Grounded Semantic Role Labelling from Synthetic Multimodal Data for Situated Robot Commands

Claudiu Daniel Hromei[†] **and Antonio Scaiella**[‡] **and Danilo Croce**[†] **and Roberto Basili**[†]
[†]Department of Enterprise Engineering, University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy
[‡]Reveal S.r.l. - Via Kenia 21, 00144, Rome, Italy
{croce,basili}@info.uniroma2.it

Abstract

Understanding natural language commands in situated Human-Robot Interaction (HRI) requires linking linguistic input to perceptual context. Traditional symbolic parsers lack the flexibility to operate in complex, dynamic environments. We introduce a novel *Multimodal Grounded Semantic Role Labelling* (G-SRL) framework that combines frame semantics with perceptual grounding, enabling robots to interpret commands via multimodal logical forms. Our approach leverages modern Vision Language Models (VLMs), which jointly process text and images, and is supported by an automated pipeline that generates high-quality training data. Structured command annotations are converted into photorealistic scenes via LLM-guided prompt engineering and diffusion models, then rigorously validated through object detection and visual question answering. The pipeline produces over 11,000 image-command pairs (3,500+ manually validated), while approaching the quality of manually curated datasets at significantly lower cost.

1 Introduction

Robots operating in human-centred environments require robust methods for *situated language understanding*, enabling the interpretation of natural language commands grounded in their current perceptual context (Tellex et al., 2011; Shridhar et al., 2020; Padmakumar et al., 2022; Xiao et al., 2024). Such interpretation benefits from structured semantic representations, or *primitives*, that bridge linguistic instructions and executable robot plans while remaining intuitive for users (Vanzo et al., 2020).

Consider the instruction *"Bring the phone on the bed in the living room"* (see Figure 1). To execute this command, a robot must resolve linguistic references to concrete visual entities (e.g., identifying the phone), reason about spatial relationships (e.g., the position of the phone relative to the bed), and ground symbolically represented locations (e.g., the living room) to known topological places. We formalise this task as *grounded semantic role labelling* (G-SRL), a hybrid representation combining predicate-argument structures with explicit perceptual grounding (Gildea and Jurafsky, 2002; Yang et al., 2025), in the form of absolute position bounding boxes that refer to the mentioned linguistic entities in the image:

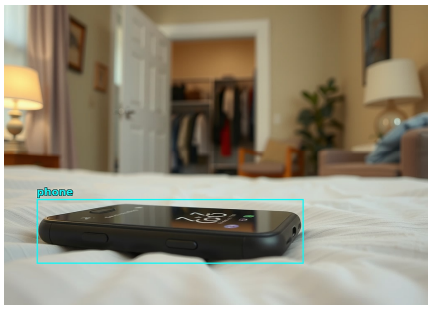


Figure 1: Example of linking command roles to visual referents in the perceived scene.

BRINGING(
  (THEME, "*phone*" , [170, 541, 852, 940])
  (GOAL, "*living room*", <ROOM>)
)

Here, $[170, 541, 852, 940]$ refers to the bounding box drawn around the phone entity in the image and is expressed as a list of absolute numbers $[x_1, y_1, x_2, y_2]$. Notably, the instruction is ambiguous: the phrase *"on the bed in the living room"* could refer either to the current location of the phone or to the intended destination. If the phone is not located on the bed, interpreting the *bed* as the GOAL is to be preferred to it being the SOURCE. Our G-SRL framework captures such ambiguities by adapting roles according to the perceived scene.

Unlike existing approaches that typically assume fixed, context-agnostic semantic role structures (He et al., 2017), our Grounded SRL framework dynamically adjusts the argument interpretations accord-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 23759–23782
November 4-9, 2025

ing to the perceptions of the robot about the environment. This allows the same command statement to yield different frames and grounding strategies, depending on the observed scene. The emergence of such dynamic context-sensitive representations captures more closely the *iconic* and *categorial* nature of non-symbolic elements in the environment (Harnad, 1990). It thus realises a *symbol grounding* mechanism as a suitable *bridge* between language and the real world and a better model for robotic command understanding processes (Harnad, 1990). To our knowledge, this is the first framework to combine fine-grained semantic parsing with perceptual grounding in the context of situated Human-Robot Interaction (HRI). While existing resources like the HuRIC corpus (Vanzo et al., 2020) provide structured semantic annotations of natural language commands, they are inherently limited to symbolic, textual representations and fully neglect perceptual grounding, required for realistic human-robot interaction. Building multimodal datasets that align linguistic instructions with corresponding visual contexts is both costly and labour-intensive (Yang et al., 2025; Zhao et al., 2023), often requiring controlled environments or complex Wizard-of-Oz setups (Anderson et al., 2018). Moreover, mapping linguistic expressions to a knowledge base can be challenging and error-prone, thus requiring extensive manual intervention.

To address these limitations, we propose a scalable, end-to-end pipeline for the direct generation of grounded multimodal training data from text commands. Our approach comprises three core components: *i*) **extracting structured frame-semantic representations** from natural language commands, from which we derive explicit constraints on entities, spatial relations, and object states; *ii*) **generating photorealistic images** using LLM-guided prompt engineering that reflects these semantic constraints; and *iii*) **validating the generated scenes** to ensure consistent semantics and prevent hallucinations. This final stage combines object detection and visual question answering (VQA) to check whether the generated images satisfy the constraints extracted in step (i). Object detectors verify the visual accessibility of relevant entities (e.g., `visible(phone)`, `not visible(living room)`), while VQA models assess relational, spatial or other object properties (e.g., whether the phone is near the bed, or whether the oven is open). In our implementation, we use GroundingDINO (Liu et al., 2023) for open-vocabulary object detection and MiniCPM-V for visual reasoning. This design builds on the principles of domain randomisation (Tobin et al., 2017) and synthetic data generation for embodied tasks (Gao et al., 2022; Lin et al., 2022; Pramanick et al., 2023), allowing the automatic creation of diverse, contextually accurate visual scenes. Unlike text-only annotated corpora, our method captures both the linguistic and perceptual dimensions, making it a comprehensive approach to grounded semantic role labelling. We applied our pipeline to augment the HuRIC corpus, generating over 11,000 image-command-logical form triplets. The data is used to fine-tune state-of-the-art vision-language models. Our experiments show that models trained on the automatically validated data sets perform comparably to those trained on manually curated instances, significantly reducing the cost and effort of multimodal data creation.

Contributions. We present: *i*) a novel G-SRL framework for perceptually grounded, context-aware interpretation of robot commands; *ii*) an automatic pipeline that synthesises high-quality multimodal training data from text-only input; *iii*) a robust validation procedure combining object detection and VQA to ensure rich and diverse semantics in synthetic data[1].

The remainder of this paper surveys related work (Section 2). Section 3 presents our G-SRL framework. Section 4 details the multimodal data generation pipeline. Section 5 reports the experimental results. Section 6 concludes the paper.

2 Background and Related Work

Situated Language Understanding (SLU) seeks to ground natural language commands in the perceptual context of an agent. Early symbolic systems mapped language to handcrafted spatial or action templates (Tellex et al., 2011; Chen and Mooney, 2011), but lacked robustness to ambiguity and generalisation beyond predefined environments.

Recent approaches adopt a multimodal perspective, leveraging visual and textual inputs to improve grounding performance. Simulated environments like ALFRED (Shridhar et al., 2020) and TEACh (Padmakumar et al., 2022) have played a central role, offering large-scale datasets of language-conditioned tasks in 3D domestic environments. However, these frameworks are constrained by

[1]Data and code are released via a public repository, accessible at `https://github.com/crux82/GroundedSRL4HRI`.

fixed libraries and limited scripted scene logic, making it difficult to generate visually diverse and semantically precise scenarios. For example, ensuring that an object is simultaneously "*on the bed*" and "*near the window*" often requires manual scene tuning. HOLODECK (Yang et al., 2024), which procedurally generates 3D scenes from textual descriptions, addresses scalability but lacks mechanisms for enforcing fine-grained spatial or relational constraints: this often results in semantically inconsistent renderings. Grounded SRL provides a principled way to map linguistic predicates and arguments to real-world referents, bridging symbolic semantics and perceptual grounding (Gildea and Jurafsky, 2002; He et al., 2017). Prior systems (Bisk et al., 2018; Misra et al., 2017) modelled this alignment through direct symbolic mappings, with limited linguistic coverage and generalisation capabilities. More recent efforts like SHERLOCK (Hessel et al., 2022) integrate perception and language via probabilistic inference or cross-modal attention, but still treat parsing and grounding as separate stages. These heavily rely on manually annotated datasets with constrained visual variation. Notably, few existing methods allow dynamic adaptation of frame structures based on visual context, a critical ability for situated agents operating in partial observability conditions. (Yang et al., 2016) address grounding by visually aligning arguments of a given frame with perceptual entities, assuming the frame is already provided as textual input. While this represents an important step toward connecting language with visible objects, the approach remains largely symbolic and template-driven, and does not support interpretation for free and informal natural language commands. In contrast, a G-SRL framework should jointly interpret entire commands and dynamically adapt frame-semantic structures to the perceptual context, offering greater flexibility and more context-sensitive grounding in HRI scenarios.

Vision-Language Models (VLMs) such as UNITER (Chen et al., 2020), BLIP (Li et al., 2023), and PaLM-E (Driess et al., 2023) have dramatically improved multimodal understanding by learning joint embeddings across text and images. Instruction-tuned models like MiniCPM-V (Yao et al., 2024) push this further, enabling compositional visual reasoning. However, most VLMs are trained for free-form tasks (e.g., captioning, VQA) and lack the ability to produce structured outputs (e.g., logical forms or frame-role graphs) required for symbolic execution. Furthermore, their reliance on large, human-curated datasets creates a bottleneck for adapting to new robotic domains or low-resource instruction types.

Synthetic generation techniques have been widely adopted to address data scarcity. Domain randomisation (Tobin et al., 2017) and 3D simulation-based augmentation (Anderson et al., 2018; Shridhar et al., 2020) offer scalable alternatives but often fail to enforce relational constraints. Diffusion-based methods like those used in DIALFRED (Gao et al., 2022) or EGOVLP (Lin et al., 2022; Pramanick et al., 2023) improve visual realism, yet suffer from a lack of fine-grained control: objects may be hallucinated, spatial relations misrepresented, and constraints violated. These systems also typically assume fixed visual layouts or asset constraints, limiting diversity and domain transfer. The HuRIC corpus (Vanzo et al., 2020) provides FrameNet-style semantic annotations for robot commands, but lacks visual grounding for situated execution. Building on this, we introduce a fully automated pipeline that transforms logical forms into photorealistic, semantically validated training data. Our approach uniquely combines frame-semantic parsing, perceptual grounding, and scalable multimodal data generation, without manual scene design.

3 Grounded Semantic Interpretation

Robust situated HRI requires a representation that (i) is directly executable by a robot, (ii) adapts to what the robot perceives, and (iii) remains human-interpretable. We therefore cast command understanding as a mapping $f : \langle \mathcal{C},\ I \rangle \longrightarrow l$, where l is a relational, frame-based representation (Baker et al., 1998; Fillmore, 1985) consisting of a list of frames $l = [F_1, \ldots, F_m]$, whose $F_k = \{(r_j, h_j, g_j)\}_{j=1}^{n_k}$ correspond to a set of n_k of roles, whose triples include frame element types r_j, lexical heads h_j and grounding information g_j.

Notice that g_j can be **visual**, when expressed as a bounding box $[x_{min}, y_{min}, x_{max}, y_{max}]$ relative to the image (I), or **symbolic**, as a linguistic naming of abstract concepts, such as known locations, e.g. <ROOM> or operational states, e.g., <STATUS>.

For instance, given the command *"Bring the phone on the bed in the living room"* and an image containing a phone and a bed, the only frame in l might be:

BRINGING(

(THEME, "*phone*" , [170, 541, 852, 940])
(GOAL, "*living room*", <ROOM>)
)

where two frame elements are found: the THEME role is visually grounded to the bounding box representing the *phone*, while the GOAL role is symbolically linked to a known location. Note that the bed is not explicitly included, as the bounding box around the phone is sufficient for the robot to resolve the reference, reflecting the way robots often simplify spatial reasoning.

The structure adapts to perception. If the phone is already visible in the living room, the command yields a different frame:

BRINGING(
(THEME, "*phone*", [483, 548, 531, 633]),
(GOAL, "*bed*", <MISSING>)
)

In this case, the implicit spatial context changes the action that must be carried out by changing the destination, i.e., the GOAL role, allowing the robot to proceed to bring the phone to the new location. However, not all frame elements can be visually grounded. To capture these cases, we define a set of symbolic tags: <ROBOT> for the agent (e.g., the command executor); <PERSON> for a human interlocutor (e.g., the speaker or recipient); <ROOM> for known locations from the internal knowledge base; <POSITION> for deictic or underspecified spatial references (e.g., *here*, *there*); <STATUS> for operational states (e.g., *on*, *closed*); <ITEM> for unspecified or pronominal references (e.g., *it*, *this*); and <MISSING> for expected referents that are not visible in the current perceptual domain, i.e., a *not visible* object. More examples of logical forms and their corresponding prompts can be found in Appendix A.

To generate the structured logical form l, we fine-tune a Vision-Language Model as a perceptual semantic parser: given a command $\mathcal{C}$ and an image I, it autoregressively outputs a grounded, frame-based interpretation. Models such as MiniCPM-V 2.6 (Yao et al., 2024) or Qwen2.5-VL (Bai et al., 2025) are well suited for this task, as they can attend to both modalities while producing structured outputs. The command $\mathcal{C}$ is tokenized into language embeddings $\mathbf{h}_{\mathcal{C}} \in \mathbb{R}^{n \times d}$, and the image I is mapped to visual embeddings $\mathbf{h}_I \in \mathbb{R}^{m \times d}$ using a visual encoder (e.g., SigLIP (Zhai et al., 2023) or ViT (Dosovitskiy, 2020)). These are fused (via cross-attention or joint projection) into a multimodal representation $\mathbf{h}_{\mathcal{C},I} \in \mathbb{R}^{(n+m) \times d}$, which conditions the decoder. The decoder generates a logical form l as a sequence of frame labels, roles, semantic heads, and groundings (e.g., `BRINGING(, (Theme, phone, [170,541,852,940]), (Goal, living room, <ROOM>))`.

Despite the finite set of frames, the task demands fine-grained multimodal reasoning: entity recognition, reference resolution, and spatial grounding. This process is data-intensive, especially for grounding via bounding boxes, which requires visually consistent supervision across diverse contexts. To address this, we generate semantically validated multimodal training data directly from textual inputs.

4 Multimodal Data Generation

Multimodal training data required in grounded language understanding must reflect both the linguistic structure of commands and some spatial or relational constraints reflected in the possible perceptual grounding. However, collecting such data at scale remains a challenge. Existing simulators offer limited visual diversity and require manual scene design, while diffusion-based image generators, though visually rich, lack semantic control and can easily hallucinate and result in misleading training evidence.

To overcome these limitations, we introduce an intelligent pipeline capable of generating consistent and photorealistic training data from structured semantic representations. By combining prompt generation, image synthesis, and automated post-hoc validation, our method aims to ensure that each generated image is both visually plausible and semantically aligned with the intended interpretation of the command. Notice that the intended interpretation refers specifically to the semantic interpretation that each individual generated image satisfies, depending on the visible context. The primary motivation is precisely to expose future models to genuine perceptual ambiguities that naturally arise in embodied situations. Accordingly, the intelligent pipeline generates multiple distinct images for each command, explicitly representing diverse semantic groundings. For instance, the instruction "*Bring the phone on the bed*" may yield:

- An image with a visible bed and the phone placed somewhere else, grounding the GOAL (*the bed*) visually with a bounding box.

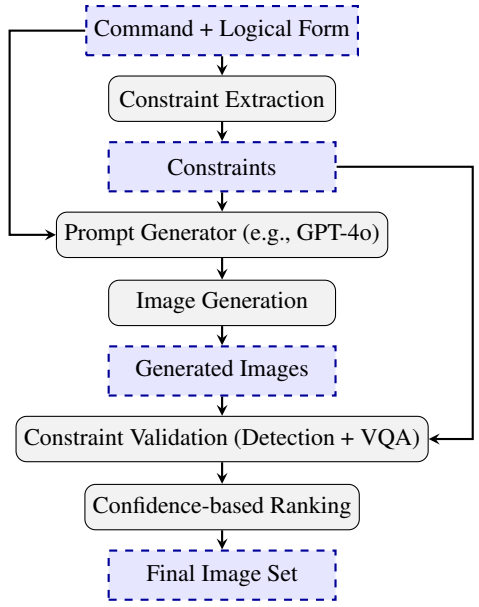


Figure 2: Dataset generation pipeline from semantic interpretation to validated multimodal data.

- An image without any visible *bed*, thus grounding the GOAL role symbolically as `<MISSING>`.

Crucially, this process should involve no human intervention, enabling the scalable creation of grounded multimodal datasets. Figure 2 provides an overview of the proposed pipeline, outlining the four main stages: constraint extraction, prompt generation, image synthesis, and multimodal validation. Unlike traditional 3D simulators, our method decouples linguistic interpretation from physical scene construction, enabling the generation of diverse and contextually rich training data without relying on handcrafted environments. Each training instance begins with a natural language command and its corresponding logical form, which jointly define the intended semantics. The logical form specifies both the core requests as frames (e.g., BRINGING or TAKING) and grounded frame elements through bounding boxes or symbolic references (e.g., `<ROOM>`, `<STATUS>`). These structures guide the generation of photorealistic images that faithfully reflect the spatial, relational, and referential constraints expressed in the original command.

Constraint Extraction. In real-world human-robot interaction, visual access to all referenced entities cannot be assumed: a robot may perceive some, all, or none of the objects mentioned in a command. To reflect this variability, we target multiple perceptual situations for one instruction $\mathcal{C}$, each corresponding to a distinct combination of visible and nonvisible referents. These scenarios drive the construction of alternative logical forms: sometimes, frame elements are visually realised via a bounding box. Alternatively, they are symbolically annotated (e.g., `<MISSING>`). This approach enables robust and flexible generalisation: a command such as *"Take the phone on the bed in the living room"*, has two roles (`Theme`, `Location`) and yields four distinct interpretations based on visibility: phone and bed can be both, or independently, visible, with bounding boxes, or not. The variants result in four distinct logical forms for an image I, which in turn defines semantic constraints: a set $C(\mathcal{C}) = A \cup S \cup O$ can be used to guide the prompt generation and image validation stages. Here **Accessibility constraints** (A) enforce the visibility of specific objects. The **Spatial constraints** (S) specify geometric relationships, such as `ontop(phone, bed)`, that are the formulas to be satisfied in a target image I, e.g. true only if the phone is on the bed. Finally, **Object State constraints** (O) are true only when an object is in a given state in the target image I, such as `open(door)` or `switched-on(tv)`.

For instance, a command $\mathcal{C}$ in a partially visible scenario might yield constraints $C(\mathcal{C})$ such as:

$$A = \{\texttt{visible(phone)}, \texttt{visible(bed)}, \neg\texttt{visible(living room)}\}$$
$$S = \{\texttt{ontop(phone, bed)}\} \quad O = \emptyset$$

$C(\mathcal{C})$ not only governs what should appear in the image, but also determines the format of the logical form that the model should learn to predict. By varying perceptual conditions in this controlled way, we ensure that the model is exposed to realistic ambiguities and learns to treat visual grounding in a visual context-dependent manner, and not just by linguistic descriptions.

Prompt Generation. Given a set of constraints $C(\mathcal{C})$, we use an LLM, such as GPT-4o, to generate rich, semantically controlled textual prompts suitable for guiding diffusion-based image synthesis. Inspired by recent approaches like Holodeck (Yang et al., 2024), which use LLMs to orchestrate entire simulated home layouts, we adopt a lighter but equally expressive strategy: we produce short but vivid scene descriptions that encode all semantic constraints specified in $C(\mathcal{C})$, including required entities, spatial relationships, and object states. This generation process supports two key goals. First, it ensures that each prompt re-

flects the intended configuration of visible and non-visible elements, including explicit negations (e.g., "no cups or robots") or symbolic references (e.g., <MISSING>). Second, it introduces visual diversity by prompting the same scenario from multiple viewpoints, such as *close-up*, *wide shot*, *long shot*, *low-angle*, and *high-angle*, thus enabling the training of grounding models that are robust to position changes. For example, a BRINGING frame involving a phone and bed could yield prompts such as: *"A close-up of a smartphone clearly visible on the bed in a cozy bedroom"*, or *"A high-angle view of a bedroom where a phone is resting near the foot of the bed"*. The LLM ensures that each variation remains fluent, realistic, and consistent with the constraints $C(\mathcal{C})$, even in negative or ambiguous cases. Prompts are generated using metaprompt templates that embed the content of $C(\mathcal{C})$ via slot filling, while allowing for lexical variability and contextual naturalness. This setup avoids rigid templates while maintaining strict control over consistency. Full metaprompt examples, including cases of entity inclusion and exclusion, are provided in Appendix B.

Image Generation. Each structured prompt is passed to a diffusion-based image synthesis model (e.g., `FLUX.1-schnell`[2]), which transforms the textual description into a photorealistic scene that visually expresses the semantic constraints encoded in C. The use of diffusion models enables high-quality image generation without requiring manual scene assembly or rigid 3D simulation pipelines. Compared to simulation platforms like AI2-THOR (Kolve et al., 2022), which rely on fixed assets and scripted layouts, our approach enables greater diversity and flexibility. It produces semantically faithful yet visually varied scenes across a range of indoor contexts, as each image is directly conditioned on constraint-driven prompts aligned with the original command $\mathcal{C}$.

Constraint Validation. To ensure that each generated image faithfully reflects the intended semantics of the command, we validate it against the constraint set $C(\mathcal{C}) = A \cup S \cup O$ using a two-stage process. This step is crucial, as diffusion-based models often produce visually plausible yet semantically inaccurate outputs, including hallucinated objects, incorrect spatial relations, or missing referents. In the first stage, we apply GroundingDINO[3], an open-vocabulary object detection model, to verify accessibility constraints A. For each constraint $a_i \in A$, i.e., an expected or forbidden object, the model returns a bounding box and an associated probability p_i: for all $a_i \in A$, the confidence score $\sigma_i^{GD}(I) = p_i$. The missing detection (or very low bounding box probability scores) is interpreted as a strong signal that an object is not visually available. This is a direct support to negative constraints, for which $\sigma_i^{GD}(I) = 1 - p_i$. In cases where objects are expected to be visible, we extract both the detection probability and the spatial grounding, which can be reused during training if the image is retained.

In parallel, we use MiniCPM-V 2.6[4] to verify spatial ($s_j \in S$) and state-based ($o_k \in O$) constraints through targeted yes/no questions automatically derived from the constraint set. The model's confidence in the first generated token ("*yes*" or "*no*") is used as the probability ($\sigma_j^S = p_j$ or $\sigma_k^O = p_k$) of the expected outcome, and inverted ($\sigma_j^S = 1 - p_j$ or $\sigma_k^O = 1 - p_k$) when supporting negative cases. This process results in a set of $|A| + |S| + |O|$ probabilistic scores σ, each indicating how well the image satisfies specific constraints in $C(\mathcal{C})$. These scores are combined into a global consistency score $\sigma(I)$ for a generated image I, via log-likelihood summation:

$$\sigma(I) = \sum_{a_i \in A} \log(\sigma_i^{GD}) + \sum_{s_j \in S} \log(\sigma_j^S) + \sum_{o_k \in O} \log(\sigma_k^O)$$

Confidence-Based Ranking and Selection. Rather than enforcing rigid thresholds on individual constraints (e.g. a_i), the high-throughput of our pipeline is exploited to generate multiple candidate images I per command and select those with the highest aggregate scores $\sigma(I)$. The top-k selection strategy over $\sigma(I)$ balances precision and coverage: while some semantically inaccurate samples may still receive high $\sigma(I)$, this happens very rarely, as confirmed by our empirical analysis.

Crucially, even without assuming perfect filtering, the ranking mechanism supports the emergence of rich semantic phenomena, i.e. scenes where language, structure, and perception are most tightly aligned. In later sections, we show how training on these sets of decreasing quality levels (i.e. $\sigma(I)$ scores) supports good generalisation and

[2] https://huggingface.co/black-forest-labs/FLUX.1-schnell

[3] https://github.com/IDEA-Research/GroundingDINO

[4] https://huggingface.co/openbmb/MiniCPM-V-2_6

Top-k	Malformed	Anomalous Elements	BBox Errors	State Errors	Spatial Errors
Top-1	1.18%	1.40%	2.32% / 2.37%	0.16% / 0.88%	1.26% / 2.87%
Top-2	1.34%	1.45%	2.40% / 2.43%	0.32% / 1.75%	1.42% / 3.23%
Top-3	2.30%	1.95%	2.69% / 2.74%	0.32% / 1.75%	1.50% / 3.42%

Table 1: Manual validation error rates across Top-1, Top-2, and Top-3 image subsets. For BBox, State, and Spatial categories, both absolute error rates (over all images) and relative rates (only over applicable cases) are reported.

maximises the benefits from increasing data diversity: residual noise is evidently limited. The proposed validation pipeline, beyond its filtering role, acts as an effective inductive scaffold for grounded learning by pushing for weak but effective supervision.

5 Experimental Evaluation

We evaluate our approach by (*i*) measuring the precision of our dataset validation, and (*ii*) fine-tuning VLMs on validated vs. unvalidated data to assess robustness and grounding quality at scale.

Data Generation and Validation. To assess the quality of the generated data, we apply our multimodal pipeline to the HuRIC corpus[5], which includes 650 commands annotated with frame semantics. After discarding utterances involving human references (e.g., "*follow John*"), we retain 619 commands suitable for visual rendering.

Each command is used to generate approximately 90 candidate images via prompt-based synthesis. These are scored through our multimodal constraint validation process (GroundingDINO for object detection, MiniCPM-V for spatial and state verification). Images are ranked by $\sigma(I)$ confidence score, i.e. the cumulative log-likelihoods across accessibility, spatial and object constraints.

We define the resulting collection as the *Complete Dataset*, comprising all automatically validated samples across Top-1, Top-2, and Top-3 ranking levels. Here, Top-k indicates the retention of the k highest-scoring images per command. This yields approximately 1,265 images per level and 3,796 total (about 6 per command), which are used for both training and evaluation. To estimate the true quality of these samples, we perform a full manual validation of the Complete Dataset. Two annotators independently reviewed all images according to five criteria: malformed rendering, anomalous or implausible elements, incorrect bounding boxes, inconsistent object states, and invalid spatial relations. An image was accepted into the *Validated Dataset* only if it passed all five validation dimensions without any flagged errors. In case of disagreement between annotators, conflicts were resolved through discussion until consensus was reached. The resulting set includes 3,399 images (about 93% of the Complete set, up to Top-3), averaging 5.5 verified samples per command.

Annotation was conducted by two annotators via a structured interface presenting each image alongside the corresponding command, required and forbidden entities, expected object states, and spatial constraints. Validation options followed standardised categorical labels, with minimal subjectivity. More information in Appendix D. Table 1 reports the distribution of error types across the Top-1, Top-2, and Top-3 subsets. For bounding box, state, and spatial errors, we report both absolute values (over the full set) and relative values (restricted to applicable cases). Malformed and anomalous images remain below 2% for Top-1 and Top-2 sets, but rise to 2.30% for the Top-3 set. Bounding box errors affect 2.32% of Top-1 images (2.37% relative), while state and spatial violations remain under 1.5%. As expected, noise increases slightly with lower-ranking images, reflecting the trade-off between data volume and semantic precision. This validation confirms that our ranking strategy reliably filters noise, yielding high-quality, semantically consistent images. The Validated Dataset serves as a robust benchmark for evaluation, while the larger Complete Dataset supports scalable training with minimal annotation cost. In the next experiments, we compare model performance across both sets and assess robustness to unvalidated data.

Model Fine-Tuning and Evaluation. We evaluate the task of grounded semantic interpretation using two recent vision-language models: Qwen-VL 2.5 and MiniCPM-V 2.6. The objective is to generate a structured FrameNet-style logical form from a natural language command and an associated image. Outputs include the correct semantic

[5] https://github.com/crux82/huric

Fine Tuning	Frames	Frame Elements	Semantic Head	Grounding		
				F1	IoU	IoU (match-only)
NO	*62.39%*	*33.83%*	*24.52%*	*48.63%*	*6.94%*	*20.82%*
Top-1	**97.69% / 97.69%**	**96.09% / 95.79%**	**94.78% / 94.90%**	84.78% / 83.81%	44.72% / 43.75%	57.68% / 54.84%
Top2	97.32% / 95.79%	95.84% / 94.42%	93.22% / 93.13%	83.15% / 83.26%	47.30% / 39.31%	66.82% / 58.32%
Top-3	96.58% / 96.12%	94.37% / 95.18%	93.51% / 94.74%	**86.15% / 88.6%**	**52.12% / 52.44%**	**71.37% / 67.64%**

Table 2: MiniCPM-V 2.6 performance on grounded semantic interpretation. Metrics (micro-F1 %) cover Frames, Frame+Element pairs, and full semantic structures (incl. surface forms). Grounding is evaluated via F1 on symbolic tags (e.g., <ROOM>, <SPEED>), average IoU on all expected boxes, and IoU on correctly matched elements (*match-only*). Results shown for zero-shot (*NO*) and fine-tuning on Top-1/2/3 ranked images from Complete and Validated sets (format: *Complete* / *Validated*).

frame, its frame elements (FEs), their lexical fillers, and, when applicable, grounded bounding boxes. Representative examples are shown in Appendix A.

We consider both zero-shot and fine-tuned settings. In the supervised case, models are trained on either the Complete or Validated version of the dataset, enabling a direct comparison of performance under different levels of annotation noise. All experiments use English text and follow an 80/10/10 split for training, validation, and test. We thus consider three progressively larger training sets: Top-1, Top-2, and Top-3, containing respectively 1,016, 2,031, and 3,046 training instances, corresponding to the top-ranked 1, 2, or 3 images per command, cumulatively. A manually validated subset of these sets includes 953 (Top-1), 1,911 (Top-2), and 2,865 (Top-3) examples. The corresponding development sets contain 124, 248, and 372 instances (or 104, 216, and 327 for the validated subset). Evaluation is performed on the validated Top-1 set for all experiments to ensure a consistent test benchmark across training conditions[6]. Full training details and hyperparameters are provided in Appendix E. We report microaveraged Precision, Recall, and F1 scores over multiple levels of interpretation: *i*) frame classification, *ii*) frame plus frame element (FE) assignment, and *iii*) full semantic tuples including the lexical head of each FE (case insensitive, whitespace normalised). We additionally report: *iv*) F1 for special symbolic tags (e.g., <ROOM>, <SPEED>) that do not require visual grounding, and *v*) average Intersection over Union (IoU) for visually grounded arguments. All metrics are computed per command, enabling a comparison between models and training conditions. Table 2 reports the performance of MiniCPM-V 2.6 on the Visual Grounded SRL task on the Validated Test Set, both in zero-shot (row *NO*) and after fine-tuning on progressively larger subsets of the generated dataset. Metrics are micro-averaged, and scores are reported as trained on the *Complete / Validated* sets, where applicable. In the zero-shot setting, the model exhibits limited interpretive ability: while Frame prediction reaches 62.39% F1, performance sharply drops for Frame Elements (33.83%) and for surface-level realisations (24.52%). Visual grounding is particularly weak, with average IoU at 6.94% and semantic tag F1 at 48.63%, indicating that the zero-shot model lacks the inductive bias to associate structured semantics with visual evidence. Fine-tuning leads to a substantial performance boost. On Top-1 training data, the model surpasses 97% F1 on Frame prediction, over 95% on Frame Elements, and around 95% on complete semantic tuples. While interpretive scores are nearly identical across the Complete and Validated datasets, grounding quality is slightly higher for the Complete dataset, with better F1 (84.78% vs. 83.81%), IoU (44.72% vs. 43.75%), and bounding box alignment. As more image candidates are added (Top-2 and Top-3), interpretive performance slightly declines due to increased noise, yet grounding consistently improves. Average IoU grows from 44.72% to 52.12% (Complete), and from 43.75% to 52.44% (Validated). The *IoU (match-only)* metric, which only considers elements correctly identified and grounded, also improves markedly (from 57.68% to 71.37% in the Complete dataset), confirming that visual diversity enhances spatial grounding quality. To further investigate whether grounding performance continues to improve with additional data, we extended training to include Top-4 through Top-10 images per command (see Figure 3). Each increment adds

[6]To increase training diversity, we apply a basic data augmentation technique by horizontally flipping each image and adjusting the corresponding bounding boxes. This procedure doubles the number of training samples for each set. Although flipping may occasionally introduce inconsistencies with the original command, e.g., left / right references, it consistently improves performance across all training configurations.

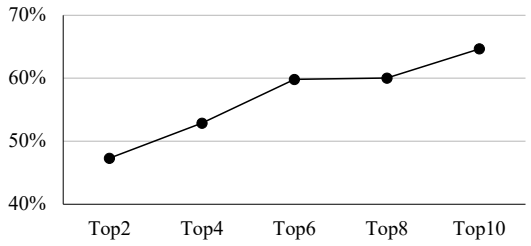


Figure 3: Grounding accuracy improves with larger training sets. The figure shows the *IoU* on the Validated Set as more Top-k image candidates per command are used during training. Each Top-k batch includes roughly $1,000$ new examples.

approximately 1,000 new training examples, all without manual validation. Results show a consistent increase in the IoU score, which reaches nearly 64.66% on Top-10 (with 10,153 images used for training and 1,240 for validation, over 11,300 images in total), compared to 52.44% for Top-3. This confirms that spatial grounding consistently benefits from increased visual coverage, suggesting that diminishing returns have not yet been reached and that further gains may be possible with even larger sets. A full error analysis is provided in Appendix F. To contextualise these results, we also evaluated Qwen-VL 2.5 in zero-shot. Its performance aligns with that of MiniCPM in the same setting, with F1 scores of 63.81% (Frames), 40.28% (Frame Elements), and average IoU of 6.02%. We additionally tested GPT-4 in zero-shot, but results were inconsistent, likely due to the lack of exposure to FrameNet semantics, an observation consistent with Cheng et al. (2024). Given these factors, and to optimise training resources, we focused fine-tuning efforts exclusively on MiniCPM. Overall, these findings confirm that increasing visual coverage, even with some noise, meaningfully improves multimodal grounding and strengthens the connection between language and perception.

6 Conclusion and Future Work

We presented a novel framework for Grounded Semantic Role Labelling (G-SRL) to interpret robot commands through joint linguistic and visual grounding. Our main contributions are: *i*) a unified semantic representation that links FrameNet roles to perceptual anchors; *ii*) a scalable pipeline for generating and validating multimodal data from symbolic input; and *iii*) empirical evidence that such data notably improves both interpretation and grounding in vision-language models. To the best of our knowledge, no public dataset currently supports the full scope of our G-SRL framework, which integrates frame semantics, visual and symbolic grounding, and perceptual variability. We therefore built such a resource, demonstrating its effectiveness, while leaving adaptation to new domains as future work. Results confirm that training on diverse, ranked visual contexts, even with some noise, substantially enhances grounding. Although our evaluation relies on synthetic images, this controlled setting enabled us to rigorously assess perceptual-semantic alignment at scale, while manual validation ensured that the test set remained synthetic and close to the real-world but free of hallucinations or missing information. Future work will explore how models can autonomously infer visual constraints from language, and how our structured outputs can align with broader semantic formalisms such as AMR (Banarescu et al., 2013). We are also interested in training and evaluation on data from 3D simulation environments such as Holodeck (Yang et al., 2024) or DIALFRED (Gao et al., 2022), though their visual outputs are typically less realistic than diffusion-based images. Ultimately, we aim to enable robots to interpret natural instructions by reasoning about both what they see and what they should expect to see, including multimodal instruction following (e.g., virtual or assistive agents), visual question answering, symbolic planning tasks and evaluating generalisability to real-world, noisy perceptions and actual Human-Robot Interaction scenarios.

Acknowledgments

We acknowledge financial support from the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU and support from Project ECS 0000024 Rome Technopole - CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, funded by the European Union - NextGenerationEU.

Limitations

While our resource provides a rich and coherent mapping between linguistic expressions and visually grounded situations, it currently exhibits some limitations. First, the domain is restricted to domestic environments, reflecting the original scope of the HuRIC dataset. Although this focus ensures coverage depth and contextual realism, it may limit the immediate applicability to other domains such

as outdoor or industrial scenarios. Second, all images are synthetic, synthesised via diffusion models to enable full control and annotation. Nevertheless, human annotators were involved to ensure the perceived realism and plausibility of each scene, especially for evaluation purposes. Moreover, our current diffusion-based image generation approach is scene-agnostic: while we focused specifically on per-image semantic alignment, future work will consider topological consistency across multiple scenes (e.g., coherent views of multiple rooms). Third, our current setup is based on 2D renderings, without incorporating 3D geometry or depth information, which could be relevant in future extensions involving spatial reasoning. Fourth, the selected frame set, while representative of everyday interactions, does not exhaustively cover the full range of FrameNet frames; future work may explore extending the coverage. Lastly, the dataset is grounded in English and reflects a culturally specific domestic setting; multilingual and cross-cultural extensions would be valuable directions to enhance generalizability.

References

Peter Anderson, Qi Wu, Damien Teney, Jeffrey Bruce, Mark Johnson, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2018. Learning interpretable spatial operations in a rich 3d blocks world. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

David Chen and Raymond Mooney. 2011. Learning to interpret natural language navigation instructions from observations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):859–865.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *ECCV*.

Ning Cheng, Zhaohui Yan, Ziming Wang, Zhijie Li, Jiaming Yu, Zilong Zheng, Kewei Tu, Jinan Xu, and Wenjuan Han. 2024. Potential and limitations of llms in capturing structured semantics: A case study on srl. *arXiv preprint arXiv:2405.06410*.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, and 3 others. 2023. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*.

Charles J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S. Sukhatme. 2022. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

Jack Hessel, Jena D. Hwang, Jae Sung Park, Rowan Zellers, Chandra Bhagavatula, Anna Rohrbach, Kate Saenko, and Yejin Choi. 2022. The abduction of sherlock holmes: A dataset for visual abductive reasoning. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, page 558–575, Berlin, Heidelberg. Springer-Verlag.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. 2022. Ai2-thor: An interactive 3d environment for visual ai. *Preprint*, arXiv:1712.05474.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven CH Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICLR*.

Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z. XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, WANG HongFa, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. 2022. Egocentric video-language pretraining. In *Advances in Neural Information Processing Systems*, volume 35, pages 7575–7586. Curran Associates, Inc.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and 1 others. 2023. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Dipendra Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015, Copenhagen, Denmark. Association for Computational Linguistics.

Vishvak Murahari Padmakumar, Nasrin Mostafazadeh, Mohammad Rastegari, and Maya Cakmak. 2022. Teach: Task-driven embodied agents that chat. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. 2023. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. *Preprint*, arXiv:2307.05463.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Wenlong Han, Roozbeh Mottaghi, and Dieter Fox. 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)*.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25(1):1507–1514.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 23–30. IEEE Press.

Andrea Vanzo, Emanuele Bastianelli, Roberto Basili, and Daniele Nardi. 2020. Grounded language interpretation of robotic commands through structured learning. *Artif. Intell.*, 278.

Linhui Xiao, Xiaoshan Yang, Xiangyuan Lan, Yaowei Wang, and Changsheng Xu. 2024. Towards visual grounding: A survey. *Preprint*, arXiv:2412.20206.

Shaohua Yang, Qiaozi Gao, Changsong Liu, Caiming Xiong, Song-Chun Zhu, and Joyce Y. Chai. 2016. Grounded semantic role labeling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 149–159, San Diego, California. Association for Computational Linguistics.

Xuzheng Yang, Junzhuo Liu, Peng Wang, Guoqing Wang, Yang Yang, and Heng Tao Shen. 2025. New dataset and methods for fine-grained compositional referring expression comprehension via specialist-mllm collaboration. *Preprint*, arXiv:2502.20104.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, and 1 others. 2024. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

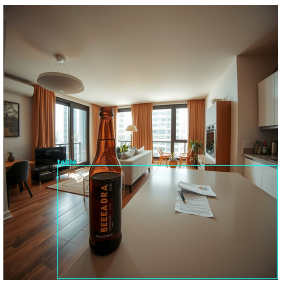
Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Differentiable parsing and visual grounding of natural language instructions for object placement. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11546–11553.

A Appendix: Grounded Semantic Representation Examples

This Section shows different examples for grounded semantic role labelling applied to natural language commands. Each command ($\mathcal{C}$) is mapped to one or more FRAMES with role (Frame Elements) grounded either *visually* (bounding boxes) or *symbolically* (e.g., <ROBOT>, <ROOM>). For each example, we present the natural language command ($\mathcal{C}$), the constraints ($C(\mathcal{C})$) derived from the environment as described in Section 4 and its corresponding interpretation.

(a). The image for Example 1.

(b). The image for Example 2.



(c). The image for Example 3.

Example 1 – Navigating to a room.
Command: $\mathcal{C}$ = *"could you go to the kitchen please"*
Constraints: $A = \{\neg\texttt{visible(kitchen)}\}$ $S = \emptyset$ $O = \emptyset$.
The robot must move to a known location. The GOAL is grounded symbolically via its internal map.

```
MOTION(
  (AGENT, "you", <ROBOT>),                    // symbolic grounding
  (GOAL, "kitchen", <ROOM>)                   // symbolic grounding
)
```

Example 2 – Delivering an object.
Command: $\mathcal{C}$ = *"bring me the bottle"*
Constraints: $A = \{\texttt{visible(bottle)}\}$ $S = \emptyset$ $O = \emptyset$.
The robot is asked to bring the visible object in Figure 4b to a person.

```
BRINGING(
  (THEME, "bottle", [402, 171, 576, 821]]),   // visual grounding
  (BENEFICIARY, "me", <PERSON>)               // symbolic grounding
)
```

Example 3 – Retrieving an object not currently visible.
Command: $\mathcal{C}$ = *"take the wallet from the table"*
Constraints: $A = \{\neg\texttt{visible(wallet)}, \texttt{visible(table)}\}$ $S = \emptyset$ $O = \emptyset$.
The wallet is not in the current scene; the table is grounded visually.

```
TAKING(
  (THEME, "wallet", <MISSING>),
  (SOURCE, "table", [201, 597, 1021, 1020])
)
```

Example 4 – Turning on a device.
Command: $\mathcal{C}$ = *"turn on the tv"*
Constraints: $A = \{\texttt{visible(tv)}\}$ $S = \emptyset$ $O = \{\texttt{on(tv)}\}$.
The device is visible; the state is symbolically grounded.

```
CHANGE_OPERATIONAL_STATE(
  (DEVICE, "tv", [313, 348, 854, 652]),
  (OPERATIONAL_STATE, "on", <STATUS>)
)
```

(a). The image for Example 4.

(b). The image for Example 5.

(c). The image for Example 6.

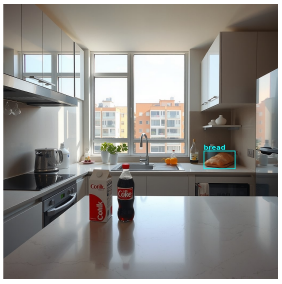
Example 5 – Placing an object in a vague location.
Command: $\mathcal{C}$ =*"put the keys there"*
Constraints: $A = \{$`visible(keys), visible(table)`$\}$ $S = \emptyset$ $O = \emptyset$.
The destination is deictic ("there") and symbolically marked.

PLACING(
(THEME, "*keys*", [783, 884, 947, 994]),
(GOAL, "*there*", `<POSITION>`)
)

Example 6 - Handling pronominal references.
Command: $\mathcal{C}$ =*"pick it up"*
Constraints: $A = \{$`visible(phone), visible(apples)`$\}$ $S = \{$`near(phone, apples)`$\}$ $O = \emptyset$.
The object "*it*" (referring to the phone) is underspecified and grounded symbolically.

TAKING(
(THEME, "*it*", `<ITEM>`)
)

(a). The image for Example 7.



(b). The image for Example 8.

(c). The image for Example 9.

Example 7 – Switching off a device.
Command: $\mathcal{C}$ =*"switch off the lamp"*
Constraints: $A = \{$`visible(lamp), visible(sidetable)`$\}$ $S = \{$`ontop(lamp, sidetable)`$\}$
$O = \{$`on(lamp)`$\}$.
The device is grounded visually; the state "*off*" is symbolic.

CHANGE_OPERATIONAL_STATE(
(DEVICE, "*lamp*", [385, 139, 714, 804]),
(OPERATIONAL_STATE, "*off*", `<STATUS>`)
)

Example 8 – Composed command with two frames.
Command: $\mathcal{C}$ =*"can you go to the kitchen and bring me some bread"*
Constraints: $A = \{$`visible(kitchen), visible(countertop), visible(bread)`$\}$
$S = \{$`ontop(bread, countertop)`$\}$ $O = \emptyset$.
The instruction involves a sequential action: navigating to a room and then bringing an object to the speaker.

MOTION(
(AGENT, "*you*", <ROBOT>),
(GOAL, "*kitchen*", <ROOM>)
)

BRINGING(
(BENEFICIARY, "*me*", <PERSON>),
(THEME, "*bread*", [746, 547, 864, 615])
)

Example 9 – Multiple bounding boxes required.
Command: $\mathcal{C}$ ="*bring the box near the closet of the bedroom"*
Constraints: $A = \{\texttt{visible(bedroom)}, \texttt{visible(box)}, \texttt{visible(closet)}\}$
$S = \{\texttt{far(box, closet)}\}$ $O = \{\texttt{closed(closet)}\}$.
The instruction requires moving an object near another, both of which must be grounded in the image.

BRINGING(
(THEME, "*box*", [415, 694, 608, 863]),
(GOAL, "*closet*", [445, 481, 607, 636])
)

B Appendix: Metaprompt Examples for Controlled Image Generation

This Section presents two examples of metaprompts used to generate natural language descriptions that serve as inputs to a diffusion-based image synthesis model (`FLUX.1-schnell`). These prompts are crucial for generating grounded visual scenes aligned with robot commands, particularly for training vision-language models to perform semantic grounding.

The first metaprompt enforces the exclusion of specific entities, producing images where target objects such as `Person`, `Robot`, or `Cup` are absent. This is useful for generating *negative samples*: examples in which grounding should fail or where references are visually unsupported. The second metaprompt, in contrast, ensures the inclusion of a specified entity, thereby supporting the generation of *positive samples* where visual grounding is explicitly correct and interpretable. Both strategies are essential for learning robust multimodal grounding functions.

Each metaprompt generates five diverse image descriptions, each corresponding to a different visual perspective (close-up, wide-shot, long-shot, low-angle, and high-angle). This variation helps the model generalise across different spatial framings and enhances its robustness to visual composition changes.

Metaprompt: Entity Exclusion

This prompt is designed to synthesise scenes in which specific entities must *not* appear. It was used to generate images that challenge the grounding model by omitting objects referenced in the command. Such examples are important for ensuring that the model does not hallucinate entity bindings when visual evidence is absent.

> You are a prompt engineer and an expert in computer vision.
>
> Your task is to generate five detailed prompts that will be used to create images for a grounding detection model based on the FLUX.1-schnell model. These images will be paired with robot commands to support grounded interpretation in real-world environments.
>
> **Input Description:**
>
> You will receive:
>
> - Important ENTITIES to exclude: Objects that must **never** appear in any prompt.
> - Optional ENTITIES: Objects that may appear in some prompts but not all.
> - Spatial Relations: Positional relationships between important entities (if any).
> - Entity Location: The environment in which entities should be situated.
> - Frame Semantics: A semantic representation of a robot command, describing the intended action.
>
> **Prompt Requirements:**

- All five prompts must **omit** the excluded entities.
- Describe the scene from five distinct visual perspectives:
  - Close-up
  - Wide-shot
  - Long-shot
  - Low-angle
  - High-angle
- Use a domestic indoor setting.
- Vary optional entities across prompts.
- Maintain realism and contextual plausibility.

Example Input:

- Entities to exclude: Person, Robot, Cup
- Optional entities: Plate, Cabinet, Pasta, Can, Bag
- Location: Apartment
- Command: "`please carry the mug to the bathroom`"
- Frame Semantics:
  - Frame: `Bringing`
  - Theme: `Cup`
  - Goal: `Bathroom`

The following output was generated based on this metaprompt:

[
"A wide shot of a modern kitchen interior with a plate and a cabinet visible. The focus is on an empty table, and there are no cups, robots, or people present.",
"A close-up shot of a dining room table with a can of pasta and a paper bag. The image is cropped tightly to exclude any view of a cup or person.",
"A high-angle shot of a bathroom entrance showing the floor and a cabinet. A plate is barely visible in the hallway, emphasising the path to the bathroom.",
"A long shot from the living room toward the kitchen, with no cup, person, or robot visible. A bag and a can are placed near the door.",
"A low-angle shot from the corridor looking up at a cabinet near the bathroom door. Optional objects like pasta are included, but all excluded entities are absent."
]

These prompts help ensure that the resulting images do not contain visual evidence for entities that should not be grounded. This discourages over-reliance on textual priors during multimodal training and supports more reliable detection of false positives.

Metaprompt: Entity Inclusion

This second metaprompt is used to enforce the inclusion of a specific visual entity in every generated image. In this case, the target object (e.g., `Cup`) must be clearly visible and semantically consistent with the frame semantics interpretation of the command. Such prompts are vital for generating positive training instances where grounded references are unambiguous and visually present.

You are a prompt engineer and an expert in computer vision.

Your task is to generate five detailed prompts that will be used to create images for a grounding detection model based on the FLUX.1-schnell model. These images will be paired with robot commands to support grounded interpretation in real-world environments.

Input Description:

You will receive:

- Important ENTITIES to include: Objects that must appear in every prompt.
- Entities to exclude: Objects that must **not** appear.
- Optional ENTITIES: Objects that may or may not appear.
- Spatial Relations: Positional relationships to enforce (if any).
- Entity Location: The setting of the scene (e.g., apartment).
- Frame Semantics: A semantic representation of the command.

Prompt Requirements:

- All five prompts must include the target entity.

- Describe the scene from five distinct visual perspectives:
  - Close-up
  - Wide-shot
  - Long-shot
  - Low-angle
  - High-angle
- The target object must be clearly visible and not occluded.
- Use a realistic domestic setting.
- Vary optional entities across prompts.

Example Input:

- Entities to include: Cup
- Entities to exclude: Person, Robot
- Optional entities: Plate, Cabinet, Pasta, Can, Bag
- Location: Apartment
- Command: "`please carry the mug to the bathroom`"
- Frame Semantics:
  - Frame: `Bringing`
  - Theme: `Cup`
  - Goal: `Bathroom`

The LLM-generated output for this configuration was:

[
"A close-up of a ceramic cup resting on a wooden countertop in a kitchen, with a bag of pasta in the background. The cup is clearly visible.",
"A wide-angle shot of an apartment living area showing a cup placed centrally on a table. Other optional objects like a cabinet are visible.",
"A long shot of a corridor leading to the bathroom, with a cup on a tray carried by an unseen agent. No person or robot is visible.",
"A low-angle shot of a shelf where a cup is positioned, taken from the floor perspective. Optional cans and bags are out of focus in the corner.",
"A high-angle shot of a bathroom entrance with a cup on the floor near the door. The cup is the primary object of focus, with no occlusion."
]

C Appendix: Question Templates for Constraints Validation

In this Appendix Section, we report the templates used during the automatic Constraints Validation process with GroundingDINO and MiniCPM for computing the quality scores of an image. These templates asses the $C(\mathcal{C}) = A \cup S \cup O$ set of constraints from Section 4:

- A, i.e., Accessible Entities, constraints are validated through the invocation of GroundingDINO over an image with a simple request "*a* `entity_name`". For instance, for Figure 1 we would ask for "*a phone*" or "*a bed*", to exploit its Object Detection capability from short sentences.

- S, i.e., the Spatial Relations, constraints are checked using MiniCPM over an image with a natural language question "*Is the* `entity_name_1` *{far from/close to/on top of}* `entity_name_2`*? Answer only yes or no.*" in order to exploit its Question Answering capability as a Visual LLM. As an example, for Figure 1, a question would be "*Is the phone on the bed? Answer only yes or no.*".

- O, i.e., the Object State Properties, constraints are validated using MiniCPM again over an image with a natural language question "*Is the* `entity_name` *{closed/open/on/off}? Answer only yes or no.*" in order to exploit its Question Answering capability as a Visual LLM. For example, for Figure 1, a question could be "*Is the phone on? Answer only yes or no.*".

D Appendix: Annotation Guidelines for Image Validation

This appendix details the protocol followed for manually validating the automatically generated dataset. Each image was assessed independently by two annotators along five distinct dimensions covering visual realism, object plausibility, grounding accuracy, and semantic consistency. The goal of the annotation was to identify and exclude images exhibiting any form of visual or semantic failure, thereby constructing a high-precision evaluation set. The following definitions clarify the criteria applied to each validation dimension and correspond directly to the error categories reported in Table 1.

Annotation task. Each generated image was manually evaluated along five dimensions to assess its semantic and visual correctness. Two annotators performed the validation process using a structured interface. Each dimension is linked to a specific error category reported in Table 1. An image was retained in the Validated Dataset only if it exhibited no errors in any category.

Malformed. Flags visually broken or stylistically inconsistent images. An image was marked as malformed if it appeared corrupted, incoherent, or rendered in a non-photorealistic style (e.g., cartoon, sketch).

Anomalous Elements. Captures physically implausible or semantically inconsistent elements, such as floating objects, disembodied limbs, or presence of irrelevant figures (e.g., robots or humanoids without justification).

Bounding Box Errors (BBox). Evaluates whether all expected entities listed as "must be visible" are correctly localised with bounding boxes. Errors include missing boxes, inaccurate placements, or false positives for entities that should be absent. Errors are reported both as absolute percentages (over the full dataset) and relative percentages (restricted to cases where bounding boxes are required).

State Errors. Applies when the command specifies expected object states (e.g., "oven should be open"). Errors are marked if the visual depiction contradicts the expected state. Only images with explicit state constraints were considered for this check.

Spatial Errors. Assesses whether spatial relationships (e.g., "on top of", "near to", "inside") are correctly realised. A relation was considered violated if the visual arrangement contradicted the specified constraint. "Far from" conditions are considered satisfied if one of the entities is absent, as this implies distance.

Notes. Annotators selected the most appropriate category via dropdown fields. When uncertain, comments were recorded for review. All errors were treated as independent, and no partial credit was assigned: a single violation in any category led to exclusion from the Validated Dataset.

E Appendix: Training Hyper-parameters

All models were trained using **2 NVIDIA A100 80GB GPUs** in parallel with **bfloat16 (bf16)** precision. We adopted the DeepSpeed framework with **ZeRO Stage 3** for memory and compute optimisation.

Training Strategy. All model components were fine-tuned end-to-end. The optimal training configuration was selected based on performance on the development set. Unless otherwise specified, the following hyperparameters were used:

- **Learning rate:** 1e-6
- **Batch size:** 1
- **Gradient accumulation steps:** 1
- **Epochs:** 3

MiniCPM-V Specifics. MiniCPM was fine-tuned using the AdamW optimiser with default parameters. The model was trained using full-resolution images without pixel downscaling. The best results were achieved with:

- `weight_decay` = 0.1
- `adam_beta2` = 0.95
- `warmup_ratio` = 0.01

Optimisation and Checkpoints. Training progress was monitored on the development set using the Cross-Entropy Loss as the main criterion for early stopping and hyperparameter selection. All runs used the default AdamW settings and did not require any form of gradient clipping or loss stabilisation beyond the specified `max_grad_norm`. Checkpoints were saved at regular intervals, and the best model according to development performance was used for final evaluation.

Training and Inference Prompt. The following structured prompt is used for both models during Training and Inference.

> You are given a natural language command and a corresponding image depicting a domestic environment. Your task is to convert the command into a **structured, grounded semantic representation** in the form of **predicate-argument structures**.
>
> This means you must:
>
> - Identify the semantic **frames** expressed in the command (e.g., MOTION, TAKING, BRINGING, LOCATING, PLACING, CHANGE_OPERATIONAL_STATE, etc.)
> - For each frame, extract a list of **frame elements**, grounded visually whenever possible.
>
> Your output must be a **list of dictionaries**, one per frame. Each dictionary contains:
>
> - "frame": the name of the frame (string), following FrameNet conventions.
> - "elements": a list of dictionaries, each with:
>   - "name": the role of the frame element (e.g., Theme, Goal, Agent, Beneficiary, etc.)
>   - "surface": the exact text span from the command. It must be the semantic head, i.e., the word referring to the object.
>   - "bbox_2d", either:
>   - a list of four integers [x1, y1, x2, y2] representing a visible object's bounding box in the image
>   - or a symbolic tag among <ROBOT>, <PERSON>, <ROOM>, <POSITION>, <STATUS>, <MISSING> and <ITEM>
>
> **Important guidelines:**
>
> - Only annotate what is **visually present** in the image and is relevant for the command (as a frame element).
> - Use <MISSING> if the referred object **cannot be seen**.
> - Use <ROBOT> for the robot, and <PERSON> for a human addressee.
> - Use <POSITION> when the target is an abstract spatial reference like "there" or "here".
> - Use <STATUS> for expressions of operational state (e.g., "on", "off").
> - Use <ITEM> for references to unspecified or deictic objects (e.g., pronouns like "this", "that", or "it" when referring to a visible item).
> - All tags must be typed **exactly as given**, with no added characters or typos.
> - Always extract **surface** spans literally from the input command.
> - Your output must be in **valid JSON**, and formatted as **a single line**.
>
> —
>
> **Examples:**
>
> <IMAGE>
> **Command:** *could you go to the kitchen please*
>
> **Output:**
>
> ```
> [{'frame': 'MOTION', 'elements': [{'name': 'Agent', 'surface': 'you', 'bbox_2d': '<ROBOT>'},
> {'name': 'Goal', 'surface': 'kitchen', 'bbox_2d': '<ROOM>'}]}]
> ```
>
> —
>
> <IMAGE>
> **Command:** *bring me the bottle*
>
> **Output:**
>
> ```
> [{'frame': 'BRINGING', 'elements': [{'name': 'Agent', 'surface': 'you', 'bbox_2d': '<ROBOT>'},
> {'name': 'Theme', 'surface': 'bottle', 'bbox_2d': [210, 612, 780, 830]},
> {'name': 'Beneficiary', 'surface': 'me', 'bbox_2d': '<PERSON>'}]}]
> ```

—

<IMAGE>
Command: *take the wallet from the table*

Output:

```
[{'frame': 'TAKING', 'elements': [{'name': 'Agent', 'surface': 'you', 'bbox_2d': '<ROBOT>'},
{'name': 'Theme', 'surface': 'wallet', 'bbox_2d': '<MISSING>'},
{'name': 'Source', 'surface': 'table', 'bbox_2d': [480, 660, 960, 900]}]}]
```

—

<IMAGE>
Command: *turn on the tv*

Output:

```
[{'frame': 'CHANGE_OPERATIONAL_STATE', 'elements': [{'name': 'Agent', 'surface': 'you',
'bbox_2d': '<ROBOT>'}, {'name': 'Operational_state', 'surface': 'on', 'bbox_2d': '<STATUS>'},
{'name': 'Device', 'surface': 'tv', 'bbox_2d': [251, 337, 832, 708]}]}]
```

—

Now do the same for the following example:
<IMAGE>
Command: <INPUT_COMMAND>

Notice: For illustrative purposes, throughout the main text and appendix, the model outputs are displayed in a bracketed format. However, in the actual prompts, the expected output is in standard JSON format, which is also the format natively generated by the trained models.

F Appendix: Error Analysis

In this section, we provide an error analysis of the trained models on our multimodal dataset for the Grounded Semantic Role Labelling task. Each input consists of an image, a textual system prompt (Appendix E), and the natural language command. The expected output is a well-structured JSON object, as defined in Appendix E. We categorise the observed errors into three main types, illustrated below with representative examples:

- **Semantic Misinterpretation**: The model misrepresents the intended meaning of the input command. For instance, it may classify an action as `Bringing` (involving both a *Source* and a *Goal*) instead of the correct `Taking` frame, where the action involves only picking up an object. These errors reflect a failure to capture the semantics of the command.

- **Structural Inconsistency**: The model produces an output that deviates from the expected frame structure. This includes introducing superfluous frames or frame elements, or omitting mandatory ones. For example, in a command involving a sequential action (e.g., moving and then interacting), the model might omit the `Motion` frame or wrongly insert it when it is not required. These issues are not about semantics but about conformance to the expected structure.

- **Grounding Errors**: The model fails to correctly associate frame elements with visual objects in the image. It may hallucinate an object and provide a bounding box for an entity that is not present, or omit a required bounding box entirely. These errors point to failures in visual grounding and perceptual alignment.

Additionally, a fourth hypothetical category could be defined as **Invalid Output**, referring to completely malformed or irrelevant generations outside the frame semantics domain. However, such cases did not occur in our experiments, likely due to the strong language modelling capabilities of the underlying LLMs.

Semantic Misinterpretation. This is the rarest category of error and was observed in a single case involving the command: "*go to the table and take the fork near the microwave on the shelf*", paired with the image in Figure 7. The correct interpretation requires a `Motion` action towards the *table*, followed by a `Bringing` action involving the *fork* to a place near the *microwave*. Due to a degree of linguistic ambiguity, the model incorrectly interprets the command as a simple `Taking` action, and produces an incomplete grounding:

MOTION(
  (GOAL, "*table*", [2, 498, 1021, 1019]),
)

TAKING(
  (THEME, "*fork*", <MISSING>)
)

Interestingly, the model trained on the Top-5 dataset (with broader image exposure) is able to produce a bounding box for the *fork*, albeit still within the incorrect `Taking` frame. This suggests that increased visual variety enhances grounding capabilities, even when semantic disambiguation remains an issue.

Figure 7: The image associated with the command "*go to the table and take the fork near the microwave on the shelf*", where the models misinterpret the command with a `Taking` action instead of `Bringing`.

Structural Inconsistency. This type of error is infrequent, as most models learn to respect the structure of Semantic Role Labelling and the correct associations between Frames and Frame Elements. However, some models occasionally introduce unnecessary Frames. Consider the command "*take the bottle of water on the table*" and the corresponding image in Figure 8. The model trained on the Top-1 dataset produces the following output:

TAKING(
  (THEME, "*table*", [210, 612, 780, 830]),
)

BEING_LOCATING(
  (THEME, "*bottle*", <MISSING>),
  (LOCATION, "*table*", [480, 660, 960, 900])
)

Although this output is not semantically incorrect, the inclusion of the `Being_Located` frame is unnecessary. The added frame provides further detail by reiterating that the *bottle* is located on the *table*, but this level of elaboration was not required. Hence, the output may be considered redundant rather than erroneous. Models trained on the Top-2 and Top-3 datasets correctly omit the additional frame, demonstrating that broader visual coverage helps reinforce structural precision too.

Figure 8: The image associated with the command "*take the bottle of water on the table*", where the models introduce a Frame in the interpretation.

Grounding Errors. These errors form almost 27% of the cases and occur when the model fails to associate the relevant frame elements with corresponding visual entities in the image. Some of them are due to the fact that the entities are blurred or are in the back of the represented environment. For instance, in the command "*get me my catalogue near the bed*" for the image in Figure 9, the expected interpretation is:

BRINGING(
  (BENEFICIARY, "*me*", <PERSON>),
  (THEME, "*catalogue*", [492, 624, 644, 684])
  (SOURCE, "*bed*", [653, 332, 1022, 1018])
)

While all the models succeed in generating the correct frame structure and surface forms, they consistently fail to ground either the *catalogue* or the *bed*. One explanation could be the fact that *the catalogues* are in the back and are blurred. This points to limitations in visual alignment rather than linguistic interpretation.

A similar failure is observed for the command "*take the wine bottles in the kitchen*" for Figure 10. Here, the expected output is:

TAKING(
  (THEME, "*bottles*", [419, 227, 583, 910])
)

Despite the clear and prominent presence of the *bottle* in the foreground of the image, none of the models is able to generate the correct bounding box. The object is fully visible, centrally located, and unoccluded, making this failure particularly unexpected. One reason could be due to the plural usage of the entity name (*the wine bottle**s*** and the fact that there are several other bottles in the back of the image, so the model may be unsure about which bottle.

Figure 9: The image associated with the command "*get me my catalogue near the bed*", where the models fail to ground the mentioned entities.

Figure 10: The image associated with the command "*take the wine bottles in the kitchen*", where the models fail to ground the wine bottle.

Finally, a common instance of Grounding Errors (the most frequent error type) occurs when the model misgenerates the bounding box coordinates, producing a box that does not match the Gold Standard and thus fails to fully capture the target entity. Consider the command "*switch on the tv*", associated with the image in Figure 11a. In Figure 11b, we show the bounding boxes overlaid in post-processing for the entity TV, which must be correctly grounded in the scene. The Gold Standard bounding box is shown in cyan, while the outputs from the models trained on different subsets are colour-coded as follows: red (Top-1), orange (Top-2), purple (Top-3), and blue (Top-5). A clear trend emerges: as the training dataset becomes more diverse, the predicted bounding boxes become more accurate. The best-performing model (Top-5) achieves an Intersection over Union (IoU) score of 94% for the TV, well above the threshold for accurate

localisation. In summary, grounding errors, especially in visually ambiguous or underspecified contexts, remain the main source of mistakes, while semantic and structural errors are infrequent. Crucially, our results show that expanding visual diversity in training data leads to consistent reductions in both error types, highlighting the importance of large-scale, diverse synthetic datasets for robust multimodal semantic interpretation.

(a). The original image associated with the command "*switch on the tv*", where you can see the tv in the top left corner.

(b). The same image where we added the Gold bounding boxes (in cyan) and the bounding boxes predicted by the models (other colours).

G Appendix: Frames and Frame Elements Distribution

This section briefly presents some statistics in Table 3 about the distribution of the elements in the interpretation of the commands. The dataset exhibits a rich and diverse distribution of semantic frames and frame elements. Among the most frequent frames are BRINGING (328 occurrences), MOTION (260), TAKING (195), and LOCATING (189), all of which are central to spatial and manipulation tasks. Each frame is associated with a variable set of frame elements: for instance, BRINGING includes up to seven distinct roles such as *Theme* (328), *Goal* (200), and *Beneficiary* (123), while MOTION prominently features *Goal* (239) and *Theme* (39). Less frequent but semantically precise frames, such as MANIPULATION, CLOSURE, and ATTACHING, maintain a smaller yet well-defined set of elements. This distribution reflects the linguistic complexity of grounded robotic commands and supports the design of models capable of fine-grained semantic parsing across a broad range of contexts.

Frame (Total)	Frame Element	Frequency	Coverage (%)
ARRIVING (22)	Goal	22	100.0%
	Manner	2	9.1%
	Path	11	50.0%
ATTACHING (20)	Goal	20	100.0%
	Item	12	60.0%
BEING_IN_CATEGORY (21)	Category	21	100.0%
	Item	21	100.0%
BEING_LOCATED (84)	Location	77	91.7%
	Place	2	2.4%
	Theme	86	102.4%
BRINGING (328)	Agent	74	22.6%
	Area	2	0.6%
	Beneficiary	123	37.5%
	Goal	200	61.0%
	Manner	2	0.6%
	Source	48	14.6%
	Theme	328	100.0%
CHANGE_DIRECTION (11)	Angle	3	27.3%
	Direction	11	100.0%
	Speed	1	9.1%
	Theme	1	9.1%
CHANGE_OPERATIONAL_STATE (104)	Agent	33	31.7%
	Device	104	100.0%
	Operational_state	91	87.5%
CLOSURE (39)	Agent	13	33.3%
	Container_portal	16	41.0%
	Containing_object	23	59.0%
	Degree	4	10.3%
COTHEME (4)	Cotheme	4	100.0%
	Manner	1	25.0%
GIVING (22)	Donor	9	40.9%
	Reason	3	13.6%
	Recipient	22	100.0%
	Theme	22	100.0%
INSPECTING (55)	Desired_state	18	32.7%
	Ground	53	96.4%
	Inspector	10	18.2%
	Unwanted_entity	4	7.3%
LOCATING (189)	Cognizer	7	3.7%
	Ground	74	39.2%
	Manner	4	2.1%
	Perceiver	15	7.9%
	Sought_entity	189	100.0%
MANIPULATION (12)	Entity	12	100.0%
MOTION (260)	Area	3	1.2%
	Direction	13	5.0%
	Distance	1	0.4%
	Goal	239	91.9%
	Manner	5	1.9%
	Path	18	6.9%
	Source	2	0.8%
	Theme	39	15.0%
PERCEPTION_ACTIVE (12)	Phenomenon	12	100.0%
PLACING (107)	Agent	15	14.0%
	Area	2	1.9%
	Goal	105	98.1%
	Theme	107	100.0%
RELEASING (19)	Goal	11	57.9%
	Theme	19	100.0%
TAKING (195)	Agent	18	9.2%
	Source	43	22.1%
	Theme	195	100.0%

Table 3: Detailed distribution of Frame Elements per Frame. The frequency column indicates how often each element appears in association with a given frame; the percentage indicates its relative frequency with respect to the total frame occurrences.