Multi-agent Planning using Visual Language Models

Michele Brienza[*,1], **Francesco Argenziano**[*,1], **Vincenzo Suriani**[†,1], **Domenico D. Bloisi**[§,1] **and Daniele Nardi**[*]

[*]Sapienza University of Rome, Rome RM 00181, Italy
[†]University of Basilicata, Potenza PZ 85100, Italy
[§]International University of Rome UNINT, Rome RM 00147, Italy
ORCID (Michele Brienza): https://orcid.org/0009-0000-1549-9500, ORCID (Francesco Argenziano): https://orcid.org/0009-0004-2028-7253, ORCID (Vincenzo Suriani): https://orcid.org/0000-0003-1199-8358, ORCID (Domenico D. Bloisi): https://orcid.org/0000-0003-0339-8651, ORCID (Daniele Nardi): https://orcid.org/0000-0001-6606-200X

Abstract. Large Language Models (LLMs) and Visual Language Models (VLMs) are attracting increasing interest due to their improving performance and applications across various domains and tasks. However, LLMs and VLMs can produce erroneous results, especially when a deep understanding of the problem domain is required. For instance, when planning and perception are needed simultaneously, these models often struggle because of difficulties in merging multimodal information. To address this issue, fine-tuned models are typically employed and trained on specialized data structures representing the environment. This approach has limited effectiveness, as it can overly complicate the context for processing. In this paper, we propose a multi-agent architecture for embodied task planning that operates without the need for specific data structures as input. Instead, it uses a single image of the environment, handling free-form domains by leveraging commonsense knowledge. We also introduce a novel, fully automatic evaluation procedure, PG2S, designed to better assess the quality of a plan. We validated our approach using the widely recognized ALFRED dataset, comparing PG2S to the existing KAS metric to further evaluate the quality of the generated plans.

1 Introduction

Foundation Models (FMs) are machine learning models that are trained on a broad (*Internet-scale*) amount of data and can be refined to be used in a wide range of downstream applications [6]. Initial examples of these models, i.e., Large Language Models (LLMs) [9, 7, 1, 31], were inherently of the Natural Language Processing (NLP) field. Nevertheless, in the last years, we have witnessed the emergence of multi-modal LLMs, which can handle non-textual inputs and outputs. Visual Language Models (VLMs) [16, 22] have particular relevance in this category since they can take as input images and/or textual queries and generate contextual high-quality outputs. Additionally, the birth of many toolkits like HuggingFace [35] or LangChain [5] have contributed to the outburst and the distribution of such models, widening their domain of applications.

It has been demonstrated that LLMs can be used as zero-shot [12] and few-shot [28] planners. This is due to the fact that these models have been trained on huge amounts of data, therefore they incorporate the commonsense knowledge proper of humans [14].

[1] Corresponding Author. Email: {brienza, argenziano}@diag.uniroma1.it, vincenzo.suriani@unibas.it, domenico.bloisi@unint.eu

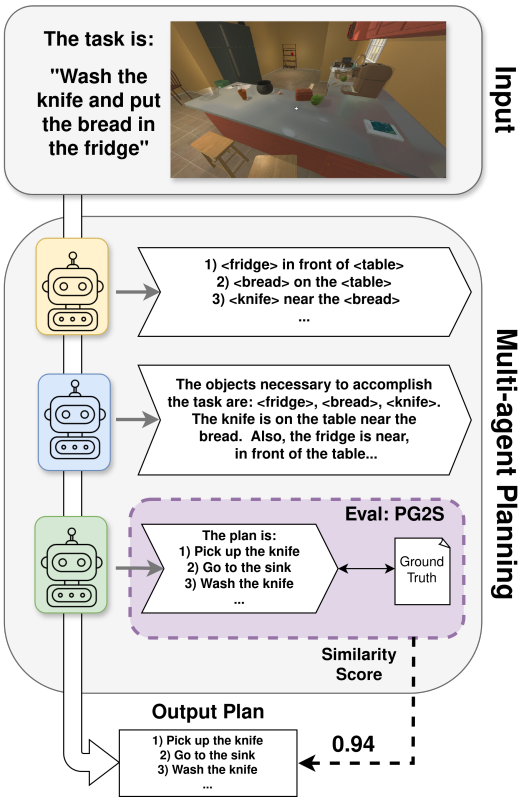

Figure 1. Overall view of the proposed framework. Given a task description and an image of the scene, the plan is obtained with multi-agent planning and assessed with the new score.

An agent with commonsense knowledge acquires complex reasoning capabilities via chain-of-thought [34] and it becomes able to correctly generate a plan to achieve the desired goal. The generated plans are grounded in the sense that actions, objects, and states all refer to the specific environment the embodied agent is deliberating in, thanks to the information incorporated in the queries. Existing solutions for grounding concern encoding the environment in a structured manner, i.e., using tables or graphs [15, 23], since they are easily promptable to the model once converted in some sort of streamable

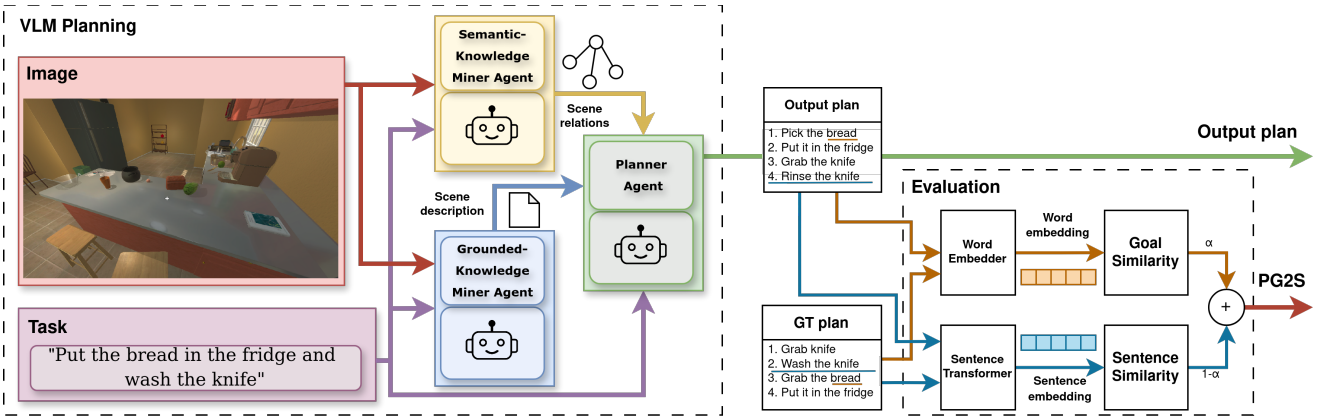

Figure 2. Complete and detailed architecture of the proposed method. The task description and the image are given in input to the agents that extract meaningful information from the scene. Their output is then processed by the planner agent that obtain the final plan. Such plan is then compared with the ground truth and evaluated according our new metric that takes into account semantically meaningful information.

format. However, these kinds of representations grow very quickly as the environment grows in size, thus it becomes difficult to incorporate them in a language model query prompt. The context window, namely the amount of text the model can handle as input when generating language, becomes very big as the prompt increases, and the output could be affected by several hallucinations [17]. This is a problem in any LLMs application, but particularly when we are trying to plan a specific procedure to achieve a certain goal. For this reason, it is important to keep the input to the model as small as possible, including only the necessary information to carry out the desired task. In fact, decomposing the goal into several sub-goals for multiple independent agents can drastically improve the final output, thus performing better w.r.t. single-agent architectures [33, 30, 20].

In this paper, we analyze the capabilities of FMs when used as reasoning components. In particular, we use FMs to deploy grounded plans for embodied agents in free-form domains, i.e. domains that come without a structured representation. All we need is a picture of the scene that captures the most relevant aspects of the environment, and a textual query on the goal we want to achieve. Our approach exploits a hierarchical multi-agent structure, in the sense that every agent is a different VLM/LLM instance which addresses to solve only one aspect of the whole planning procedure, according to the definition given by [30]. In this way, every agent has a limited context window, thus being less prone to hallucinations.

We conduct a comparative study to demonstrate that our single-image multi-agent scheme outperforms both an architecture utilizing a structured environment representation (such as tabular) and a single-agent architecture where all input is fed to the VLM simultaneously.

We also propose *Planning Goal Semantic Score* (PG2S), a new metric that does not rely on user validations to evaluate the results. PG2S does not consider the partial ordering of actions needed to carry out the plan to achieve a goal, and it is semantically sound in the sense that it deals with synonyms without losing the meaning of the plan. In Fig. 1 it is possible to see an overall view of our system. In summary, the contribution of this work is three-fold:

- We demonstrate that it is possible to soundly plan to achieve a task with a VLM and a query input, relaxing the assumptions about complex structured query;
- We propose a multi-agent framework to decompose the final task into different sub-tasks, thus reducing the risk of hallucinations and other harmful phenomena;
- We introduce a new metric, *PG2S*, to autonomously evaluate the correctness of a plan expressed in natural language that is partial-ordering agnostic and semantically aware.

We validate our approach on ALFRED [27], a benchmark purposefully designed to evaluate natural language instructions mapping in household environments, built upon the AI2-THOR framework [13]. For comparison, we use G-PlanET [15], which contains all the simulation environments of ALFRED but represented in a tabular form. We release the code, the prompts used and the results obtained on our project website.[2]

2 Related Work

In this section, we discuss existing solutions about both using LLMs for planning and adopting a multi-agent architecture for prompting.

2.1 *LLMs as Planners*

A pioneering work that exploits the use of LLMs for embodied agents is SayCan [2], where a robot can behave as "hands and eyes" for an LLM when grounding tasks in real-world scenarios, taking advantage of the semantic knowledge of the model when performing complex instructions. Following this research, several approaches started to emerge that tried to use LLMs as the planning component in many different use cases.

Huang et al. [12] demonstrate that LLMs behave like zero-shot planners when they are correctly prompted. In contrast, Song et al. [28] show that tuning these models in a few-shot setting, allows them to surpass state-of-the-art Vision Language Navigation (VLN) models even if they are trained on a broader amount of data, thanks to LLMs' embedded commonsense knowledge.

LLMs' capabilities change when the query in input is not completely textual, but can assume a more structured form, e.g., a tabular structure [15], a graph-like structure [23] (such as 3D Scene Graphs [4]), or even LTL formulas [8]. Incorporating this additional information is useful to improve the overall performance in the desired tasks. However, the biggest drawback of these techniques is that they require a very high computational cost when applied to real-world scenarios, where the environment is unstructured.

[2] https://lab-rococo-sapienza.github.io/map-vlm/

New research directions were possible thanks to advancements in VLMs, such as the possibility of directly processing visual queries given to embodied agents. Allowing systems to take in input images of the environment can solve the problem of creating complex structured representations, saving time and computations while still maintaining the reasoning power of LLMs. For example, [10] shows that it was possible to use VLMs to find objects in the environment that are described by natural language descriptions given by humans (like a "cat-shaped mug").

However, choosing which VLM to use is not a trivial task. Many VLMs are built upon CLIP [21], but these models present *bags-of-words* behaviors [38], namely they ignore semantic structures of inputs, such as spatial relationships between objects. To cope with this problem, LLM-Grounder [37] shows that it was indeed possible to exploit the power of VLMs to plan for embodied agents while reducing the effects of the *bags-of-words* phenomenon. In our approach, we adopt VLMs to get rid of complex structured inputs, but at the same time, we reduce as much as possible the *bags-of-words* behavior by using a multi-agent approach. Our approach is able to decompose the task into sub-tasks in such a way that potentially problematic relations are handled by a specific agent.

2.2 Multi-agent Prompting

As LLMs became more and more diffused, it was discovered that specific prompting patterns produced better results than free-form prompts (*prompt engineering*) [39]. In planning applications, *chain-of-thought* reasoning [34] has marked a notable advance, with multi-step reasoning.

Another important step in prompt engineering with LLMs is achieved by leveraging the power of multi-agent systems. In [30], a collaborative environment where multiple agents with different roles had to work together to accomplish a task, is demonstrated to have better performance w.r.t. a single-agent. Moreover, results improved not only in settings with many role-specific agents but also in settings with multi-persona self-collaborating agents[33].

Several frameworks started to emerge, simplifying the development of multi-agent applications [36, 24, 26]. As a drawback, these frameworks intrinsically increase the complexity of the systems that adopt them.

3 Methodology

The typical interaction between an LLM and a user consists of a trial-and-error process to obtain the desired result by refining the prompt. The accuracy of the environmental information is crucial to obtain a correct plan. Usually, this information comes from tables or structured data. Our method is based on relaxing the structured information known a priori from a previous labeling process. In our architecture, we use a multi-agent pipeline that takes as input only an image of the environment, along with the task to execute. Then, we show how this strategy allows us to have a correct plan, even in free-form domains. To assess the correctness, we use our PG2S metric by comparing the plans devised from images and those from tables by referring to ALFRED's annotations.

3.1 Multi-agent Planning

Our solution employs three agents, each representing a phase in the planning generation process: the *Semantic-Knowledge Miner Agent* (SKM), the *Grounded-Knowledge Miner Agent* (GKM), and the *Planner Agent* (*P*). GPT-4V is used for agents that process images, while GPT-4 is used for the planning agent [1].

The SKM Agent identifies object classes within the image and establishes the scene's ontology. It also determines relationships between objects, creating a knowledge graph. The GKM Agent grounds these objects, providing short descriptions that include their relationships with surrounding objects, resulting in a high-level yet structurally sound scene description. The *P* Agent then generates a plan using the information from the SKM and GKM Agents. This method minimizes hallucinations and focuses the plan on the relevant objects in the scene.

Using a Visual Language Model (VLM), we achieve better results with a multi-agent strategy compared to a single-agent approach. In a single-agent setup, the prompt directs the VLM to create a plan from the input image. In contrast, the multi-agent setup allows the Miner Agents to enrich the Planning Agent's knowledge with detailed environmental information, as illustrated in Fig. 2.

The multi-agent strategy enhances plan quality by distributing the workload among agents, each handling specific tasks. This division reduces the risk of hallucinations by maintaining smaller, more focused prompts within each agent's context window [17]. By splitting the task into simpler sub-tasks, our pipeline ensures more accurate and coherent responses, following the "divide and conquer" principle.

3.2 Evaluation

Choosing an adequate metric to evaluate the quality of produced plans is not trivial. Usually, only the Success Rate (SR) or the SR weighted by the inverse path length (SRL) are used to evaluate the plan correctness [28, 10]. However, these metrics are not very convenient to compute, and researchers often rely on Amazon Mechanical Turk to check the correctness using human experts. Moreover, they do not evaluate the quality of the plan: they state how many times the goal is achieved and how the length of the plan influences the result.

G-PlanET [15] tries to define a new metric to cope with this problem: inspired by metrics used for semantic captioning like CIDEr [32] and SPLICE [3], it proposes KeyActionScore (KAS). KAS builds a set of key action phrases obtained from every step of the generated plan $\hat{S}_i$, and from the reference plan of the dataset S_i. Then, by checking how many action phrases in $\hat{S}_i$ are covered by S_i, and by computing this precision, it is possible to evaluate the matching quality of the two sets for the i-th step of the plan.

This metric present two main limitations. The first is that it always assumes that the reference plan is correct, which is not always true as we found some examples in the ALFRED dataset of plans that are not completely correct: e.g. the reference plan for the goal *"Put a hot bread in the refrigerator"* has as one of the steps the action *"put the knife in the microwave"* which is extremely dangerous and globally incorrect for the desired goal. The second is that in definition of KAS, a mapping is considered correct *if and only if* it follows the order of actions given by the step. This is a strong assumption, since there are many plans in which the order of actions is not necessary to reach a goal [19], so it can penalize plans that are actually correct.

To this end, we propose a new metric, **PG2S**, that copes with this problem. As an example, we show a reference plan that can be used as a ground truth plan and a possible predicted plan (see Table 1). The predicted plan to reach the goal *"Wear a pair of shoes"* is correct for a human evaluator. Despite this, the plan is different from the ground truth in the order of the actions, and the evaluation should be able to take into account this possibility. Using the KAS metric the similar-

Table 1. Example of partial-ordering of actions.

Ground truth plan:	**Predicted plan:**
Carefully wear the left sock.	Wear the right sock with care.
Wear the right sock.	Carefully wear the right shoe.
Put on the snug left shoe carefully.	Gently wear the left sock first.
Slip into the right shoe comfortably.	Slide into the left shoe carefully.

Algorithm 1 PG2S Evaluation Procedure

Require: $\mathcal{P}_{gt}$ ground truth plan, $\mathcal{P}_{pred}$ predicted plan
Ensure: $PG2S$
1: $MaxSimilPlan, MaxSimilGoal \leftarrow [\,]$
2: **for** $s_i \in \mathcal{P}_{gt}$ **do**
3: find the most similar sentence s_j in $\mathcal{P}_{pred}$
4: **if** exists: add 1 to $MaxSimilPlan$; **otherwise** add 0
5: $\mathcal{P}_{pred}$.pop(s_j)
6: $S_{plan} \leftarrow$ mean($MaxSimilPlan$)
7: $A_{gt}, A_{pred} \leftarrow [\,]$
8: **for** $s_i, s_j \in \mathcal{P}_{gt}, \mathcal{P}_{pred}$ **do**
9: add actions in A_{gt} and A_{pred} with $Framing()$
10: **for** $a_i \in A_{gt}$ **do**
11: find the most similar action a_j in $\mathcal{A}_{pred}$
12: **if** exists: add 1 to $MaxSimilGoal$; **otherwise** add 0
13: $\mathcal{A}_{pred}$.pop(a_j)
14: $S_{goal} \leftarrow$ mean($MaxSimilGoal$)
15: $PG2S \leftarrow \alpha * S_{plan} + (1-\alpha) * S_{goal}$

ity score is equal to 0.33; while for PG2S (ours) the similarity score obtained is equal to 0.83. Algorithm 1 presents the procedure used to compute such an evaluation score. More in detail, given two sets of planning descriptions, $\mathcal{P}_{gt}$ and $\mathcal{P}_{pred}$, respectively the ground truth plan and the predicted plan, we aim at quantifying their similarity, using two levels of evaluation, namely a *sentence-wise* and a *goal-wise*, both based on the semantic values. To determine if two embeddings are similar we use a threshold mechanism. In particular, we adopt the approach presented in [25], where the authors obtain thresholds that vary according to the dimensionality of the embedding vector and verify that their use allows to obtain only semantically similar elements.

Sentence-wise similarity. To compute the sentence similarity, we deploy embedding vector representations for each sentence using a *Sentence Transformer*. In particular, we use MPNet [29], which achieves better results in semantic evaluation tasks compared with previous state-of-the-art pre-trained models [29] (e.g., BERT, XLNet, and RoBERTa). For each sentence $s_i \in \mathcal{P}_{gt}$ and $s_j \in \mathcal{P}_{pred}$, we obtain the similarity between their embeddings (v_i and v_j) using the cosine similarity $cos(v_i, v_j)$. For each s_i, we identify the most similar sentence in $\mathcal{P}_{pred}$ (line 3) and remove it from the set (line 5). The value of each similarity yields a list of maximum similarity scores. The sentence-wise similarity is the average of these scores (line 6).

$$S_{\text{plan}}(\mathcal{P}_{gt}, \mathcal{P}_{pred}) = \frac{1}{N}\sum_{i=1}^{N} \text{MaxSimilPlan}_i \tag{1}$$

Goal-wise similarity. To compute the goal similarity, we first perform a POS tagging pre-processing stage using *spaCy* [11], and then, for each sentence we extract the main action using a *Framing()* procedure (line 9). This procedure works as follows: for each word in a sentence, we add it in the action set if it is either *i.* a central ('root') verb (VERB), or *ii.* if it is a noun (NOUN) and its dependency tag is either a 'direct object' (DOBJ) or the 'nominal subject' (NSUBJ). In this way, for each step we obtain the main action and the involved objects. For each action $a_i \in \mathcal{A}_{gt}$ and $a_j \in \mathcal{A}_{pred}$, we obtain a similarity value from the product between the mean of nouns similarity and the verbs similarity, obtained from a *WordEmbeddingSimilarity()* tool (Word2Vec [18]).We consider two nouns and two verbs to be similar if their similarity value exceeds a threshold $\tau = 0.708$, according to [25].

For each action a_i in $\mathcal{A}_{gt}$ we identify the most similar action in $\mathcal{A}_{pred}$ and remove it from the set. The most similar action is found using the combined similarity computed with the product of both values (line 11) and removed from $\mathcal{A}_{pred}$ (line 13). The value of each action similarity yields a list of maximum similarity scores. The average of these scores gives us the goal-wise similarity of the sets (line 14).

$$S_{\text{goal}}(\mathcal{A}_{gt}, \mathcal{A}_{pred}) = \frac{1}{N}\sum_{i=1}^{N} \text{MaxSimilGoal}_i \tag{2}$$

PG2S. The final similarity score is our metric PG2S, which is a weighted average of the sentence-level and action-state similarities, where α is a weighting factor, set to 0.5 to equally balance the contributions of the two scores:

$$PG2S = (1-\alpha) * S_{\text{plan}}(\mathcal{P}_{gt}, \mathcal{P}_{pred}) + \alpha * S_{\text{goal}}(\mathcal{A}_{gt}, \mathcal{A}_{pred}) \tag{3}$$

Another issue arises because KAS employs a set intersection, whereby terms that are not equal are not considered for the similarity calculation. This can result in the problem of having the same action with a subject that is not appropriate for use in the case of goal similarity. To illustrate this aspect, consider the action "Walk to the desk" in comparison to "Walk to the moon". In the case of KAS, the resulting similarity score is 0.67 because two out of three elements are equal, whereas in PG2S, the similarity score is 0. This discrepancy can be attributed to the fact that KAS does not consider the nuances of natural language, whereas PG2S does.

4 Experimental Results

This section presents the outcomes of the conducted experiments, which were designed to test the proposed architecture's validity. The results obtained using a single image are presented and then compared with a structured perception of the environment, as seen in state-of-the-art works. The output plans regarding home scenarios tasks are taken from the ALFRED dataset using the AI2Thor environment. Chosen the image and the environment, for each of those we have found the plan associated with the scene and saved the ground truth plans that we have used to compare our results. The environment scenarios are chosen by selecting several different situations in order to have various complexity and domains of application according to the chosen fields of ALFRED such as: picking up objects and placing them; picking up objects, heating or cooling them, and place them somewhere else; cleaning objects and examining under the light; and more.

4.1 Evaluation of our PG2S Metric

During the experimental phase of PG2S development, a series of tests were conducted to ensure the correctness of the metric. Specifically, we compared ALFRED plans with those predicted by our architecture, together with their corrupted version. During the test phase, several examples were selected from the ALFRED dataset. The plans obtained were checked qualitatively and it was possible to verify that the plans generated by the multi-agent architecture are correct

in terms of the sequence of actions. The corrupted plans ensure that the goal similarity will be respected and will not return a high level of similarity in case of ambiguity.

An example of ground truth plan taken from the ALFRED dataset for the task "cool the tomato in the refrigerator, put it by the toaster" is the following:

- `Turn right, walk across the room to the counter on the right side of the stove.`
- `Grab the tomato on the counter.`
- `Turn around, walk to the refrigerator.`
- `Put the tomato in, close the refrigerator, wait a while, open the refrigerator, take the tomato out, close the refrigerator.`
- `Turn around, walk to the stove.`
- `Put the tomato down on the counter in front of the toaster.`

The corresponding predicted plan by our architectur for the same task is:

- `Walk to the kitchen island, which is in front of the kitchen counter.`
- `Pick up the tomato, which is on the kitchen island.`
- `Walk to the refrigerator, which is to the left of the kitchen island.`
- `Open the refrigerator door.`
- `Place the tomato inside the refrigerator.`
- `Close the refrigerator door.`
- `Wait for the tomato to cool.`
- `Open the refrigerator door.`
- `Take the cooled tomato out of the refrigerator.`
- `Close the refrigerator door.`
- `Walk to the kitchen counter, which is to the right of the refrigerator.`
- `Place the cooled tomato by the toaster, which is on the kitchen counter.`

The generated plan and the ground truth plan, compared with both metrics, obtain a 50% percentage of similarity with PG2S compared to a 30% percentage by KAS. The most notable distinction between the two plans is the navigation sequence required to reach the object in question. In the ground truth plan, the kitchen objects are reached differently than the predicted plan. Additionally, there is a notable difference in the number of actions required to cool a tomato in the refrigerator. In the predicted plan, there are seven steps, while in the ground truth, there is only one. Then, the predicted plan was corrupted by replacing *kitchen* $\rightarrow$ *bathroom* and *tomato* $\rightarrow$ *bottle*, and we verified the drop of similarity from 50% to 25% in PG2S, while KAS decreased from 30% to 26%. In this case, the PG2S similarity is halved, while in KAS the modification of the plan does not affect much. The step-by-step comparison performed by KAS does not allow an analysis of whether the goals are carried out during the planning; moreover, the cross-comparison of words loses sight of the semantic content. Table 2 shows how the similarity score drops after corrupting the plan by changing the objects due to the PG2S semantic search. The plan available presents an id "trail_ID" associated with a ground truth plan and a goal that can be found in the G-Planet dataset.[3] The metric does not evaluate the success score of the plan but compares the steps with a semantic evaluation. Table 2 also illustrates the scores obtained by comparing the plans obtained with KAS and PG2S before corruption. In each case, the degree of similarity obtained is superior to that of KAS. Both metrics are used in the following section to evaluate the plan correction.

[3] https://huggingface.co/datasets/yuchenlin/G-PlanET/viewer/default

Figure 3. One of the scenes used for the experimental tests. A screenshot from AI2Thor is used to perform the planning.

4.2 Evaluation of our Architecture

To evaluate the presented methodology, we have chosen ten different rooms of an apartment, such as a living room, a kitchen, and a toilet. Frames were captured for each room as in the example in Fig. 3 which depicts a kitchen. The complexity of generating a plan is evident, given that an entire scene is represented by a single image and that some of the objects needed can be quite small. Our tests demonstrated that even in complex situations, the VLM is capable of identifying objects and perceiving their relationships, allowing it to define a correct plan. The ten environments chosen allow us to obtain thirty tasks to perform and, for each of these plans, we have obtained the plan using four approaches: two using a single-agent architecture and two using a multi-agent architecture. In both single-agent and multi-agent evaluations, the plan was obtained using a table describing the environment rather than a single image.

Table 3 presents the results, highlighting instances where the KAS metric fails, resulting in *None* values. This failure occurs because the KAS metric cannot evaluate plans of different lengths, which was common in the "with table" setups.

The results show how using a single image the architecture generates a plan similar to the ground truth plan. Furthermore, we demonstrated to obtain improved results in multi-agent architecture using a single image.

5 Discussion

The current state of the art involves the use of traditional Success Rate metrics to evaluate a plan, where the plan is considered correct in cases where execution leads to the desired outcome. However, this metric is not sufficient or suitable for all cases where the correctness of a task execution plan is to be analyzed. In particular, in cases where the plan is complicated, it should be evaluated before execution to avoid damage to the environment or simply unsuccessful executions and ensure that time and resources are not wasted in a new execution. The advent of LLMs has made it possible to easily generate plans that previously required model training or other more complex techniques. Given that these models can' hallucinate' or generate incorrect responses, there could be errors present. Therefore, these inaccuracies could lead to failures when evaluating them based on success rates. Our work seeks to define a new PG2S metric for plan evaluation based only on natural language processing

Table 2. Similarity values are calculated between predicted plans and ALFRED-annotated plans, where the predicted plans may be corrupted by substituting object names in the steps. For corrupted plans, lower similarity values are preferred because they indicate a greater difference from the ground truth plan. Conversely, for uncorrupted plans, higher similarity values are desirable.

Corrupted plan	PG2S	KAS	PG2S not corrupted	KAS not corrupted
trial_T20190909_075240_427378 (laptop → bread; pen → knife)	**0.138**	0.148	**0.458**	0.211
trial_T20190906_185208_580877 (bathroom → kitchen; bottle → tomato)	**0.167**	0.260	**0.417**	0.302
trial_T20190907_020543_865134 (monitor → statue; fire → lamp)	**0.000**	0.208	**0.500**	0.311
trial_T20190907_143702_923249 (moon → desk; mouse → card)	**0.000**	**0.000**	**0.875**	0.090
trial_T20190907_171916_941174 (pizza → coffee)	0.083	**0.055**	**0.167**	0.104
trial_T20190909_035341_047789 (card → pencil; restaurant → desk)	**0.000**	0.036	**0.500**	0.102

Table 3. Similarity values among predicted and ALFRED annotated plans using thirty plans in ten different scenes using tables and ground truth plans by the G-PLANET dataset.

Task_ID	Single-agent w/ table		Multi-agent w/ table		Single-agent w/ image		Multi-agent w/ image	
	PG2S	KAS	PG2S	KAS	PG2S	KAS	PG2S	KAS
trial_T20190907_161326_928347	0.00	**0.25**	0.00	**0.31**	**0.30**	0.11	**0.10**	0.05
trial_T20190910_173916_331859-1	**0.14**	None	**0.14**	None	**0.39**	0.15	**0.43**	0.19
trial_T20190909_004531_429065-1	**0.14**	None	**0.14**	None	**0.29**	None	**0.21**	None
trial_T20190907_114323_767231-1	**0.10**	None	0.30	**0.32**	**0.80**	0.38	**0.30**	0.24
trial_T20190906_234735_610018-1	**0.24**	None	**0.00**	None	**0.46**	None	**0.46**	None
trial_T20190907_200154_378982-1	**0.37**	None	**0.27**	None	**0.63**	0.19	0.20	**0.44**
trial_T20190907_114323_767231-2	**0.00**	None	0.10	**0.42**	**0.47**	0.26	**0.30**	0.09
trial_T20190906_234735_610018-2	**0.41**	None	0.07	**0.23**	**0.41**	None	**0.61**	0.06
trial_T20190909_082934_483899-1	0.31	**0.47**	**0.31**	None	**0.29**	0.19	**0.39**	None
trial_T20190909_100946_496614-1	**0.27**	None	**0.37**	None	**0.37**	0.21	**0.47**	0.17
trial_T20190907_200154_378982-2	**0.55**	None	**0.70**	None	**0.65**	0.20	**0.65**	0.18
trial_T20190909_082934_483899-2	**0.14**	None	**0.14**	None	**0.42**	0.14	**0.29**	0.14
trial_T20190909_012550_586494-1	**0.30**	None	**0.30**	None	**0.29**	0.05	**0.43**	0.08
trial_T20190909_082934_483899-3	**0.29**	None	**0.29**	None	**0.57**	0.21	**0.57**	0.10
trial_T20190909_100946_496614-2	**0.55**	None	**0.35**	None	**0.65**	0.34	**0.65**	0.42
trial_T20190906_234735_610018-3	**0.54**	None	**0.41**	0.35	**0.48**	None	**0.61**	None
trial_T20190909_012550_586494-2	**0.20**	None	**0.20**	None	**0.29**	None	**0.34**	None
trial_T20190907_114323_767231-3	**0.00**	None	**0.10**	None	**0.30**	0.17	**0.30**	0.13
trial_T20190907_114323_767231-4	**0.50**	None	0.00	**0.60**	**0.20**	0.12	**0.50**	0.23
trial_T20190909_004531_429065-2	**0.23**	None	0.23	**0.47**	**0.29**	0.23	**0.44**	0.15
trial_T20190909_193045_208933-1	**0.35**	None	**0.35**	None	**0.55**	0.24	**0.40**	0.19
trial_T20190909_193045_208933-2	**0.35**	None	**0.45**	0.33	**0.65**	0.27	**0.30**	0.20
trial_T20190907_114323_767231-5	**0.23**	None	**0.23**	None	**0.33**	0.23	**0.33**	None
trial_T20190910_173916_331859-2	**0.14**	None	**0.07**	None	**0.36**	0.11	**0.36**	0.12
trial_T20190909_004531_429065-3	**0.37**	None	**0.23**	None	**0.37**	None	**0.43**	0.21
trial_T20190907_200154_378982-3	**0.20**	None	**0.00**	None	**0.00**	None	**0.55**	0.04
trial_T20190910_173916_331859-3	**0.24**	None	0.31	**0.39**	**0.43**	0.16	**0.43**	0.20
trial_T20190907_114323_767231-6	**0.00**	None	0.00	**0.39**	**0.20**	None	**0.40**	0.30
trial_T20190909_193045_208933-3	0.00	**0.26**	0.00	**0.35**	**0.20**	None	**0.30**	0.21
trial_T20190909_012550_586494-3	**0.31**	None	**0.24**	None	**0.21**	0.18	**0.21**	0.19

while avoiding execution of the obtained plan to ensure the correctness of results. Although the presented metric can only provide a limited evaluation in cases where the final goal is not detailed enough, to the best of our knowledge, PG2S is the first contribution that addresses the problem in a way as general as possible. This paves the way for novel approaches where language processing techniques are adopted for the plan evaluation task. Future advances may improve the presented metric.

6 Conclusion

In this paper, we, first, introduced a multi-agent planning framework that leverages the capabilities of Visual Language Models (VLMs) to improve planning for embodied agents without the need for pre-encoded environmental data structures. Our approach simplifies the input requirements by utilizing a single environmental image and also enhances the adaptability and effectiveness of the planning process through a multi-agent system. This innovation addresses the limitations of traditional models that rely heavily on structured data, providing a more flexible and dynamic planning mechanism that is particularly effective in unstructured, real-world scenarios.

The empirical results, validated using the ALFRED dataset, demonstrate the efficacy of our approach, especially when compared to existing metrics like the KAS metric. We, then, introduce a new metric for the plan evaluation. The newly proposed PG2S metric, which assesses planning quality based on semantic understanding rather than strict action order, has shown superior performance in capturing the variations of plan execution.

The presented approach can address some of the current limitations in embodied agent planning and can open future research in the application of VLMs and multi-agent systems. Future studies might explore the scalability of our approach to more complex multi-agent environments and the integration of more diverse modalities to enhance the agents' understanding of their operational contexts. PG2S explores novel possibilities in the plan evaluation, focusing on semantic integrity rather than strict action sequencing. We believe that the research community can take advantage of the proposed approach, considering semantic coherence as a critical component of plan success, especially in applications requiring high reliability and safety.

Acknowledgements

This work has been carried out while Francesco Argenziano and Michele Brienza were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome.
This work has been partially supported by PNRR MUR project PE0000013-FAIR. The research reported in the paper was partially supported by the project "Tech4You (ECS00000009) - Spoke 6", under the NRRP MUR program funded by the NextGenerationEU.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[3] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer, 2016.

[4] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5664–5673, 2019.

[5] R. Asyrofi, M. R. Dewi, M. I. Lutfhi, and P. Wibowo. Systematic literature review langchain proposed. In *2023 International Electronics Symposium (IES)*, pages 533–537. IEEE, 2023.

[6] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] Z. Dai, A. Asgharivaskasi, T. Duong, S. Lin, M.-E. Tzes, G. Pappas, and N. Atanasov. Optimal scene graph planning with large language model guidance. *arXiv preprint arXiv:2309.09182*, 2023.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] V. S. Dorbala, J. F. Mullen Jr, and D. Manocha. Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation. *IEEE Robotics and Automation Letters*, 2023.

[11] M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420, 2017.

[12] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[13] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[14] X. L. Li, A. Kuncoro, J. Hoffmann, C. d. M. d'Autume, P. Blunsom, and A. Nematzadeh. A systematic investigation of commonsense knowledge in large language models. *arXiv preprint arXiv:2111.00607*, 2021.

[15] B. Y. Lin, C. Huang, Q. Liu, W. Gu, S. Sommerer, and X. Ren. On grounded planning for embodied tasks with language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13192–13200, 2023.

[16] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[17] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[19] S. Minton, J. Bresina, and M. Drummond. Total-order and partial-order planning: A comparative analysis. *Journal of Artificial Intelligence Research*, 2:227–262, 1994.

[20] E. Musumeci, M. Brienza, V. Suriani, D. Nardi, and D. D. Bloisi. Llm based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. *arXiv preprint arXiv:2402.14871*, 2024.

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[22] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[23] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *arXiv preprint arXiv:2307.06135*, 2023.

[24] S. Rasal. Llm harmony: Multi-agent communication for problem solving. *arXiv preprint arXiv:2401.01312*, 2024.

[25] N. Rekabsaz, M. Lupu, and A. Hanbury. Exploration of a threshold for similarity based on uncertainty in word embedding. In *Advances in Information Retrieval: 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings 39*, pages 396–409. Springer, 2017.

[26] M. Shanahan, K. McDonell, and L. Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

[27] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[28] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

[29] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020.

[30] Y. Talebirad and A. Nadiri. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*, 2023.

[31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[32] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[33] Z. Wang, S. Mao, W. Wu, T. Ge, F. Wei, and H. Ji. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*, 1(2):3, 2023.

[34] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[35] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[36] Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

[37] J. Yang, X. Chen, S. Qian, N. Madaan, M. Iyengar, D. F. Fouhey, and J. Chai. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. *arXiv preprint arXiv:2309.12311*, 2023.

[38] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022.

[39] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2023.