Hierarchical Semantic Frames for Grounding Language in Robot Control Primitives

Emily Sheetz[1,2], Matthew Shannon[1], Cameron Kisailus[1], Adam Ingerman[1], Shaun Azimi[2]

***Abstract*— As robots become increasingly present in human environments, we need robots to be intuitively commanded by and effectively communicate with humans. In particular, non-expert users should be able to communicate task goals with robots. Language emerges as a logical mode of interaction due to its ubiquity in human environments and, more importantly, as the way humans naturally express tasks. Natural language commands present challenges in that robots must reason over ambiguous language probabilistically and reason over commands they may not be able to execute. We present *hierarchical semantic frames*, which ground commands in robot control primitives through *hierarchies* that construct high-level commands from lower-level commands. We demonstrate that *hierarchical semantic frames* allow robots to understand and execute a variety of commands, such as those involving multiple verb meanings, command variations, and compound nouns. The robot quickly processes *hierarchical semantic frames* and accurately grounds and executes the commanded tasks, demonstrating the power of *hierarchical semantic frames* for allowing users to intuitively interact with robots.**

I. INTRODUCTION

Enabling intuitive communication between human users and autonomous robots is a crucial capability to enable seamless collaboration with robot assistants. Previous work has investigated intuitive ways to interact with and program robots, such as gestures or facial expressions [6], [54], eye-tracking [4], [36], and learning from demonstration [3], [27], [34], [35]. We see language as an intuitive interface that can provide a wide variety of rich input signals for commanding robots [52], especially for non-expert users who do not have programming or robotics experience. A *scalable restricted language* will allow robots to unambiguously understand and execute commands from non-expert users. Our previous work, RoboFrameNet [53], demonstrates the power of semantic frames [55] to bridge the gap between language and goal-directed robot actions. However, RoboFrameNet does not scale well to new actions and focuses more on frame instantiation than command execution.

Scaling robot systems to understand all possible mappings of natural language commands to actions remains challenging due to the ambiguity of language. For example, natural language commands present challenges such as difficulty differentiating verbs with multiple meanings ("set the table" or "set down the object"), grounding command variations (such as "go to Alice's desk" or "go to Charlie's desk"), or recognizing compound nouns (such as "Alice's desk" or "left hand"). Systems such as Amazon's Alexa and Apple's Siri indicate that restricted language could be a more scalable solution to providing an intuitive interface for commanding robots [43], [11], [28]. Even when using restricted language, previous implementations require distinct semantic frames to represent small changes in commands (for example, "pick up the red cup" and "pick up the blue cup"), which is not scalable for new actions or objects. We posit that adding flexibility to semantic frames will address the challenge of creating a scalable restricted language for commanding robots.

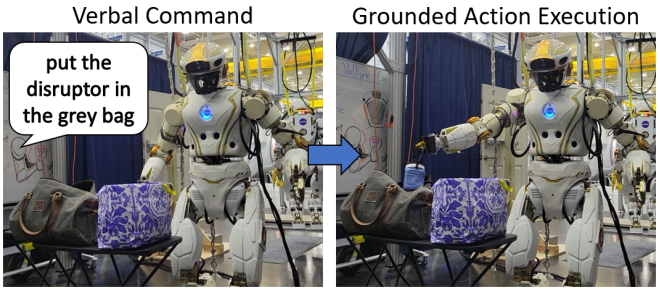


Fig. 1: *Hierarchical semantic frames* provide a scalable restricted language that ground commands in robot action.

We take insight from hierarchical robot control schemes and the hierarchical structure of language to address the challenges of creating a scalable restricted language for robot commands. In robot control, notions of hierarchy and composition allow simple primitives to be combined to create more complex behaviors [2], [18], [41]. Combining these simple primitives in different ways offers much flexibility and variation in the resulting behaviors. Similarly, we argue that by building high-level commands from lower-level semantic frames grounded in action, robots can scalably recognize and execute complex commands.

In this paper, we propose *hierarchical semantic frames* (HSFs) as a method for: (1) grounded task execution through hierarchies of a limited number of low-level manipulation policies, and (2) creating a restricted language for commanding robots to perform tasks that is scalable in the number of semantic frames needed to express actions. HSFs allow robots to understand commands without the need to instantiate distinct frames for each command variation. HSFs also guarantee frame actions grounded in robot control primitives, effectively mapping language to robot action. We test our approach on two robot platforms and multiple tasks

The authors are with the [1]University of Michigan and [2]NASA Johnson Space Center (NASA).

and compare the scalability of HSFs to RoboFrameNet. Our experimental results demonstrate that our proposed HSFs can be used for successful task execution on a real robot, and effectively ground a variety of commands in robot action. Our proposed *hierarchical semantic frames* allow robots to recognize and execute commands, abstracting the details of action execution and allowing robots to interact with non-expert users through language.

II. RELATED WORK

A. Language as a Robot Percept

Researchers have long investigated different ways for users to intuitively interact with robots, due to the rich input signals different modalities can provide. In particular, researchers have explored how to use language to intuitively communicate task goals to robots. In the 1970s, the system SHRDLU [57], [58] was developed, which carried out natural language commands in a virtual environment. Since then, researchers have aimed to expand the use of natural language to command intelligent agents and robots [52].

Many works demonstrate the power of using natural language to command robots. Dzifcak et al. [13] explore how to translate natural language instructions into descriptions of task goals and actions. Chernova et al. [9] use data-mining for robots to ground action-oriented natural language. Tellex et al. [51] present Generalized Grounding Graphs for probabilistically inferring the sequence of actions required to execute a command. Matuszek et al. [32] investigate how robots can learn what objects are being referred to in deictic gestures and language (gestures and language that draw attention to objects without naming them directly). Several works explore how robots can ground abstract spatial concepts (such as relative relationships between objects) to execute natural language commands [37], [46], [50]. Many works explore understanding natural language in route navigation tasks [29], [26], [30], [31], including commands involving verbs that imply motion [25], commands that imply navigation constraints [21], and commands that imply environment information [19], [12]. Google's SayCan [1] combines a large language model with affordance knowledge to allow robots to reason over natural language in long-horizon tasks. These works demonstrate widespread interest in using natural language to command robots and the challenges of grounding natural language in robot understanding. However, scaling to new commands or domains and reliably grounding ambiguous natural language in robot action remains an open question.

Due to the challenges of scaling to natural language, commanding robots using restricted language is a useful approach. Voice interfaces—such as Amazon's Alexa, Apple's Siri, Google's Assistant, and Microsoft's Cortana [43], [11], [28]—are part of everyday life. These systems demonstrate the power of restricted language for commanding intelligent agents. Some research indicates that restricted language allows users to achieve similar or better task performance than natural (unrestricted) language without detracting from overall user experience [33]. These works demonstrate the power of using restricted language to communicate task goals intuitively to robots.

B. Semantic Frames

Semantic frames are used in natural language processing (NLP) to represent a scene being acted out [48], [55], [56], [20]. FrameNet [48] emphasizes that a verb alone is not sufficient to describe a scene or action, and *frame elements* are necessary to describe agents and direct and indirect objects involved in the action. For example, the verb "give" cannot be acted out until we know what object is being given and to whom. FrameNet uses hand-annotated *lexical units* to map language into the appropriate semantic frame by expressing how frame elements relate to a command.

RoboFrameNet [53] extends FrameNet [48] and uses semantic frames as a middle-ground between spoken commands and robot action. RoboFrameNet interprets spoken commands as text, then parses the text to instantiate a semantic frame. Representations of object affordances for robotics generally do not explicitly note the direct and indirect objects being acted on, which limits the complexity of robot action that can be performed [59]. In contrast, semantic frames augment robot understanding of the action being performed by describing the objects being acted on.

RoboFrameNet demonstrates the power of semantic frames in allowing robots to comprehend spoken commands. We extend RoboFrameNet by advancing the capabilities and scalability of semantic frames, so that fewer frames are required to express actions. Rather than focusing on semantic frame instantiation, we place greater emphasis on the execution of the actions represented by semantic frames.

C. Hierarchical Robot Control

Many works control robots using hierarchical control [2] or subsumption architectures [18]. Robots can execute object affordances [14] using a control basis of object-centric [5] controllers. A control basis builds up complex actions from simple behavioral building blocks such as grasping [41], [40], [42] or conditioning behaviors [16] such as avoiding joint limits and singularities. Executing complex tasks requires composition of the low-level building blocks [47] and sequencing these behaviors [7] to achieve a task goal.

We take inspiration from the *hierarchies* of manipulation policies seen in robot control to ground our *hierarchical semantic frames* in robot action. Similar to how complex robot actions are comprised of simple, low-level behaviors, our *hierarchical semantic frames* are constructed from simple, low-level grounded commands. Hierarchies allow our pipeline to scalably ground high-level commands.

III. METHODS

A. Problem Formulation

A command Λ in a restricted language Σ_{R} is a sequence of words $\lambda_1, \ldots, \lambda_N$. Given a command $\Lambda \in \Sigma_{\mathrm{R}}$ and a set of robot control primitives Φ, the robot needs to determine the sequence of grounded robot control primitives $\phi_1, \ldots, \phi_M \in \Phi$ needed to execute the command.

B. Hierarchical Semantic Frames

Consider a command $\Lambda \in \Sigma_{\text{R}}$ that uses verb $v \in V_{\text{R}}$, where V_{R} is the set of all verbs recognized in Σ_{R}. Each of the N words $\lambda_1, \ldots, \lambda_N$ in the command Λ can be parsed into a corresponding grammatical relation (part of speech), $\psi_1, \ldots, \psi_N \in \Psi$. For robots to understand a command, we need to specify a *lexical unit* that defines a command. Lexical units contain important information such as synonymous verbs $(v_1, \ldots, v_i \equiv v)$ and grammatical relations $(\psi_1, \ldots, \psi_j)$ that may be used in the command involving that verb. *Semantic frames* are evoked by a verb in a lexical unit and describe an action being taken in a scene, essentially mapping grammatical relations ψ to words λ in the verbal command Λ. Semantic frames can also have *children semantic frames*, which are more specific versions of a command. For example, a `turn` semantic frame is evoked by verbs "turn" or "twist" (as defined by the lexical unit), and may have more specific children frames `turn_left` and `turn_right`.

Our proposed *hierarchical semantic frames* (HSFs) are a data structure that allow robots to scalably map from restricted language commands $\Lambda \in \Sigma_{\text{R}}$ to grounded robot control primitives $\phi_1, \ldots, \phi_M \in \Phi$. Let $\Psi_{\text{req}} = \{(\psi_{\text{req}}, \psi_{\text{head}})\}$ and $\Psi_{\text{opt}} = \{(\psi_{\text{opt}}, \psi_{\text{head}})\}$ be sets of frame elements (ordered pairs of grammatical relations and their head dependency relations), where each frame element is required or optional, respectively. Formally, we define a HSF evoked by the command Λ as:

$$\text{HSF} = \begin{cases} (\Lambda, \Psi_{\text{req}}, \Psi_{\text{opt}}, \{\text{HSF}_{\text{child}}\}) & \text{parent HSF} \\ (\Lambda, \Psi_{\text{req}}, \Psi_{\text{opt}}, (a_1, \ldots, a_S)) & \text{grounded HSF} \end{cases}$$

where a parent HSF has a set of more specific children HSFs and a grounded HSF contains a sequence of frame actions, $a_1, \ldots, a_S \in A$. Frame actions can either be robot control primitives or other HSF commands in the restricted language, so $A = \Phi \cup \Sigma_{\text{R}}$. HSFs offer several improvements upon previous implementations of semantic frames, specifically RoboFrameNet [53]. The following sections detail the improvements in our HSFs, including required Ψ_{req} and optional Ψ_{opt} frame elements, head dependency relations ψ_{head}, and frame actions A.

1) Optional Frame Elements: Frame elements are grammatical relations $\psi_1, \ldots, \psi_j$ involved in a verbal command. RoboFrameNet [53] requires all frame elements to be parsed to evoke the lexical unit. This requirement is not robust to variations in command structure. HSFs differentiate between *required frame elements* Ψ_{req} and *optional frame elements* Ψ_{opt}, which allows more variation in commands, since command variations may not use all frame elements.

For example, the commands "give me the block" and "give me the red block" both evoke the `give_object` semantic frame. Defining an optional adjective modifier in Ψ_{opt} allows both commands to be recognized by telling the HSF to not always expect a modifier. "Red" provides optional information to differentiate one block from another but is not required to understand the command. Optional frame elements take advantage of the fact that within a restricted language, we expect some limited number of possible grammatical relations within a command. HSFs can be instantiated as long as each of the required elements Ψ_{req} are identified during parsing and can be mapped to words λ in the command Λ; the optional elements Ψ_{opt} provide helpful information, but do not prevent the HSF from being instantiated.

2) Argument Substitution through Head Dependency Relations: Head dependency relations tie a frame element to other elements it depends on. For example, the command "pick up the blue block" includes an adjectival modifier element "blue" and a direct object element "block." The head dependency relation ψ_{head} for "blue" would be the "block" since the adjectival modifier describes the direct object. HSFs make more use of the descriptors in a command by taking *head dependency relations* from the parser as *arguments* during instantiation. Note that any *dependency parser* will output these required *dependency relations*.

RoboFrameNet [53] and our HSF pipeline use the Stanford parser [24], which parses the head dependencies of each word in a command. However, whereas RoboFrameNet does not make use of these head dependency relations, our HSF pipeline does. Specifying the head dependency relations allows HSFs to differentiate between frame elements with the same grammatical relation type, and therefore recognize more complex commands. For example, consider the command "stack the red block on the blue block." When parsed, this command involves a direct object ψ_{dobj} (the block being stacked) and an indirect object ψ_{iobj} (the block being stacked on top of). Each of these objects have adjectival modifiers ψ_{adj} to differentiate the two objects. Previous implementations of semantic frames would not be able to differentiate between the two blocks or determine which block to act on, since $\psi_{\text{adj}} \rightarrow$ "red" and $\psi_{\text{adj}} \rightarrow$ "blue" but "red" $\neq$ "blue". In contrast, the `stack` HSF uses head dependency relations to differentiate between the adjectival modifiers; the head dependency relation for "red" is the direct object while the head dependency relation for "blue" is the indirect object. The grammatical relations and head dependency relations $(\psi_i, \psi_{\text{head}})$ of the words λ_i in the command are passed as arguments from the parser to the HSF to evoke the frame. Passing grammatical and head dependency relations as arguments from the parser allows HSFs to differentiate between words with the same grammatical relation type, make use of additional information in commands by substituting arguments within the frame, and understand more complex commands.

3) Frame Actions: The most important feature of HSFs is that they are grounded in robot action. Each HSF contains a sequence of frame actions $a_1, \ldots, a_S \in A$ required to carry out the commanded task. For commands to be grounded in robot action, the actions a_k listed within the HSF must be either: (a) a robot motion control primitive, $a_k \in \Phi$ (described further in Section III-C), or (b) a HSF command, $a_k \in \Sigma_R$. Note that the actions within a HSF come from action space $A = \Phi \cup \Sigma_R$. By allowing actions within a

HSF to be other HSF commands, we can create semantic frames that are ***hierarchies*** of other semantic frames. The hierarchical nature of HSFs results in a recursive expansion of actions to obtain a complete sequence of grounded actions for a command.

Hierarchy makes our HSF pipeline modular and scalable. Hierarchies of HSFs mean that high-level commands can be created from lower-level commands. For example, the `grasp` HSF is useful when defining higher-level frames such as `pick`, `place`, `put`, and `give_object_to`. With a basis of HSFs grounded in robot control primitives, we guarantee that high-level commands comprised of these basis elements can also be grounded in robot action. Users can communicate high-level commands to robots without thinking about the low-level action execution.

C. Robot Control Primitives

As discussed previously, the most important feature of HSFs is that each command Λ is grounded in robot actions $\phi_1, \ldots, \phi_M \in \Phi$. Depending on the robot, the robot control primitives can take various forms. HSFs do not depend on the form of the robot control primitives, just that some basis of primitives exists. For example, robot control primitives can be *affordance templates* [17], *affordance primitives* [38], [39], or a *control basis* [41]. One work defines an example control basis—a set of controller building blocks—as 6D pose, 3D position, alignment (relative rotation), and screw controllers [49].

In our experiments, we use the following robot control primitives Φ: open/close hand, move end-effector to target 6D pose, plan footstep trajectory, and execute footstep trajectory. For the robots we tested on (Fetch robot and NASA Johnson Space Center's Valkyrie robot [45], [22]), we determined that these primitives were sufficient for a wide variety of tasks and correspond directly to ROS actions within the motion control libraries running on these robots (MoveIt [10] for Fetch and IHMC Open Robotics Software [44] for Valkyrie).

IV. EXPERIMENTS AND RESULTS

Figure 2 describes the pipeline for using *hierarchical semantic frames* to ground verbal commands and execute the commanded actions. Figure 3 further details the grounding of verbal commands into sequences of robot control primitives. We assume human-in-the-loop perception through interactive object registration [15]. For the robot to execute a command, it needs to know where the required objects are in the scene.

To demonstrate the capabilities of HSFs, we performed several experiments on the Fetch robot and NASA Johnson Space Center's Valkyrie robot [45], [22], [23].

A. Verbs with Multiple Meanings

A single verb can have many different meanings depending on the context. HSFs are able to unambiguously determine what frame corresponds to the command. For example, as seen in Figure 4, the lexical unit for "give" defines several optional frame elements. The HSF for "give" has two

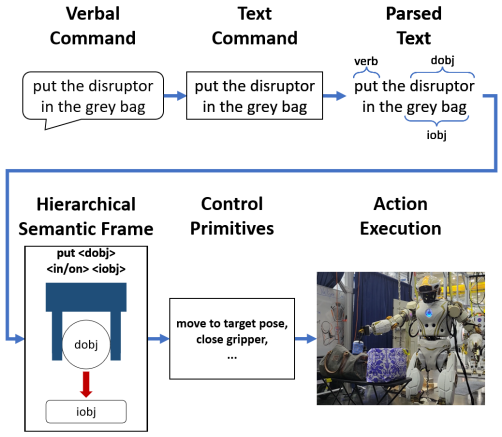


Fig. 2: Our HSF pipeline. The verbal command is parsed and used to evoke a HSF, which contains the sequence of actions needed to execute the command.

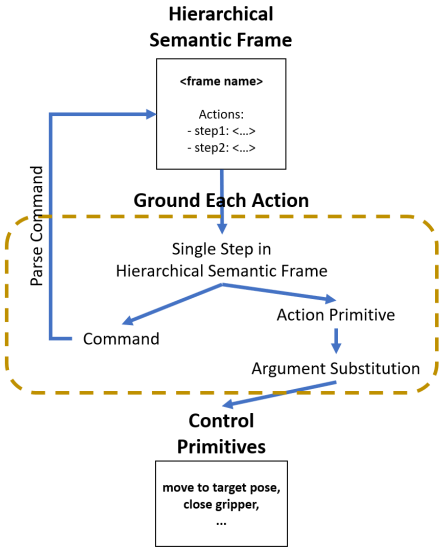


Fig. 3: Our pipeline for converting HSFs to the corresponding sequence of grounded robot control primitives. Each action in a HSF is either a grounded control primitive or another HSF command that can be recursively grounded.

children, `give_high_five` and `give_object_to`. Each child requires substitution of optional frame elements. Based on parsing of the grammatical relations, the HSF pipeline identifies which of the children is commanded.

Using the HSF pipeline, Valkyrie is able to understand and execute the commands "give me a high five" and "give Emily the disruptor," as seen in Figure 5. Though both commands use the same verb "give," the robot understands that the affordances required to execute these commands depend on the direct objects ("high five" and "disruptor" respectively). We see that HSFs allow the robot to accurately comprehend multiple meanings associated with these verbs and execute the commanded tasks.

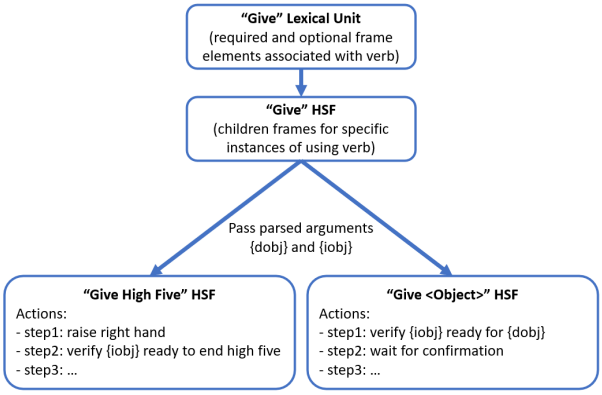
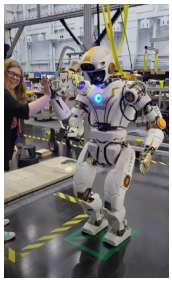
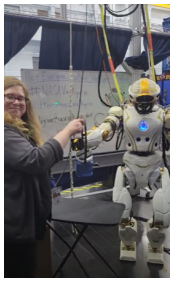


Fig. 4: Graphical representation of argument passing and instantiation of HSFs. Lexical units define optional frame elements and head dependency relations for each frame element, which give HSFs flexibility to understand related commands. A general `give` HSF contains two related children frames. Each child HSF contains actions that can use argument substitution and hierarchies of other recognizable commands such as "raise right hand," and "wait for confirmation."

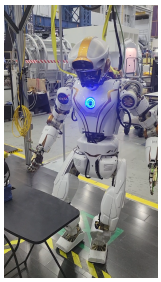


(a) "Give me a high five."

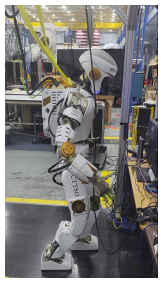


(b) "Give Emily the disruptor."

Fig. 5: HSFs allow robots to understand that the same verb (in this case, "give") involves different actions depending on the objects being acted on.



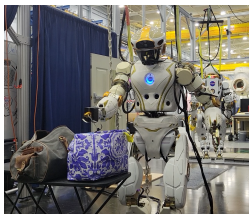
(a) "Go to Emily's desk."



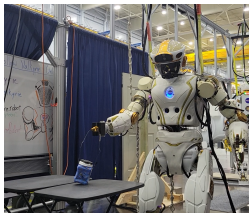
(b) "Go to Steven's desk."

Fig. 6: HSFs allow robots to scalably understand command variations. A single HSF (`go_to_desk`) represents all desk destinations by using argument substitution.

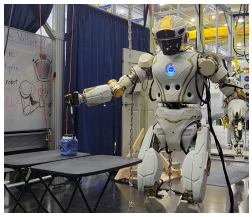
(a) "Put the disruptor in the grey bag."



(b) "Put the disruptor in the white bag."



(c) "Put the disruptor on the left table."



(d) "Put the disruptor on the right table."

Fig. 7: HSFs allow robots to understand compound nouns and differentiate between multiple similar objects (such as grey bag, white bag, and bag).

B. Command Variations

HSFs can scalably instantiate command variations due to argument substitution, as seen in Figure 6. Commands "go to Emily's desk" and "go to Steven's desk" only differ in terms of the final destination. Previous implementations of semantic frames require separate frames for each destination. Because of argument substitution, HSFs can represent all variations using a single frame. Both commands use a single HSF `go_to_desk` and argument substitution to handle variations in desk destination. HSFs allow the robot to scalably comprehend and execute command variations.

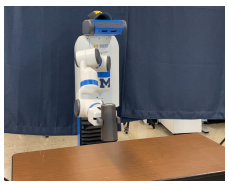
C. Compound Nouns

Previous implementations of semantic frames cannot effectively instantiate frames involving compound nouns. This means that any distinctions between nouns such as "grey bag" or "white bag" cannot be understood or acted on appropriately. Due to optional frame elements, head dependency relations, and argument substitution, HSFs can comprehend compound nouns and act on these objects accordingly, as seen in Figure 7. Because the robot comprehends compound nouns, it correctly differentiates between similar object types and executes the appropriate affordances with respect to those objects. Furthermore, these experiments demonstrate recognition of command variations, as all of these experiments use the same `put_object_in_on` HSF.

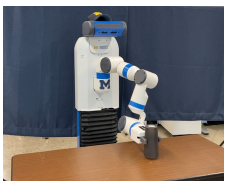
D. Command Ambiguity

Our HSF pipeline works on multiple robots that execute grounded control primitives in different ways. Figure 8 shows the Fetch robot executing the command "move that to the left." Since HSFs do not restrict the form of robot control primitives, commands can be executed on multiple robots. This experiment also demonstrates that the HSF pipeline can make sense of some command ambiguity. Since the only object present is the cup, the robot understands that the only possible grounding for "that" is the cup. HSFs effectively handle some ambiguity in language and can be executed by multiple robots.

(a) Operator: “Move that to the left.”



(b) Fetch picks up the cup.



(c) Fetch places the cup to its left.

Fig. 8: HSFs allow multiple robots to ground commands in action. The robot can also understand some ambiguity in the command, and understands that “move *that* to the left” can only refer to the cup.

Task Type	Total Commands	Mean Time (s)	SD Time (s)
Multiple Verb Meanings	24	0.309	0.189
Command Variations	5	0.365	0.179
Compound Nouns	76	0.268	0.201
Additional Trials	32	0.324	0.163
All Tasks	**137**	**0.292**	**0.192**

TABLE I: Mean and standard deviation (SD) of processing times for HSF commands. Processing times are reported for each task type as well as aggregate data for all HSF commands.

E. Command Processing

To evaluate the safety and responsiveness of our HSF pipeline, we recorded the processing time for each HSF. Due to the added capabilities of HSFs—specifically recursive definitions of frame actions—we need to ensure that commands can be processed quickly, especially if command execution needs to be interrupted for safety purposes.

Table I shows the mean and standard deviation of processing times for each task type and aggregate data across all experiments. Trials were repeated to improve the human-in-the-loop object registration and verify command grounding. We see that HSFs can be processed quickly, ensuring responsiveness and safety of robots listening for HSF commands.

F. Scalability of HSFs

To demonstrate the scalability of HSFs, we considered taking actions on a fixed set of 16 objects from the YCB dataset [8]. On this fixed set of objects, we consider the actions “put” (“put the pear in the bowl” or “put the tuna fish can on the cracker box”), “pour” (“pour the mug into the pitcher”), and “stir” (“stir the skillet with the spoon”). To express these actions over our subset of objects, we would need 18 HSFs or almost 800 RoboFrameNet semantic frames, as seen in Figure 9. Since RoboFrameNet requires distinct semantic frames for every command variation, recognizing a new action requires significantly more semantic frames. In contrast, the flexibility of HSFs means that the number of HSFs only increases when new verbs or verb meanings need to be recognized. Based on these findings, we conclude that compared to RoboFrameNet, HSFs scale much better in terms of number of semantic frames required to understand new actions.

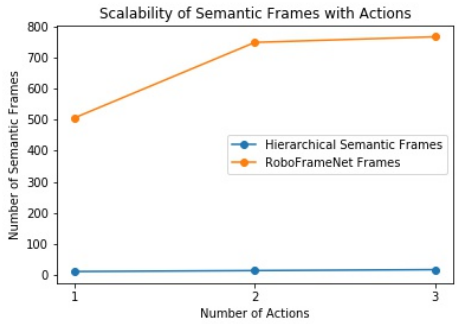


Fig. 9: Scalability of HSFs and RoboFrameNet [53] semantic frames based on actions taken involving objects from the YCB dataset [8]. We consider a fixed set of 16 objects from the YCB dataset and actions “put”, “pour”, and “stir.”

V. Discussion and Conclusion

The robot’s ability to successfully perform a variety of tasks from verbal commands demonstrates the power of our proposed *hierarchical semantic frames* as a middle-ground between restricted language and robot action. HSFs allow robots to scalably recognize a wide variety of commands. Since HSFs are grounded in robot control primitives, we demonstrate that the robot can understand spoken commands and physically execute these commands. Through hierarchies of HSFs, we ensure robots can execute tasks from commands that abstract individual actions from the user.

Future work includes incorporating notions of pre- and post-conditions, actions involving greater numbers of objects, and probabilistic reasoning [51], [1] over ambiguous language. Our experiments require that the robot interacts with known, labelled, and registered objects. Future work would also involve testing the effectiveness of HSFs with autonomous segmentation and registration. Overall, our proposed HSFs demonstrate the power of commanding robots through actions grounded in robot control primitives. Our *hierarchical semantic frames* allow users to intuitively interact with robots in a variety of tasks.

Acknowledgments

This work was supported in part by NASA Space Technology Graduate Research Opportunity (NSTGRO) grant 80NSSC20K1200. The authors would like to thank the members of the NASA Dexterous Robotics Team. Special thanks to Steven Jens Jorgensen, Misha Savchenko, Mark Paterson, Ian Chase, Lewis Hill, and Mina Kian for their mentorship, input, and robot ops support.

REFERENCES

[1] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[2] J. S. Albus, A. J. Barbera, and R. N. Nagel, "Theory and Practice of Hierarchical Control," National Bureau of Standards, 1980.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A Survey of Robot Learning from Demonstration," *Robotics and Autonomous Systems*, 2009.

[4] R. Atienza and A. Zelinsky, "Active Gaze Tracking for Human-Robot Interaction," *IEEE International Conference on Multimodal Interfaces*, 2002.

[5] D. H. Ballard, "Task Frames in Robot Manipulation," pp. 16-22, 1984.

[6] V. Bruce, "What the Human Face Tells the Human Mind: Some Challenges for the Robot-Human Interface," *Advanced Robotics*, 1993.

[7] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential Composition of Dynamically Dexterous Robot Behaviors," *International Journal of Robotics Research*, 1999.

[8] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols," *IEEE Robotics and Automation Magazine*, 2015.

[9] S. Chernova, J. Orkin, and C. Breazeal, "Crowdsourcing HRI through Online Multiplayer Games," *AAAI Fall Symposium Series*, 2010.

[10] S. Chitta, I. Sucan, and S. Cousins, "MoveIt! ROS Topics," *IEEE Robotics and Automation Magazine*, 2012.

[11] B. R. Cowan, N. Pantidi, D. Coyle, K. Morrissey, P. Clarke, S. Al-Shehri, D. Earley, and N. Bandeira, ""What Can I Help You With?": Infrequent Users' Experiences of Intelligent Personal Assistants," *International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2017.

[12] F. Duvallet, M. R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz, "Inferring Maps and Behaviors from Natural Language Instructions," *Experimental Robotics, Spring International Publishing*, 2016.

[13] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to Do and How to Do It: Translating Natural Language Directives Into Temporal and Dynamic Logic Representation for Goal Management and Action Execution," *IEEE International Conference on Robotics and Automation*, 2009.

[14] J. J. Gibson, "The Theory of Affordances," pp. 67-82, 1977.

[15] M. Hagenow, M. Zinn, T. Fong, E. Laske, and K. Hambuchen, "Affordance Template Registration via Human-in-the-Loop Corrections," *arXiv prepring arXiv:2109.13649*, 2021.

[16] S. Hart and R. Grupen, "Natural Task Decomposition with Intrinsic Potential Fields," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2507–2512, 2007.

[17] S. Hart, S. Dinh, and K. Hambuchen, "The Affordance Template ROS Package for Robot Task Programming," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6227-6234, 2015.

[18] R. Hartley and F. Pipitone, "Experiments with the Subsumption Architecture," *IEEE International Conference on Robotics and Automation (ICRA)*, 1991.

[19] S. Hemachandra, F. Duvallet, T. M. Howard, N. Roy, A. Stentz, M. R. Walter, "Learning Models for Following Natural Language Directions in Unknown Environments," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[20] K. M. Hermann, D. Das, J. Weston, and K. Ganchev, "Semantic Frame identification with Distributed Word Representations," *Association for Computational Linguistics*, 2014.

[21] T. M. Howard, S. Tellex, and N. Roy, "A Natural Language Planner Interface for Mobile Manipulators," *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[22] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, M. Paterson, J. Sanchez, P. Beeson, S. Gee, S. Hart, A. H. Quispe, R. Griffin, I. Lee, S. McCrory, L. Sentis, J. Pratt, and J. S. Mehling, "Deploying the NASA Valkyrie Humanoid for IED Response: An Initial Approach and Evaluation Summary," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2019.

[23] S. J. Jorgensen, M. Wonsick, M. Paterson, A. Watson, I. Chase, and J. S. Mehling, "Cockpit Interface for Locomotion and Manipulation Control of the NASA Valkyrie Humanoid in Virtual Reality (VR)," NASA Technical Reports Server, 2022. [Online]. Available: `https://ntrs.nasa.gov/citations/20220007587`

[24] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," *Association for Computational Linguistics*, 2003.

[25] T. Kollar, S. Tellex, D. Roy, N. Roy, "Grounding Verbs of Motion in Natural Language Commands to Robots," *International Symposium on Experimental Robotics (ISER)*, 2010.

[26] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward Understanding Natural Language Directions," *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[27] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot Learning from Demonstration by Constructing Skill Trees," *International Journal of Robotics Research*, 2012.

[28] E. Luger and A. Sellen, ""Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents," *CHI Conference on Human Factors in Computing Systems*, 2016.

[29] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions," *AAAI*, 2006.

[30] C. Matuszek, D. Fox, and K. Koscher, "Following Directions Using Statistical Machine Translation," *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[31] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to Parse Natural Language Commands to a Robot Control System," *Experimental Robotics*, 2013.

[32] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox, "Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions," *AAAI Conference on Artificial Intelligence*, 2014.

[33] J. Mu and A. Sarkar, "Do We Need Natural Language? Exploring Restricted Language Interfaces for Complex Domains," *CHI Conference on Human Factors in Computing Systems*, 2019.

[34] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and Generalization of Complex Tasks from Unstructured Demonstrations," *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[35] S. Niekum, S. Chitta, A. G. Barto, B. Narthi, and S. Osentoski, "Incremental Semantically Grounded Learning from Demonstration," *Robotics: Science and Systems (RSS)*, 2013.

[36] O. Palinko, F. Rea, G. Sandini, and A. Sciutti, "Eye Gaze Tracking for a Humanoid Robot," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015.

[37] R. Paul, J. Arkin, N. Roy, and T. M. Howard, "Efficient Grounding of Abstract Spatial Concepts for Natural Language Interaction with Robot Manipulators," *Robotics: Science and Systems (RSS)*, 2016.

[38] A. Pettinger, C. Elliot, P. Fan, and M. Pryor, "Reducing the Teleoperator's Cognitive Burden for Complex Contact Tasks using Affordance Primitives," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[39] A. Pettinger, F. Alambeigi, and M. Pryor, "A Versatile Affordance Modeling Framework using Screw Primitives to Increase Autonomy During Manipulation Contact Tasks," *IEEE Robotics and Automation Letters*, 2022.

[40] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Nullspace Composition of Control Laws for Grasping," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[41] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Manipulation Gaits: Sequences of Grasp Control Tasks," *IEEE International Conference on Robotics and Automation (ICRA)* vol. 1, pp. 801–806, 2004.

[42] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Null-Space Grasp Control: Theory and Experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282–295, 2010.

[43] M. Porcheron, J. E. Fischer, S. Reeves, and S. Sharples, "Voice Interfaces in Everyday Life," *CHI Conference on Human Factors in Computing Systems*, 2018.

[44] J. Pratt, P. Neuhaus, D. Stephen, S. Bertrand, D. Calvert, S. McCrory, G. Robert, G. Wiedebach, I. Lee, D. Duran, and J. Carff, "IHMC Open Robotics Software," Institute for Human and Machine Cognition (IHMC), 2021. [Online]. Available: `https://github.com/ihmcrobotics/ihmc-open-robotics-software`

[45] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, and R. Berka, "Valkyrie: NASA's First Bipedal Humanoid Robot," *Journal of Field Robotics*, 2015.

[46] J. Roh, K. Desingh, A. Farhadi, and D. Fox, "LanguageRefer: Spatial-Language Model for 3D Visual Grounding," *Conference on Robot Learning (CoRL)*, 2021.

[47] K. Rohanimanesh, R. Platt, S. Mahadevan, and R. Grupen, "Coarticulation in Markov Decision Processes," *Advances in Neural Information Processing Systems*, 2004.

[48] J. Ruppenhofer, M. Ellsworth, M. Schwarzer-Petruck, C. R. Johnson, and J. Scheffczyk, "FrameNet II: Extended Theory and Practice," *International Computer Science Institute*, 2016.

[49] E. Sheetz, X. Chen, Z. Zeng, K. Zheng, Q. Shi, and O. C. Jenkins, "Composable Causality in Semantic Robot Programming," *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[50] M Shridhar, L. Manuelli, and D. Fox, "Cliport: What and Where Pathways for Robotic Manipulation," *Conference on Robot Learning (CoRL)*, 2022.

[51] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding Natural Language Commands for Robotics Navigation and Mobile Manipulation," *AAAI Conference on Artificial Intelligence*, 2011.

[52] S. Tellex, N. Gopalan, H. Kress-Gazit, C. Matuszek, "Robots that Use Language," *Annual Review of Control, Robotics, and Autonomous Systems*, 2020.

[53] B. J. Thomas and O. C. Jenkins, "RoboFrameNet: Verb-Centric Semantics for Actions in Robot Middleware," *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[54] T. Tojo, Y. Matsusaka, T. Ishii, and T. Kobayashi, "A Conversational Robot Utilizing Facial and Body Expressions," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2000.

[55] D. Trandabat and D. Cristea, "Natural Language Processing Using Semantic Frames," Ph.D. dissertation, University "Alexandru Ioan Cuza" of Iaşi, Romania, 2010.

[56] Y. Y. Wang, L. Deng, and A. Acero, "Semantic Frame-Based Spoken Language Understanding," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, 2011.

[57] T. Winograd, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," MIT Technical Report, 1971.

[58] T. Winograd, "SHRDLU: A System for Dialog," 1972.

[59] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, "Computational Models of Affordance in Robotics: A Taxonomy and Systematic Classification," *Adaptive Behavior*, 2017.