Lang2LTL-2: Grounding Spatiotemporal Navigation Commands Using Large Language and Vision-Language Models

Jason Xinyu Liu[1], Ankit Shah[1], George Konidaris[1], Stefanie Tellex[1], and David Paulius[1]
[1]Brown University, USA

***Abstract*— Grounding spatiotemporal navigation commands to structured task specifications enables autonomous robots to understand a broad range of natural language and solve long-horizon tasks with safety guarantees. Prior works mostly focus on grounding spatial or temporally extended language for robots. We propose Lang2LTL-2, a modular system that leverages pretrained large language and vision-language models and multimodal semantic information to ground spatiotemporal navigation commands in novel city-scaled environments without retraining. Lang2LTL-2 achieves 93.53% language grounding accuracy on a dataset of 21,780 semantically diverse natural language commands in unseen environments. We run an ablation study to validate the need for different modalities. We also show that a physical robot equipped with the same system without modification can execute 50 semantically diverse natural language commands in both indoor and outdoor environments [1].**

Fig. 1: Our system Lang2LTL-2 grounds spatiotemporal navigation commands in indoor and outdoor environments. The spatial and temporal components of the example commands are highlighted in blue and red, respectively.

I. Introduction

When giving directions, humans often use natural language that describes goals, as well as temporal and spatial constraints. For example, consider the command "Visit the Starbucks, only then go to the red car to the right of the building, and always avoid the crowded restaurant near the cafe." An autonomous robot following this spatiotemporal command must understand that it specifies a temporally extended task of visiting two locations in a strict order and avoiding the third throughout the execution. The robot must ground the three referring expressions, i.e., "the Starbucks," "the car," and "the crowded restaurant," to specific locations with respect to other landmarks in the environment.

Existing approaches focus on developing the robot's spatial or temporal reasoning ability separately. Many works develop systems to ground natural language commands that contain rich spatial relations in indoor [1, 2, 3] and outdoor [4, 5] environments. Using a map that contains multimodal semantic information enables robots to identify various target landmarks with respect to others in the environment, yet these approaches cannot handle complex temporal constraints. Separately, structured task specifications, like linear temporal logic (LTL), can capture a wide range of semantically diverse temporal patterns [6] and enable the synthesis of verifiable robot behaviors with safety guarantees. However, systems that can ground complex temporal language have limited spatial reasoning capability [7, 8, 9].

To achieve the best of both worlds, we introduce a modular system that grounds spatiotemporal navigation commands for robots. Our language grounding system Lang2LTL-2 uses large language models (LLMs) to recognize spatial referring expressions, like "the red car to the right of the building," and to translate language commands to LTL task specifications, which are compatible with many planning and reinforcement learning algorithms [10, 11, 12, 13, 14]. Using pretrained vision-language models (VLMs) and text embedding, Lang2LTL-2 grounds referring expressions to specific locations in novel city-scaled environments without retraining, given a semantic database of textual and visual descriptions of the landmarks.

We evaluated Lang2LTL-2 on a dataset of 21,780 semantically diverse spatiotemporal commands with 1,723 spatial referring expressions, 19 spatial relations, and 15 temporal patterns. We also ran an ablation study and showed that using multimodal semantic information for spatiotemporal language grounding outperforms using any modality alone. Finally, we demonstrated that a mobile robot equipped with the same system without modification could execute 50 semantically diverse spatiotemporal commands in both indoor and outdoor environments.

II. Preliminaries

A. Large Language Models and Vision-Language Models

Large language models (LLMs) are transformer neural networks [15] trained to maximize the probability of a successive token given a context window. They achieve the SoTA performance on a wide variety of natural language processing tasks [16]. Pretrained LLMs can also produce high-dimensional embedding vectors of text. We can measure

Correspondence to Jason Xinyu Liu (xinyu_liu@brown.edu)
[1]Code, datasets videos, and supplementary materials are at spatiotemporal-ground.github.io.

the semantic similarity of two pieces of text by computing the cosine similarity of their embeddings. In this work, we used OpenAI's GPT-4 model [17] and embedding API for text completion and text embedding, respectively, and a fine-tuned T5-base model [18] to translate natural language commands to temporal task specification.

Vision-language models (VLMs) are multimodal models jointly trained on text and images [19]. They produce SoTA results on many language-conditioned vision tasks [20], e.g., object detection [21, 22], image captioning [23], image retrieval [24], and visual question answering [25]. In this work, we prompted the GPT-4V(ision) model [26] to generate captions for images of landmarks and objects.

B. Temporal Task Specification

Linear temporal logic (LTL) [27] is a promising task specification language for human-centered specification elicitation [7, 28, 29, 30], planning [29, 14], and reinforcement learning [10, 31]. The syntax of LTL is defined through the following recursive grammar:

$$\varphi := \alpha \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \ \mathbf{U} \ \varphi_2 \tag{1}$$

Here α represents an atomic Boolean proposition, and φ, φ_1, φ_2 are any valid LTL formulas. The operators $\neg$ (not) and $\vee$ (or) are identical to propositional logic operators. The formula $\mathbf{X}\varphi$ holds if φ holds at the next time step, and $\varphi_1 \ \mathbf{U} \ \varphi_2$ holds if φ_1 holds at least until φ_2 first holds, which must happen at the current or a future time. LTL syntax also admits abbreviated operators defined through the compositions of the primitive operators. In this work, we use the operators $\wedge$ (and), $\mathbf{F}$ (read "finally" or "eventually"), and $\mathbf{G}$ (read "globally" or "always"). $\mathbf{F}\varphi$ specifies that the formula φ must hold at least once in the future, and $\mathbf{G}\varphi$ specifies that φ must always hold.

C. Task Execution for Temporal Task Specification

A linear temporal logic (LTL) formula can be transformed to a Büchi automaton [32, 33]. State transitions in the environment induce state transitions in the automaton, so we can track task progress by tracking the automaton's state transition. We can compute a policy on the product MDP of the task automaton and the environment MDP. Our system is compatible with many planning and reinforcement learning algorithms that solve LTL task specification [10, 11, 34, 12, 13, 14].

III. Problem Definition

Our language grounding system Lang2LTL-2 receives a natural language utterance u from users that specifies a navigation task in an environment modeled as $\langle \mathcal{S}, \mathcal{A}, T \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ represent the robot's states and actions, and $T(s, a) \rightarrow s'$ captures the transition dynamics. In this work, we consider navigational actions that transition a robot from one location to another in the environment represented as a semantic map. We assume the robot has access to a multimodal semantic database $\mathcal{D} = \{p : (d, f)\}$, where p is a proposition that uniquely represents a landmark in the environment, d is a semantic description of the landmark, and $f : \mathcal{S} \rightarrow \{0, 1\}$ is a Boolean-valued function that determines the true value of the proposition in a given state. The semantic information of a landmark can be a textual description, including its name, amenity, street address, etc, an image, or both. Lang2LTL-2 translates the input command to a linear temporal logic (LTL) formula φ and grounds its propositions to landmarks in the real world. We assume the robot can track its state in a semantic map and has access to an automated planner that, given an LTL formula as task specification, produces a trajectory in the semantic map. Many planning and reinforcement learning algorithms [10, 11, 12, 13, 14] are compatible with LTL task specification. We use the AP-MDP planner by Oh et al. [12]. Figure 2 shows an example execution by the full system, i.e., language grounding and planning.

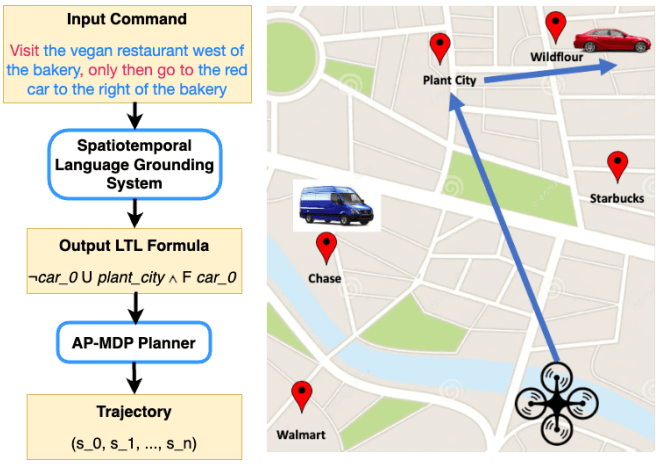


Fig. 2: An example shows an input spatiotemporal navigation command (whose spatial and temporal components are highlighted in blue and red, respectively), an output LTL formula whose propositions are grounded to physical landmarks, and an execution trajectory in the environment.

IV. Lang2LTL-2: Spatiotemporal Language Grounding

We approach the problem of spatiotemporal language grounding with a modular design, where we extract spatial referring expressions and translate temporal commands using large language models, and ground referring expressions to physical landmarks using a vision-language model and text embedding. Our system Lang2LTL-2 produces a grounded temporal task specification incorporating the grounded referring expressions and the spatial relations. Figure 3 shows an overview of our language grounding system.

A. Spatial Referring Expression Recognition (SRER)

The spatial referring expression recognition (SRER) module identifies spatial referring expressions in a given language command. Referring expressions (REs) are noun phrases, pronouns, and proper names that refer to some entity in an environment, such as landmarks and objects [35]. In this work, we only consider noun phrases and proper names

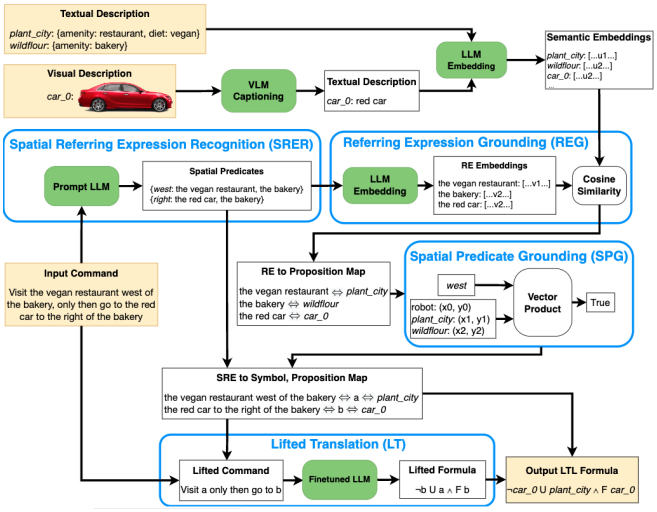


Fig. 3: Lang2LTL-2 Language Grounding System Overview: input and output are in yellow blocks; modules are in blue blocks; pretrained and fine-tuned models are in green blocks.

and leave the coreference resolution problem to future work. Spatial referring expressions (SREs) are phrases where referring expressions are connected by a spatial relation. For example, in the language command "Go to the red car to the right of the bakery," the SRE "the red car to the right of the bakery" contains two REs, "the red car" and "the bakery," termed the figure e_f and the ground e_g, respectively, by Landau and Jackendoff [36]. The figure e_f and the ground e_g are connected by the spatial relation r "to the right of." We define a diverse set $\mathcal{R}$ of 19 spatial relations, such as near, in front of, behind, to the left of, to the right of, between, and four cardinal directions. The SRER module extracts referring expressions and their spatial relations from a spatiotemporal language command by prompting an LLM. We use GPT-4 [17]. The output of the SRER module is a spatial predicate denoted by $\{r : (e_f, e_g)\}$. Please see supplementary materials for the prompt used for SRER and the complete set of spatial relations.

B. Referring Expression Grounding (REG)

To ground the referring expressions (REs) e_f and e_g to physical landmarks in the environment, we use a multimodal semantic database with textual and visual descriptions of landmarks. Having both modalities enables the referring expression grounding (REG) module to ground more complex REs and improves grounding accuracy. For certain REs, one modality is more descriptive than the other. For example, a textual description including a landmark's amenity and cuisine type better matches the RE "the vegan restaurant" than an image of the restaurant front. The REG module is important for identifying possible candidates for each RE, especially when the environment contains multiple similar landmarks or objects.

We prompt a pretrained vision-language model (VLM) to generate captions of images with the question, "What is the most obvious object in this image?" In this work, we use GPT-4V(ision) [26]. We then use an LLM to generate text embeddings for the image captions, the textual descriptions of landmarks in the semantic database, and the query REs (i.e., e_f and e_g) extracted from the language command. Finally, we use the cosine similarity between text embeddings to find the landmarks that best match the query REs. Let $g_{caption} : i \rightarrow t$ be the function that generates a caption t for image i parameterized by the weights of the VLM, and $g_{embed} : t \rightarrow z$ be the function that computes an n-dimensional embedding z of a text string t parameterized by the weights of the LLM. The cosine similarity score is defined as follows,

$$score(e_{f/g}, t) = \frac{g_{embed}(e_{f/g})^T g_{embed}(t)}{||g_{embed}(e_{f/g})|| \cdot ||g_{embed}(t)||}, \quad (2)$$

where we substitute $t = g_{caption}(i)$ when the semantic description of the landmark is an image i, and $e_{f/g}$ denotes the query RE being the figure e_f or the ground e_g.

We also explored using CLIP's text and image encoders [19] to encode text and images, then the cosine similarity of the text and image embeddings to find the best matching landmark for a query RE. However, we discovered that the gap between the text and image embedding spaces is large for the pretrained CLIP model. Liang et al. [37] documented this phenomenon in more detail. Instead of training another neural network to align the text and image embedding spaces, we use a pretrained VLM to transcribe images to text and work solely in the text embedding space.

C. Spatial Predicate Grounding (SPG)

After grounding the figure e_f and the ground e_g to candidate landmarks, we perform spatial predicate grounding (SPG) to identify the most likely landmark referred to by e_f given e_g and the spatial relation r. We assume that users give commands with respect to the robot's initial location. For each spatial referring expression (SRE) and its corresponding spatial predicate $\{r : (e_f, e_g)\}$, we rank all the candidate landmarks of e_f based on the product of the similarity scores computed by the referring expression grounding (REG) module for the candidate landmarks of

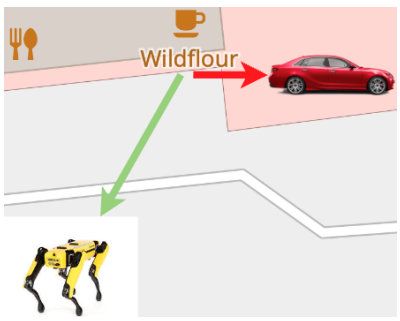


Fig. 4: An illustration of the ground vector and the figure vector, depicted as the green and the red arrow, respectively, computed by the spatial predicate grounding (SPG) module (Section IV-C) to resolve the spatial referring expression "the red car to the right of the bakery."

e_f and e_g, then select the proposition p_f with the highest product score,

$$p_f^* = \underset{p_f:(d_f,\_)\in\mathcal{D}, p_g:(d_g,\_)\in\mathcal{D}}{\arg\max} score(e_f, d_f) \cdot score(e_g, d_g). \tag{3}$$

To validate each pair of candidate landmarks, we first compute a ground vector from the ground landmark to the robot, which serves as an anchor for computing the range where the figure landmark should be. We then compute a figure vector from the ground landmark to the figure landmark. Depending on the spatial relation, we compute a range where the figure vector should lie based on the ground vector. Figure 4 illustrates the ground and the figure vectors for the SRE "the red car to the right of the bakery."

For each known spatial relation $r \in \mathcal{R}$, we specify a set of rules to validate a pair of candidate landmarks for e_f and e_g. In the example of "the red car to the right of the bakery" the spatial relation "to the right of" means the figure vector must lie within the half circle between the ground vector and 180 degrees from the ground vector. Please see supplementary materials for the definition of all spatial relations. We also specify a distance threshold in meters between a figure and the ground to eliminate candidate figures too far from the ground. To resolve an unseen spatial relation, we use LLM text embedding and cosine similarity to find the most semantically similar spatial relation $r \in \mathcal{R}$.

D. Lifted Translation (LT)

After the SRER module extracts all the spatial referring expressions (SREs) from a given command, we transform it into a lifted command by substituting the SREs with symbols, which are grounded to physical landmarks by the REG (Section IV-B) and the SPG (Section IV-C) modules. For example, the input command "Go to the red car to the right of the bakery" is transformed to a lifted command "Go to a" where the symbol a substitutes the SRE "the red car to the right of the bakery." We then translate the lifted command to a lifted LTL formula compatible with many planning and reinforcement learning algorithms [10, 11, 12, 13, 14]. We evaluate the following models for lifted translation.

Fine-tuned LLM: Liu et al. [8] tested four models that use LLMs for lifted translation. The T5-Base (220M parameters) model [18] fine-tuned on the semantically diverse dataset they collected overperformed the fine-tuned GPT-3 [38], the Prompt GPT-3 [38] and the Prompt GPT-4 [17] models. Thus, we use their best-performing model fine-tuned T5-Base through HuggingFace's Transformer library [39].

Retrieval Augmented Generation (RAG): We evaluate retrieval augmented generation (RAG), which dynamically constructs a prompt to an LLM based on the query [40] for lifted translation. To translate a lifted command to a lifted LTL formula with RAG, we use cosine similarity of text embeddings to find semantically similar commands from the lifted dataset collected in [8], then use these commands and their corresponding LTL formulas as in-context examples to query GPT-4 [17]. We test varying numbers of in-context examples. Please see supplementary materials for an example prompt used for RAG.

V. Evaluation of Language Grounding

We conducted three sets of evaluations of our spatiotemporal language grounding system Lang2LTL-2: 1) a modular evaluation, where we tested the performance of individual modules introduced in Section IV, 2) a full system evaluation, where we evaluated the final output of our system, and 3) an ablation study of the text and the image modality.

A. Dataset

Our evaluation used four city-scaled environments with an increasing number of landmarks, i.e., 9, 34, 44, and 175. The landmarks were described by text from OpenStreetMap [41] (e.g., names, street addresses, amenities, and GPS coordinates, etc.) and images from Google StreetView [42]. Having a dataset where landmarks are described by both modalities helps evaluate whether the referring expression grounding (REG) module can use a proper modality to ground referring expressions correctly to landmarks.

To obtain semantically diverse spatiotemporal navigation commands, we first collected 1,723 spatial referring expressions (SREs) with respect to the robot's initial location from human users, then substituted the SREs in the 1,089 lifted natural language commands provided by [8]. The lifted commands cover 15 temporal patterns for common robotic tasks, each with 20 to 38 lifted commands. For example, given the lifted command "Visit a only then go to b", we can substitute the symbols a and b with the SREs "the vegan restaurant west of the bakery" and "the red car," respectively, to obtain the grounded natural language command "Visit the vegan restaurant west of the bakery only then go to the red car." We constructed 21,780 unique spatiotemporal language commands using five seeds to sample SREs for substitution. The commands contain varying numbers of SREs ranging from one to five.

B. Modular Evaluation

We first evaluated each module introduced in Section IV on the semantically diverse dataset introduced in Section V-A. All results were averaged over five seeds.

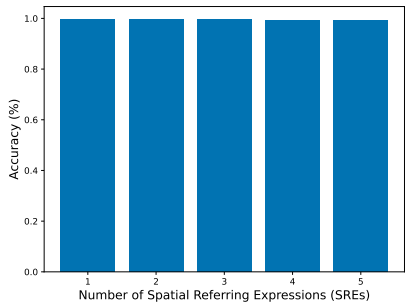
Spatial Referring Expression Recognition (SRER): We evaluated the LLM's ability to correctly extract all spatial referring expressions (SREs) from a natural language command and identify their spatial predicates described in Section IV-A, i.e., $\{r : (e_f, e_g)\}$ with spatial relation r, figure e_f and ground e_g. As shown in Table I, the SRER module can reliably recognize SREs and their corresponding spatial predicates in language commands from unseen environments. Figure 5a further demonstrates that SRER achieves nearly perfect performance across commands with varying numbers of SREs. Occasionally, when a language command contains five SREs of large lengths, the SRER module may fail to parse an SRE to the correct spatial predicate.

Referring Expression Grounding (REG): We evaluated the REG module's ability to ground referring expressions,

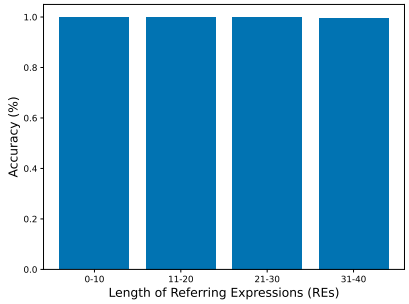
TABLE I: Modular Performance

Module		Accuracy				
		City 1 (9 landmarks)	*City 2* (34 landmarks)	*City 3* (44 landmarks)	*City 4* (175 landmarks)	*Average*
SRER		$99.45 \pm 0.12\%$	$99.43 \pm 0.26\%$	$99.56 \pm 0.63\%$	$99.39 \pm 0.21\%$	$99.46 \pm 0.34\%$
REG	Top-1	$99.68 \pm 0.72\%$	$97.98 \pm 1.07\%$	$88.74 \pm 2.14\%$	$78.35 \pm 1.97\%$	$91.19 \pm 8.84\%$
	Top-5	$100.00 \pm 0.00\%$	$100.00 \pm 0.00\%$	$99.56 \pm 0.24\%$	$99.15 \pm 0.34\%$	$99.68 \pm 0.41\%$
	Top-10	$100.00 \pm 0.00\%$	$100.00 \pm 0.00\%$	$99.70 \pm 0.17\%$	$99.98 \pm 0.05\%$	$99.92 \pm 0.15\%$
SPG		$100.00 \pm 0.00\%$	$100.00 \pm 0.00\%$	$99.53 \pm 0.33\%$	$99.35 \pm 1.46\%$	$99.72 \pm 0.75\%$
LT	Finetuned T5-base	$99.45 \pm 0.00\%$	$99.45 \pm 0.00\%$	$99.45 \pm 0.00\%$	$99.45 \pm 0.00\%$	$99.45 \pm 0.00\%$
	RAG-10	$69.33 \pm 0.25\%$	$70.34 \pm 0.13\%$	$69.65 \pm 0.58\%$	$70.39 \pm 0.84\%$	$69.93 \pm 0.62\%$
	RAG-50	$83.79 \pm 0.06\%$	$83.93 \pm 0.12\%$	$83.75 \pm 0.52\%$	$83.93 \pm 0.65\%$	$83.85 \pm 0.33\%$
	RAG-100	$88.20 \pm 0.58\%$	$88.25 \pm 1.04\%$	$87.79 \pm 0.39\%$	$87.70 \pm 0.13\%$	$87.98 \pm 0.54\%$

i.e., figures and grounds, to the correct physical landmarks described by text and images in the semantic map. We observe in Table I that the Top-1 accuracy decreases as the number of landmarks increases from City 1 to City 4. With more landmarks, there are more instances that share similar textual or visual features. For example, there may be multiple cafe shops or red bicycles in a large environment. However, as we increase the number of top candidates from 1 to 10, REG achieves nearly perfect accuracy. Since the REG module provides candidate landmarks of figures and grounds to the SPG module (evaluated next), we hypothesize that as long as the correct landmark is among the top candidates, our system can still ground figures to the correct landmarks. We used 10 as the number of candidates for REG. Figure 5b shows that as the complexity of REs increases, the REG module consistently achieves near-perfect Top-10 accuracies. These results align with that reported by Liu et al. [8].



(a) SRER Accuracy vs. Utterance Complexity



(b) REG Accuracy vs. RE Complexity

Fig. 5: Figure 5a shows the accuracies of the spatial referring expression recognition (SRER) module as the complexity of utterances (measured by the number of SREs in an utterance) increases. Figure 5b shows the accuracy of the referring expression grounding (REG) module as the complexity of REs (measured by string length) increases.

Spatial Predicate Grounding (SPG): Our evaluation of the SPG module assessed whether the correct figure landmarks could be identified using spatial reasoning described in Section IV-C. As shown in Table I, SPG performs uniformly well across all environments. The few failure cases were due to the instances where the distance between the figure and the ground landmarks was larger than the search threshold.

Lifted Translation (LT): Liu et al. [8] conducted a comprehensive evaluation of the generalization capability of various fine-tuned and pretrained LLMs for lifted translation. We compared the accuracies of the best-performing model in [8], i.e., T5-base fine-tuned on a large composted dataset, and retrieval augmented generation (RAG) [40] with varying numbers of in-context examples. The fine-tuned model achieved the highest accuracies across all environments, which indicates the composed dataset constructed by Liu et al. [8] covers most temporal patterns we consider in this work. As we increase the number of in-context examples for RAG from 10 to 100, the maximum tokens allowed by GPT-4 [17], we observe that the accuracy increases but is lower than that of the fine-tuned model. Thus, we used the fine-tuned T5-base model for lifted translation in our system. For cost effective reasons, we averaged the RAG results over two seeds per city.

C. Full System Evaluation and Ablation Study

We tested the overall performance of our language grounding system that takes a spatiotemporal navigation command as input and produces an LTL formula whose propositions are grounded to physical landmarks in the environment. To evaluate the effectiveness of multimodal semantic information for language grounding, we conducted an ablate study where we only used one modality, i.e., text or images, in the referring expression grounding (REG) module.

The full system using both modalities achieved an accuracy of 93.53%. As shown in Figure 6, it significantly outperformed the image-only system because images alone often did not provide enough information, especially when

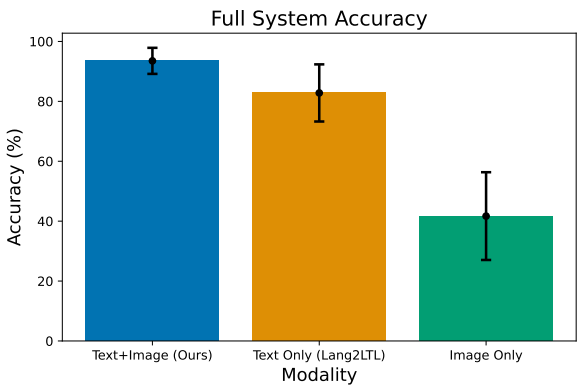


Fig. 6: This graph depicts the average accuracies of spatiotemporal language grounding systems using different modalities across four environments and five seeds per environment.

there were distractor objects with similar visual features. It outperformed the text-only system by more than 10%. The margin was much smaller than that with the image-only system because our full system essentially used textual description to ground REs after captioning images with text. Still, the additional visual features provided by images can further disambiguate similar landmarks. For example, colors can help disambiguate a red and a yellow bicycle. In reality, detailed textual descriptions of landmarks are not always available, e.g., "the red brick building," but can be easily extracted from images by querying a pretrained VLM for image captions. The accuracy of the spatial predicate grounding (SPG) module when given the top-10 candidate groundings from the referring expression grounding (REG) module was $97.26 \pm 2.07\%$. It supports our hypothesis that if the correct RE grounding is among the top candidates of REG, SPG can identify the correct figure landmark based on spatial reasoning. Note that the text-only system is the same as Lang2LTL [8]. Liu et al. [8] showed that Lang2LTL outperforms Code-as-Policies [43], a prominent system that grounds natural language instructions to Python code directly executable on robots.

VI. Robot Demonstration

To demonstrate Lang2LTL-2's ability to inform an automated planner and enable the execution of spatiotemporal commands, we deploy the same system without modification at the task planning level on a quadruped robot Spot [44] in an indoor and outdoor environment. These environments contain nine and five objects, respectively, with multiple objects and landmarks that have similar textual or visual features, e.g., tables, couches, buildings, dumpsters, and cars.

We use Spot's GraphNav software to build a semantic map of the environment and capture images of landmarks and objects of interest. We only use the image modality for indoor experiments. For the outdoor environment, we additionally download textual descriptions of landmarks in the region from OpenStreetMap [41]. Given a grounded LTL task specification output by our language grounding system Lang2LTL-2, we use the AP-MDP planner [12] to produce a sequence of locations through the semantic map. We executed 50 semantically diverse spatiotemporal natural language commands on the physical robot. With the formal safety guarantee offered by AP-MDP planner for LTL task specification, the robot was able to abort the execution when a given task was infeasible. Please see supplementary materials for the list of all the commands.

VII. Related Work

A. Grounding Spatial Commands for Robots

SLOOP [3] is a system that grounds spatial commands in partially observable environments by using the spatial relations between a target object and multiple landmarks to construct an initial belief for a POMDP planner. LanguageRefer [45] is a learned transformer-based model that takes as inputs a spatial language command, a 3D point cloud of the scene, and bounding boxes of objects, then predicts the target object. RoboHop [46] builds a topological map of the environment with image segments as nodes. Like our work, RoboHop uses an LLM to extract referring expressions (REs) from a language command. Then it uses a VLM to ground REs to nodes in the topological map.

B. Grounding Temporal Commands for Robots

Linear temporal logic (LTL) [27] is a mathematically precise language that can specify robotic tasks and provide satisfaction guarantees, especially for long-horizon, temporally-extended tasks. Early work of using LTL for temporal command grounding was limited to structured language [47]. Gopalan et al. [7] trained a Seq2Seq network [48] on natural language and LTL pairs in every new environment to ground language commands for navigation and manipulation. Like our work, Berg et al. [28] and Hsiung et al. [49] first translated commands to lifted LTL formulas then grounded the propositions to landmarks or objects but used a Seq2Seq network with limited capabilities.

To mitigate the lack of training data, Pan et al. [50] used an LLM to paraphrase structured language commands constructed from algorithmically generated LTL formulas. Patel et al. [51] and Wang et al. [52] proposed weakly supervised methods that use executed trajectories instead of LTL annotations to guide language grounding. Similarly, Lang2LTL [8] is a modular system that uses LLMs to ground temporally extended navigation commands in indoor and outdoor environments without retraining, given a text-based semantic database. However, Lang2LTL cannot ground spatial referring expressions or landmarks with visual descriptions. Our system improves upon Lang2LTL by incorporating spatial reasoning and using a vision-language model (VLM) to process images.

C. Grounding Spatiotemporal Commands for Robots

Language commands from existing works of indoor [1, 2, 3, 53] and outdoor [4, 5] navigation are rich in spatial

relations, but lack diverse temporal patterns. Lang2LTL-2 considers language commands containing 15 temporal patterns commonly used in robotics [6]. LM-Nav [5] uses an LLM to extract a sequence of referring expressions (REs) from a navigation command, then a VLM to ground the REs to images of physical landmarks. LM-Nav only grounds language commands of sequenced visit type defined in [6]. VLMaps [54] fuses pretrained vision-language features with depth information to construct a spatial map of the environment then directly indices a sequence of spatial referring expressions (SREs) extracted by an LLM in the map. LIMP [53] uses RGB-D information, an LLM and a VLM to construct a 3D semantic map conditioned on the input language for motion planning to solve indoor mobile manipulation tasks. It translates free-form language commands to one of three temporal patterns using an LLM. An additional advantage of our system is its ability to ground REs that are not easily represented by visual description, e.g., generic referring expressions like "the vegan restaurant," and proper names like "Wildflour" (the name of a bakery) by using additional textual description from OpenStreetMap [41] in grounding city-scaled navigation commands.

VIII. Conclusion

We propose a modular system that consists of pretrained large language models and a pretrained vision-language model to ground spatiotemporal navigation commands to landmarks described by text and images in a semantic map of novel indoor and outdoor environments. We evaluate the individual modules and the full language grounding system on a semantically diverse dataset of 21,780 spatiotemporal navigation commands in novel city-scaled environments. Our system achieved 93.53% accuracy, outperforming the previous SoTA. An autonomous robot with access to a semantic map and position tracking can use the same system without modification to follow spatiotemporal navigation commands in novel indoor and outdoor environments. We envision incorporating interaction with human users to further improve spatiotemporal language grounding.

ACKNOWLEDGMENT

The authors thank Nihal Nayak, Rao Fu, James Tompkin, and Peilin Yu for discussions of vision-language models, Ugur Çetintemel for discussions of vector databases, and Mingxi Jia for editing the robot demonstration videos. This work is supported by ONR under grant award number N00014-22-1-2592 and funding from Amazon Robotics.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3674–3683.

[2] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *Conference on Empirical Methods for Natural Language Processing*, 2020.

[3] K. Zheng, D. Bayazit, R. Mathew, E. Pavlick, and S. Tellex, "Spatial language understanding for object search in partially observed city-scale environments," in *IEEE International Conference on Robot & Human Interactive Communication*, 2021, pp. 315–322.

[4] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 538–12 547.

[5] D. Shah, B. Osiński, S. Levine *et al.*, "LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*. PMLR, 2023, pp. 492–504.

[6] C. Menghi, C. Tsigkanos, P. Pelliccione, C. Ghezzi, and T. Berger, "Specification patterns for robotic missions," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2208–2224, oct 2021.

[7] N. Gopalan, D. Arumugam, L. L. Wong, and S. Tellex, "Sequence-to-sequence language grounding of non-markovian task specifications," in *Robotics: Science and Systems*, 2018.

[8] J. X. Liu, Z. Yang, I. Idrees, S. Liang, B. Schornstein, S. Tellex, and A. Shah, "Lang2LTL: Grounding complex natural language commands for temporal tasks in unseen environments," in *Conference on Robot Learning*. PMLR, 2023, pp. 1084–1110.

[9] Y. Chen, R. Gandhi, Y. Zhang, and C. Fan, "NL2TL: Transforming natural languages to temporal logics using large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 15 880–15 903.

[10] M. L. Littman, U. Topcu, J. Fu, C. Isbell, M. Wen, and J. MacGlashan, "Environment-independent task specifications via GLTL," *arXiv preprint arXiv:1704.04341*, 2017.

[11] A. Camacho, R. T. Icarte, T. Q. Klassen, R. A. Valenzano, and S. A. McIlraith, "LTL and beyond: Formal languages for reward function specification in reinforcement learning," in *IJCAI*, vol. 19, 2019, pp. 6065–6073.

[12] Y. Oh, R. Patel, T. Nguyen, B. Huang, E. Pavlick, and S. Tellex, "Planning with state abstractions for non-markovian task specifications," in *Robotics: Science and Systems*, vol. 2019, 2019.

[13] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, "Reward machines: Exploiting reward function structure in reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 73, pp. 173–208, 2022.

[14] J. X. Liu, A. Shah, E. Rosen, M. Jia, G. Konidaris, and S. Tellex, "LTL-Transfer: Skill transfer for temporally-extended task specifications," *IEEE International Conference on Robotics and Automation*, 2024.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[17] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023, accessed the model on July 11, 2024.

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[20] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024.

[21] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7086–7096.

[22] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple open-vocabulary object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 728–755.

[23] J. Chen, H. Guo, K. Yi, B. Li, and M. Elhoseiny, "VisualGPT: Data-efficient adaptation of pretrained language models for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 030–18 040.

[24] Z. Liu, C. Rodriguez-Opazo, D. Teney, and S. Gould, "Image retrieval on real-life images with pre-trained vision-and-language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2125–2134.

[25] Y. Du, J. Li, T. Tang, W. X. Zhao, and J.-R. Wen, "Zero-shot visual question answering with language model feedback," in *Findings of the Association for Computational Linguistics*. Association for Computational Linguistics, 2023, pp. 9268–9281.

[26] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, "The dawn of LMMs: Preliminary explorations with GPT-4V(ision)," *arXiv preprint arXiv:2309.17421*, 2023.

[27] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (SFCS)*. ieee, 1977, pp. 46–57.

[28] M. Berg, D. Bayazit, R. Mathew, A. Rotter-Aboyoun, E. Pavlick, and S. Tellex, "Grounding language to landmarks in arbitrary outdoor environments," in *2020 IEEE International Conference on Robotics and Automation*. IEEE, 2020, pp. 208–215.

[29] A. Shah, S. Li, and J. Shah, "Planning with uncertain specifications (PUnS)," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3414–3421, 2020.

[30] A. Shah, P. Kamath, S. Li, P. Craven, K. Landers, K. Oden, and J. Shah, "Supervised bayesian specification inference from demonstrations," *The International Journal of Robotics Research*, vol. 42, no. 14, pp. 1245–1264, 2023.

[31] R. T. Icarte, T. Klassen, R. Valenzano, and S. McIlraith, "Using reward machines for high-level task specification and decomposition in reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2107–2116.

[32] M. Y. Vardi, "An automata-theoretic approach to linear temporal logic," *Logics for Concurrency*, pp. 238–266, 1996.

[33] R. Gerth, D. Peled, M. Y. Vardi, and P. Wolper, "Simple on-the-fly automatic verification of linear temporal logic," in *Protocol Specification, Testing and Verification XV: Proceedings of the Fifteenth IFIP WG6. 1 International Symposium on Protocol Specification, Testing and Verification*. Springer, 1996, pp. 3–18.

[34] G. De Giacomo, L. Iocchi, M. Favorito, and F. Patrizi, "Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, 2019, pp. 128–136.

[35] J. Lyons, *Semantics: Volume 2*. Cambridge University Press, 1977, vol. 2.

[36] B. Landau and R. Jackendoff, ""what" and "where" in spatial language and spatial cognition," *Behavioral and Brain Sciences*, vol. 16, pp. 217–265, 06 1993.

[37] V. W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Y. Zou, "Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 612–17 625, 2022.

[38] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[39] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[40] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[41] O. Contributors, "Planet OSM," https://www.openstreetmap.org, 2017.

[42] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google Street View: Capturing the world at street level," *Computer*, vol. 43, no. 6, pp. 32–38, 2010.

[43] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *IEEE International Conference on Robotics and Automation*, 2023.

[44] Boston Dynamics, "Spot® - the agile mobile robot," https://www.bostondynamics.com/products/spot.

[45] J. Roh, K. Desingh, A. Farhadi, and D. Fox, "LanguageRefer: Spatial-language model for 3d visual grounding," in *Conference on Robot Learning*. PMLR, 2022, pp. 1046–1056.

[46] S. Garg, K. Rana, M. Hosseinzadeh, L. Mares, N. Suenderhauf, F. Dayoub, and I. Reid, "RoboHop: Segment-based topological map representation for open-world visual navigation," in *IEEE International Conference on Robotics and Automation*. IEEE, 2024.

[47] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "From structured english to robot motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2007, pp. 2717–2722.

[48] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[49] E. Hsiung, H. Mehta, J. Chu, J. X. Liu, R. Patel, S. Tellex, and G. Konidaris, "Generalizing to new domains by mapping natural language to lifted LTL," in *International Conference on Robotics and Automation*. IEEE, 2022, pp. 3624–3630.

[50] J. Pan, G. Chou, and D. Berenson, "Data-efficient learning of natural language to linear temporal logic translators for robot task specification," in *IEEE International Conference on Robotics and Automation*. IEEE, 2023.

[51] R. Patel, E. Pavlick, and S. Tellex, "Grounding language to non-markovian tasks with no supervision of task specifications," in *Robotics: Science and Systems*, vol. 2020, 2020.

[52] C. Wang, C. Ross, Y.-L. Kuo, B. Katz, and A. Barbu, "Learning a natural-language to LTL executable semantic parser for grounded robotics," in *Conference on Robot Learning*. PMLR, 2021, pp. 1706–1718.

[53] B. Quartey, E. Rosen, S. Tellex, and G. Konidaris, "Verifiably following complex robot instructions with foundation models," *arXiv preprint arXiv:2402.11498*, 2024.

[54] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *IEEE International Conference on Robotics and Automation*. IEEE, 2023, pp. 10 608–10 615.