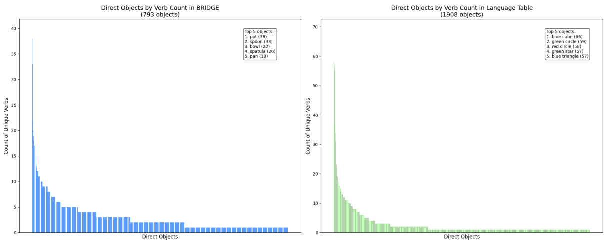


Fig. 14: Frequency Plot of Unique Verbs per Direct Object for Manipulation Datasets

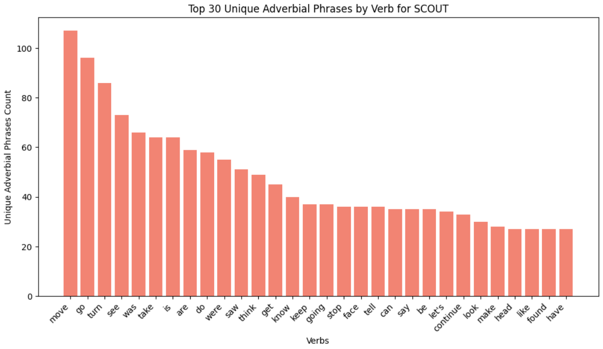


Fig. 15: VLN adverbials - limited to the top 20 verbs with most unique language use

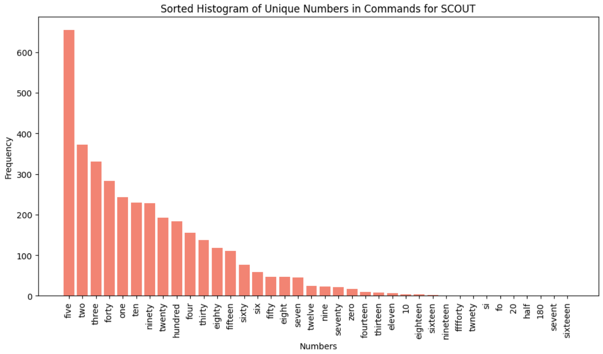


Fig. 16: SCOUT Numerics

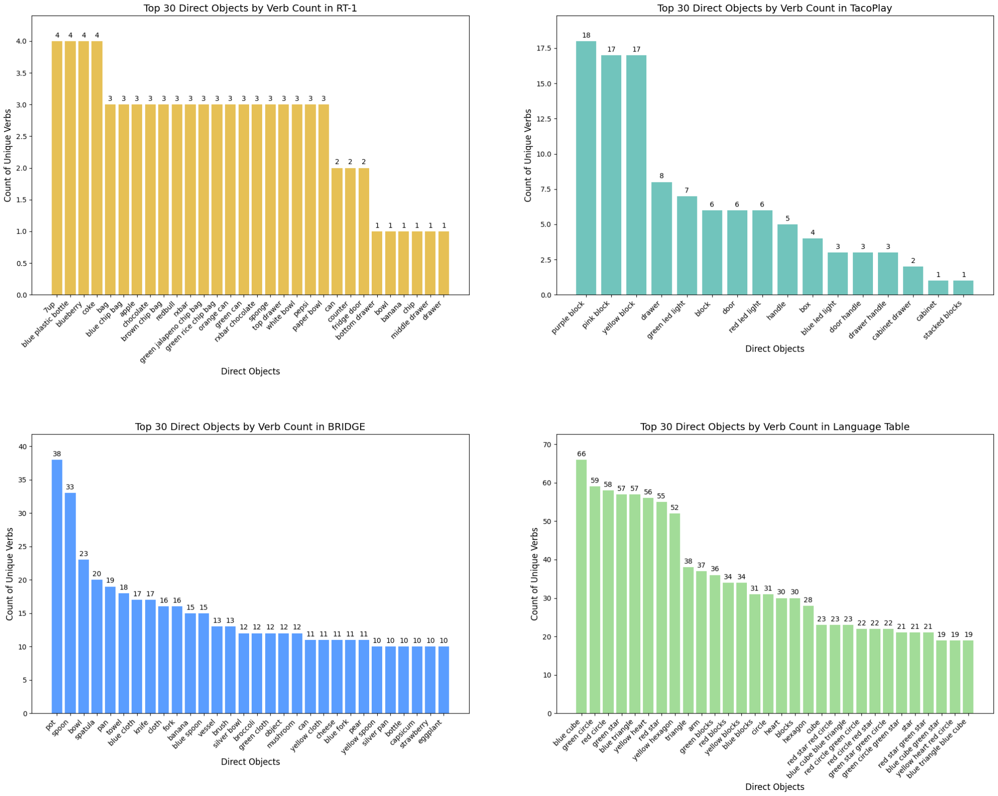
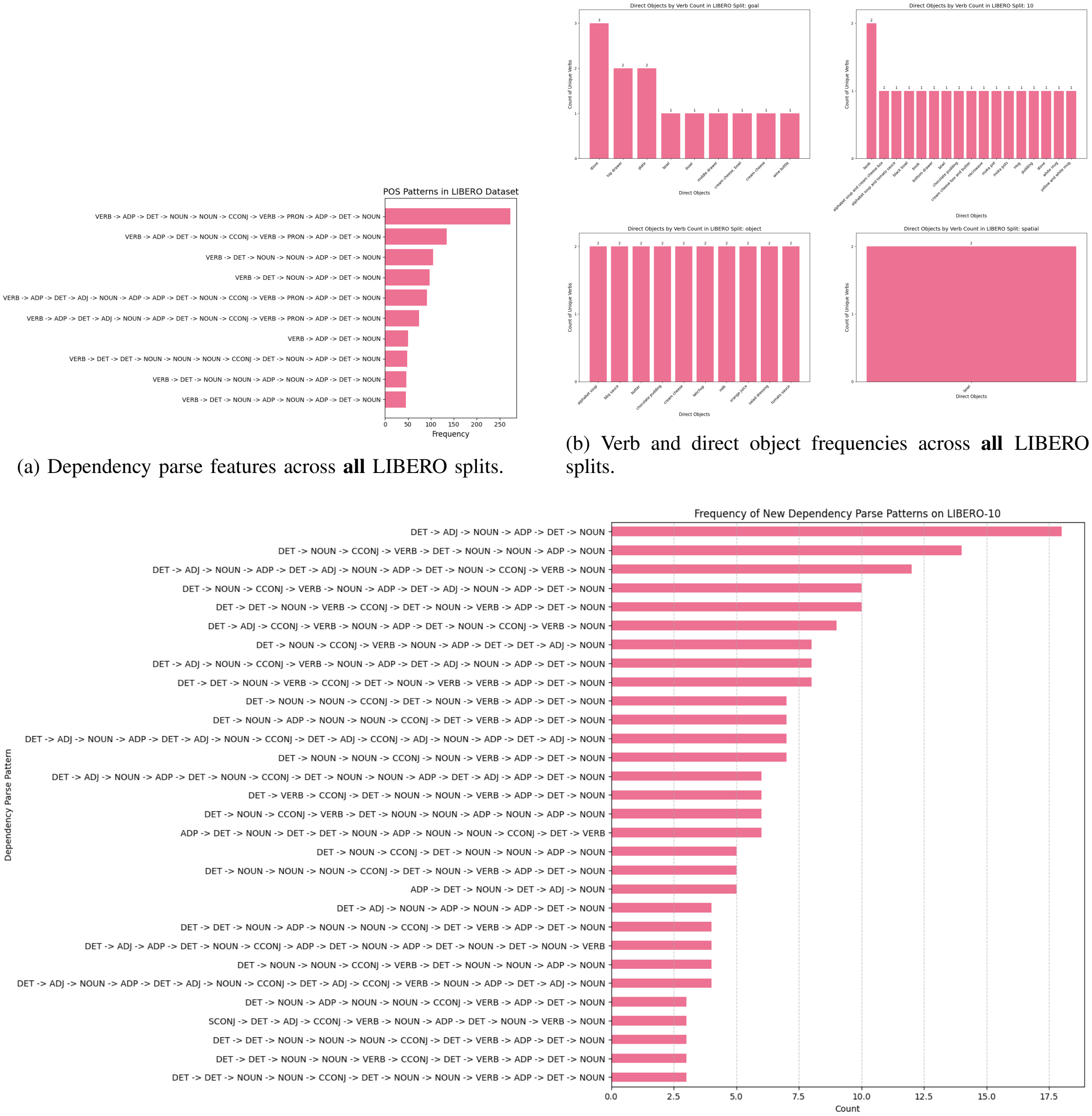


Fig. 17: Frequency Plot of Unique Verbs per Direct Object for Manipulation Datasets

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 VI: 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.

Fig. 18: 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).

Appendix VII
Limitations

Our analysis relies 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 IV-B. 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.

Acknowledgment

This work was supported by LANL (LA-UR-25-23357) (U.S. DOE NNSA Contract 89233218CNA000001).

References

[1] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 6, Apr. 2024. [Online]. Available: https://doi.org/10.1145/3649506

[2] Q. Zhang, K. Ding, T. Lv, X. Wang, Q. Yin, Y. Zhang, J. Yu, Y. Wang, X. Li, Z. Xiang, X. Zhuang, Z. Wang, M. Qin, M. Zhang, J. Zhang, J. Cui, R. Xu, H. Chen, X. Fan, H. Xing, and H. Chen, "Scientific large language models: A survey on biological & chemical domains," *ACM Comput. Surv.*, vol. 57, no. 6, Feb. 2025. [Online]. Available: https://doi.org/10.1145/3715318

[3] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. Wen, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, 2024. [Online]. Available: https://doi.org/10.1007/s11704-024-40231-1

[4] Z. Yang, F. Liu, Z. Yu, J. W. Keung, J. Li, S. Liu, Y. Hong, X. Ma, Z. Jin, and G. Li, "Exploring and unleashing the power of large language models in automated code translation," *Proc. ACM Softw. Eng.*, vol. 1, no. FSE, Jul. 2024. [Online]. Available: https://doi.org/10.1145/3660778

[5] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 2679–2713. [Online]. Available: https://proceedings.mlr.press/v270/kim25c.html

[6] E. Collaboration *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," 2024. [Online]. Available: https://arxiv.org/abs/2310.08864

[7] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," 2023. [Online]. Available: https://arxiv.org/abs/2306.03310

[8] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," 2024.

[9] AgiBot-World-Contributors, Q. Bu, J. Cai, L. Chen, X. Cui, Y. Ding, S. Feng, S. Gao, X. He, X. Hu, X. Huang, S. Jiang, Y. Jiang, C. Jing, H. Li, J. Li, C. Liu, Y. Liu, Y. Lu, J. Luo, P. Luo, Y. Mu, Y. Niu, Y. Pan, J. Pang, Y. Qiao, G. Ren, C. Ruan, J. Shan, Y. Shen, C. Shi, M. Shi, M. Shi, C. Sima, J. Song, H. Wang, W. Wang, D. Wei, C. Xie, G. Xu, J. Yan, C. Yang, L. Yang, S. Yang, M. Yao, J. Zeng, C. Zhang, Q. Zhang, B. Zhao, C. Zhao, J. Zhao, and J. Zhu, "Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems," *arXiv preprint arXiv:2503.06669*, 2025.

[10] S. Jiang, H. Li, R. Ren, Y. Zhou, Z. Wang, and B. He, "Kaiwu: A multimodal manipulation dataset and framework for robot learning and human-robot interaction," 2025. [Online]. Available: https://arxiv.org/abs/2503.05231

[11] Z. Wang, Z. Zhou, J. Song, Y. Huang, Z. Shu, and L. Ma, "Ladev: A language-driven testing and evaluation platform for vision-language-action models in robotic manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2410.05191

[12] S. Dey, J.-N. Zaech, N. Nikolov, L. V. Gool, and D. P. Paudel, "Revla: Reverting visual domain limitation of robotic foundation models," 2025. [Online]. Available: https://arxiv.org/abs/2409.15250

[13] J. Hejna, C. A. Bhateja, Y. Jiang, K. Pertsch, and D. Sadigh, "Remix: Optimizing data mixtures for large scale imitation learning," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 145–164. [Online]. Available: https://proceedings.mlr.press/v270/hejna25a.html

[14] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen, Y. Peng, F. Feng, and J. Tang, "Tinyvla: Toward fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 10, pp. 3988–3995, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272753287

[15] S.-M. Park and Y.-G. Kim, "Visual language navigation: a survey and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 1, p. 365–427, mar 2022. [Online]. Available: https://doi.org/10.1007/s10462-022-10174-9

[16] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. González, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolář, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. Ruiz, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbeláez, D. Crandall, D. Damen, G. M. Farinella, C. Fuegen, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, "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)*, June 2022, pp. 18 995–19 012.

[17] M. Shridhar *et al.*, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] S. M. Lukin *et al.*, "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)*. Torino, Italia: ELRA and ICCL, May 2024, pp. 14 445–14 458. [Online]. Available: https://aclanthology.org/2024.lrec-main.1259

[19] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," 2023. [Online]. Available: https:

//arxiv.org/abs/2212.06817

[20] H. Walke *et al.*, “Bridgedata v2: A dataset for robot learning at scale,” in *Conference on Robot Learning (CoRL)*, 2023.

[21] E. Rosete-Beas *et al.*, “Latent plans for task agnostic offline reinforcement learning,” 2022.

[22] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, “Interactive language: Talking to robots in real time,” *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.

[23] M. Fan, N. Gu, H. Qiao, and B. Zhang, “Intrinsic dimension estimation of data by principal component analysis,” 2010. [Online]. Available: https://arxiv.org/abs/1002.2050

[24] M. Verleysen and J. A. Lee, “Nonlinear dimensionality reduction for visualization,” in *Neural Information Processing*, M. Lee, A. Hirose, Z.-G. Hou, and M. K. Rhee, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 617–622. [Online]. Available: https://doi.org/10.1007/978-3-642-42054-2_76

[25] D. Cer *et al.*, “Universal sentence encoder for English,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 169–174. [Online]. Available: https://aclanthology.org/D18-2029

[26] N. Reimers *et al.*, “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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410

[27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[28] P.-A. Duquenne *et al.*, “SONAR: sentence-level multimodal and language-agnostic representations,” 2023. [Online]. Available: https://arxiv.org/abs/2308.11466

[29] T. Oates and D. Jensen, “The effects of training set size on decision tree complexity,” in *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, D. Madigan and P. Smyth, Eds., vol. R1. PMLR, 04–07 Jan 1997, pp. 379–390, reissued by PMLR on 30 March 2021. [Online]. Available: https://proceedings.mlr.press/r1/oates97b.html

[30] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, “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*, T. Linzen, G. Chrupała, and A. Alishahi, Eds. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355. [Online]. Available: https://aclanthology.org/W18-5446/

[31] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, “On the dangers of stochastic parrots: Can language models be too big?” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 610–623. [Online]. Available: https://doi.org/10.1145/3442188.3445922

[32] E. M. Bender, “The benderrule: On naming the languages we study and why it matters,” *The Gradient*, 2019.

[33] M. Honnibal *et al.*, “spacy: Industrial-strength natural language processing in python,” 2020.

[34] T. pandas development team, “pandas-dev/pandas: Pandas,” Feb. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3509134

[35] A. Moschitti, “Making tree kernels practical for natural language learning,” in *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, Apr. 2006, pp. 113–120. [Online]. Available: https://aclanthology.org/E06-1015

[36] Y. Zhang *et al.*, “Diagnosing the environment bias in vision-and-language navigation,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 890–897, main track. [Online]. Available: https://doi.org/10.24963/ijcai.2020/124

[37] K. Papineni *et al.*, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://aclanthology.org/P02-1040

[38] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[39] E. Collaboration *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models,” 2024. [Online]. Available: https://arxiv.org/abs/2310.08864

[40] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang *et al.*, “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,” 2025.

[41] M. Dehouck and C. Gómez-Rodríguez, “Data augmentation via subtree swapping for dependency parsing of low-resource languages,” in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3818–3830.

[42] H. Shi, K. Livescu, and K. Gimpel, “Substructure substitution: Structured data augmentation for nlp,” 2021. [Online]. Available: https://arxiv.org/abs/2101.00411