2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Task Planning for Long-Horizon Cooking Tasks Based on Large Language Models

Jungkyoo Shin[1], Jieun Han[2], SeungJun Kim[2], Yoonseon Oh[2], and Eunwoo Kim[1]

***Abstract*— In the field of robot manipulation, learnable task planners are gaining attention, especially for long-horizon tasks such as cooking. However, existing methods that predominantly rely on symbolic representations suffer from limitations in generalization capabilities, particularly in handling unseen objects. Given that objects may vary in real-world environments, this limitation may constrain their practical applicability. To address this issue, we propose a novel task-planning framework that leverages a pretrained large language model (LLM) for environmental interpretation. Our proposed framework extracts semantic features directly from textual data, enabling the planner to accommodate unfamiliar objects. We further incorporate a transformer-based encoder-decoder framework to understand environmental attributes derived from the language model and generate sequential predictions in line with object-oriented subgoals. To validate the effectiveness of our model, we utilize a dataset focused on cooking recipes. Going a step further, we propose a method that automatically generates object-oriented data from natural language description using recurrent LLM, enhancing the framework to manage previously unseen targets as well. Our framework shows an average success rate of 95% when validated with test sets that involve unseen objects. By providing the automatically generated dataset to the framework, we achieve a significant 27% increase in success rate on unknown target recipes. We also provide evidence of the real-world viability of our planner by successfully deploying it on a robot platform.**

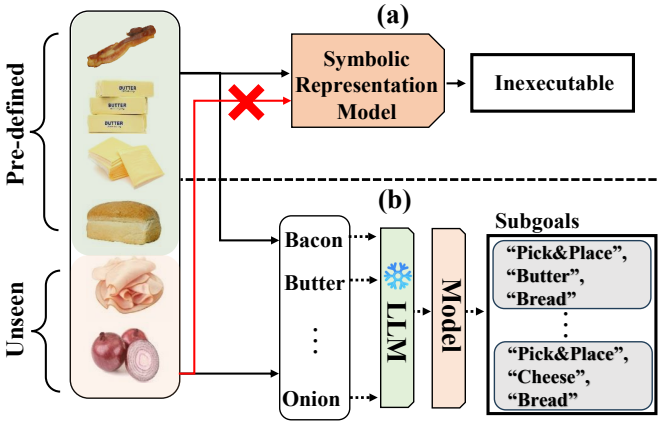

Fig. 1: (a) Conventional approaches based on symbolic representation are incapable of generating task plans when encountered with unseen objects due to their reliance on fixed domains. (b) In contrast, our framework utilizes features from the Large Language Model (LLM) with fixed parameters to interpret the environment linguistically and feed these features to a learnable model, enabling it to handle previously unseen objects in real-world scenarios and generate sub-goals for long-horizon cooking tasks.

I. Introduction

There is an increasing research interest aimed at enabling robots to execute long-horizon tasks, such as cooking [1]–[5]. Constructing task plans for real-world applications of long-horizon tasks is challenging because it requires a deep understanding of the domain and its sequential logic. In real life, many alternatives within the object domain may exist. For example, as shown in Fig. 1, when generating a sequence of task plans for a target recipe like a 'sandwich', ingredients such as 'ham' can be replaced with 'bacon', 'turkey', and so on. To utilize the recipe with multiple variations of a given object, an intelligent agent must possess a contextual understanding of the recipe's requirements and a grasp of the interrelationships among the provided ingredients. Conventional approaches [6]–[8] have often been reliant on symbolic task planners [9], [10], where planning domains are explicitly predefined by human experts. While effective to an extent, these predefined domains inherently limit the planner's ability to adapt to unseen objects or environments, thus limiting their applicability in real-world scenarios [11], [12].

To address these issues, recent studies have employed deep-learning-based task planners [1], [13], [14] that derive environmental insights directly from visual scene images. However, despite their advantages, which enable the understanding of an unconstrained environment, these methods still present certain challenges. Primarily, the implementation of image-based environment understanding requires substantial data and computational resources as images are composed of various complex visual cues [15]–[18]. However, the structural data for robot learning requires a vast amount of human labor, which limits the size of the dataset and the capability of task planners. Secondly, image-based methods make an implicit assumption of possessing a complete understanding of the observed environment. In the real world, environments frequently exhibit characteristics of non-determinism or partial observability [19]–[21]. In practical scenarios, robots may inadvertently omit certain

[1] Department of AI, Chung-Ang University, Seoul, Korea. (e-mail: {neo293, eunwoo}@cau.ac.kr). (Corresponding author: Eunwoo Kim.)
[2] Department of Electronic Engineering, Hanyang University, Seoul, Korea. (e-mail: {haneul2069, rlatmdwnseo, yoh21}@hanyang.ac.kr)

This work was supported in part by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT2002-05 and in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program(Chung-Ang University) and No. RS-2022-II220907, Development of AI Bots Collaboration Platform and Self-organizing AI).

environmental facets, and unforeseen additions or environmental changes may occur during task execution. Existing image-based approaches are challenged by accommodating such dynamic and uncertain conditions, thereby constraining the adaptability of the task planner.

We introduce a novel framework utilizing a pretrained large language model (LLM) to extract semantic features directly. Unlike image-based methods, this approach significantly reduces the need for extensive datasets and computational resources. Thus, it tackles the challenges of non-determinism and partial observability by using the natural variability in human language descriptions of objects and actions within a task sequence. The natural language inputs are transformed into expressive feature representations through a pretrained LLM, which provides flexibility beyond the constraints of predefined symbolic representations. To further enhance this capability, we utilize an external memory-based encoder-decoder neural network architecture to comprehend these features and create a sequence of subgoals, guiding the robotic actions in a structured manner.

Additionally, we present a method employing an autoregressive LLM, such as GPT-3 [22], to transform natural language description into an object-oriented structure. Despite the success of LLM, such as CLIP [23] or GPT [22] in linguistic tasks, their application in robotics has been limited by the challenge of translating diverse language inputs into actionable instructions. Our approach involves creating a pseudo dataset from internet-collected natural language description and converting it into an object-oriented structure for training our task planning framework. The integration of an LLM for task planning represents a significant advancement, enabling the integration of newly acquired knowledge from online sources into our planning process. This enhances the capability of robots to perform complex tasks with improved adaptability and knowledge.

Our key contributions are as follows:

- We demonstrate the effectiveness of our task planner integrated with the Large Language Model (LLM) to enhance adaptability through various real-world environments.
- We propose an encoder-decoder style framework for our task planner, which successfully captures the semantic features from LLM and generates subgoals across different tasks.
- By leveraging an auto-regressive LLM, we introduce a method to automatically generate a pseudo dataset to learn unknown knowledge, thereby enhancing the capability of our framework.
- We demonstrate the efficiency of our framework with a robotic platform. The proposed approach has a success rate of 95% for long-horizon cooking tasks.

II. Related Works

A. Object-Oriented Task Planner

Previous approaches in task planning mainly employ the Planning Domain Definition Language (PDDL) [24], a planning language based on symbolic representations within a predefined domain. Heuristic rules predefined by experts are often used to effectively search the optimal subgoals [6]–[8]. Furthermore, there has been a growing interest in investigating the capabilities of learning-based task planners [1], [13], [14], [25]–[27]. Employing various initial conditions, such as symbolic representations or initial scene imagery, these works generate task plans sequentially based on various neural network models, including recurrent neural networks [1] and graph neural networks [13].

Recent research [4], [5], [28] have adapted such methodologies for long-horizon tasks such as cooking recipes and proposed a Functional Object-Oriented Network (FOON) which effectively represents the relation between actions, objects, and the consequent state transition of these objects during the execution of a task. However, despite these improvements, existing task planners still rely on predefined domains. This constraint limits the possibilities of using it in real-world scenarios where the robot may encounter unseen objects. Our proposed approach leverages the capabilities of pretrained large language models (LLMs) to facilitate understanding and interaction with diverse objects upon the rich semantic information composed in the pretrained model. Consequently, this eliminates the requirement for predefined domains, empowering the model to generate subgoals even when presented with unseen objects, thereby improving its efficiency in real-world applications.

B. NLP-based task planner

Leveraging a pretrained LLM to robot task planners provides flexibility in understanding real-world domains. Various studies have employed pretrained embeddings [29], [30]. [29] extracted visual features using ResNet [31] and BERT [32] to comprehend both semantic and spatial knowledge. [30] leveraged CLIP [23], which jointly understands visual and linguistic information in a multi-modal manner. These works, however, generally do not accommodate long-horizon tasks that necessitate an understanding of extended contextual information. Emerging methodologies have started exploring the automatic generation of subgoals for task planners by leveraging an auto-regressive LLM [11], [33]–[35]. Based on generated prompts that are conditioned to salient information and robotic capabilities, these models generate text outputs that implicitly denote the subgoal actions. However, the auto-regressive LLM is not strictly conditioned to the specific robotic architecture or the environmental constraints, leading to the potential output of unfeasible objects or incomplete task plans. Such mismatches can result in execution failures. To address these concerns, our framework is designed to generate subgoals that comply with the given environmental constraints, thereby mitigating the risk of task execution failures.

III. Method

Our primary objective is to train a neural network to effectively learn long-horizon task planning through supervised learning. In this section, we define a problem formulation for

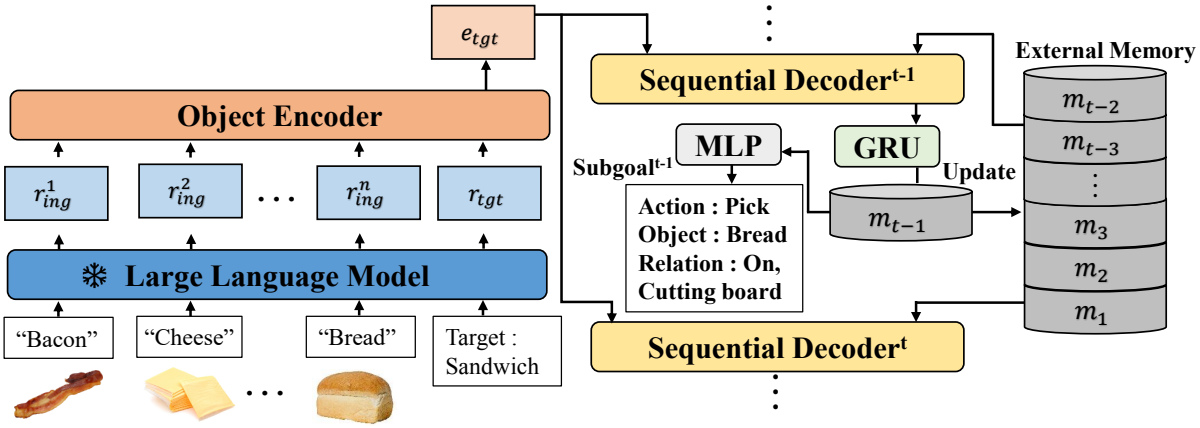

Fig. 2: Illustration of the proposed object-oriented task planner. Given a recipe name and a list of ingredients as initial input, the model understands the environment in a linguistic feature, aggregates the ingredients through an object encoder, and feeds the representation token into a sequential decoder to generate subgoals that perform a target recipe.

cooking task planners. Then, we introduce our novel encoder-decoder framework, as illustrated in Fig. 2. Our framework understands the given environment by leveraging the features of each recipe and ingredient from LLM and generates sequential subgoals. Lastly, we describe our strategy to augment the dataset by generating a pseudo-dataset using natural language description using autoregressive LLM based on well-crafted prompts.

A. Problem Statement

The task planner should discern the given environment (e.g., cooking) and generate an action sequence in an object-oriented structure for a robot to execute in the real world. The input consists of the name of the target recipe, w_{tgt}, and a list of ingredients, $W_{ing} = [w^1_{ing}, w^2_{ing}, \ldots, w^n_{ing}]$, where n is the number of the ingredients. The role of the task planner is to generate object-oriented sub-goals sequentially. The sub-goal of the task planner encompasses three primary predictions: intended robot action, objects involved in the action, and inter-relationship between objects. For example, to make a 'sandwich', our task planner generates a sequence of object-oriented subgoals such as (Action: "Pick and Place", Object: "Onion", relation : ("In", "bowl")) leading up to (Action: "Chopped", Object: "Onion", relation : ("On", "Cutting Board")).

B. Task Planning Framework

We split our proposed language-based task planner framework into three main modules. First, in Section III-B.1, we describe a pretrained linguistic encoder used to extract features from verbal descriptions of the environment. Next, Section III-B.2 introduces an object encoder that discerns specific features of individual ingredients and aggregated features for a given set of ingredients. Lastly, we elaborate on a sequential decoder that iteratively generates sub-goals for the target recipe in Section III-B.3.

1) Linguistic Encoder: By leveraging a pretrained LLM with a massive corpus dataset, we can extract latent embeddings that compose rich semantic features directly from linguistic annotations. We employ a pretrained text encoder from CLIP and extract latent embeddings. Given the list of ingredients W_{ing} and the name of target recipe W_{tgt}, we utilize the class token from CLIP or BERT as a latent embedding that represents each word, as below.

$$r_{tgt} = f_{LLM}(W_{tgt}) \in \mathbb{R}^{1\times d}, \tag{1}$$

$$r_{ing} = f_{LLM}(W_{ing}) \in \mathbb{R}^{n\times d}, \tag{2}$$

where f_{LLM} denotes the pretrained LLM, r_{tgt} and t_{ing} represent the semantic features of the target recipe and the ingredients, respectively.

2) Object Encoder: As mentioned earlier, a recipe can possess multiple variations that may replace each ingredient. To successfully generate a sequence of subgoals with diverse variations of ingredients, it is important to grasp the individual essence of ingredients and understand their synergistic interactions. By feeding the previously extracted latent embeddings through a transformer-based encoder, we assume that the semantic feature of each ingredient can be successfully obtained to make a target recipe, as below.

$$r_{in} = (r_{ing} \parallel r_{tgt}), \tag{3}$$

$$E = MHA_{enc}(r_{in}, r_{in}, r_{in}), \tag{4}$$

where $\parallel$ denotes the concatenation operation and MHA_{enc} represents the multi-head attention layer, utilizing query, key, and value inputs. We follow the approach in BERT [32], where a special class token is utilized to capture the semantic features of an entire sentence. We adopt the final vector from the encoder's output for the last input token, r_{tgt}, as a representative token embedding. We denote this token as e_{tgt} and assume it as a comprehensive representation of the ingredients for the target recipe.

3) Sequential Decoder: Based on the representative token e_{tgt}, our goal is to sequentially generate sub-goals, each of which considers both the current environment and the preceding state. To facilitate this, we adapt a novel transformer layer that incorporates external memory, allowing it to selectively retain the previous features and recurrently update them with

new features that consider the current state information. We facilitate this by utilizing cross-attention, as shown below.

$$h_t = MHA_{dec}(e_{tgt}, M_{t-1}, M_{t-1}), \tag{5}$$
$$m_t = GRU(e_{tgt}, h_t) \tag{6}$$
$$M_t = [m_1, m_2, \ldots, m_t], \tag{7}$$

where at time step t, our proposed decoder MHA_{dec} takes in e_{tgt} as query and duplicates the external memory from the previous step M_{t-1} to serve as both key and value. This process leverages the external memory of every previous stage, preventing challenges such as gradient vanishing, which can often be found in conventional recurrent neural networks. However, the continuous accumulation of external memory from every previous step is computationally inefficient and may include irrelevant information, which could complicate understanding subsequent tasks.

To address this, we retain the external memory for each step using the strategies used in the recurrent neural network [1], which employs past features to discard non-essential elements, retaining only the crucial features through a gating mechanism at every iteration. Consequently, this method saves only indispensable sequential information to the external memory, thus lowering the required computational overhead. Lastly, we feed the output of each time step m_t to a fully connected layer to extract subgoals composed of action, object, and relation, respectively, for each time step

$$y^t_{obj} = W_{obj} m_t + b_{obj}, \tag{8}$$
$$y^t_{act} = W_{act} m_t + b_{act}, \tag{9}$$
$$y^t_{rel} = W_{rel} m_t + b_{rel}, \tag{10}$$

where y^t_{obj}, y^t_{act} and y^t_{rel} represent the class probabilities of selecting the optimal ingredients, action, and relationship between the objects at time step t, respectively. We train our task planner end-to-end by minimizing the binary cross entropy loss for action prediction and the negative log-likelihood loss for object and relation prediction for every time step. We formulate the action prediction as a multi-label classification, as multiple actions for each object can happen.

C. LLM-Driven Data Generation

Training a task planner for long-horizon tasks requires a comprehensive dataset. However, creating such a dataset in an object-oriented format is demanding due to the precision needed in manual annotations. To mitigate this challenge, we propose a strategy to utilize a recursive LLM, such as GPT-3 [22], as illustrated in Fig. 3, to automatically convert real-world natural language descriptions into an object-oriented format through well-crafted natural language prompts. This method creates a pseudo dataset that facilitates the learning of unseen knowledge without the need for additional human intervention.

Due to the limited size of the training dataset, the framework may encounter an unfamiliar target recipe when generating sub-goals. To overcome this, we enhance our dataset with descriptions from the Recipe 1M+ dataset [36],

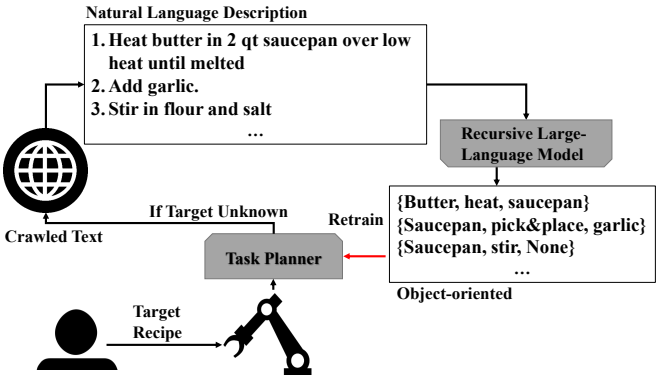

Fig. 3: Visualization of the process for generating a pseudo dataset. When the unknown target is given, we crawl the target recipe in a natural language description format to convert it into an object-oriented structure and retrain our task planner to obtain unexploited knowledge.

which contains natural language descriptions from internet crawling. Initially, we store the titles of recipes that were used to train our framework. We then measure the semantic similarity between the stored titles and titles from the Recipe1M+ dataset using the cosine distance. From the dataset, we identify unseen recipes by selecting those with a similarity score below a predefined threshold.

To supplement the knowledge of our task planning framework, we select the unseen recipe descriptions formatted in natural language. We concatenate the selected natural language descriptions with in-context prompts that reflect the capabilities and constraints of the robot, guiding GPT-3 to translate language into object-oriented structures. Outputs that fail to meet the criteria are discarded, such as suggesting impractical objects not described in the environmental constraints or incomplete structures that fail to satisfy object-oriented requirements. This selection process is repeated with another unseen recipe description when discarded until a viable output is obtained. With these outputs, we generate a pseudo-dataset to train our framework, significantly improving its ability to adapt and incorporate unseen knowledge.

IV. Experiment

This section describes the details of our training process, robot implementation, and the validation process. Our study has three primary goals: to validate the effectiveness of our framework when encountering unseen objects in real-world situations, to analyze how different architectural choices impact the performance of our framework, and to assess the applicability of our data generation method when dealing with previously unseen recipes.

A. Training Details

To train our framework, we utilized an object-oriented structured dataset based on online cooking instructional videos used in FOONets [5]. We utilized 20 raw recipes (e.g., club sandwich, banana pudding, etc.), including 160 ingredients and 8 actions. We constructed 20k samples from those

recipes by applying two augmentation methods. One involves object replacement, where we substitute ingredients used in a recipe with different ingredients. We used the synonym dictionary and reference recipes from the Recipe1M+ dataset to create a list of replaceable ingredients. For example, we could find multiple recipes for club sandwiches, where some recipes did not use hot sauce and used mayonnaise or mustard instead. By replacing the replaceable ingredients with others, we were able to augment the dataset. The other is order replacement, which entails rearranging the sequence of target objects within the same action and relation. For the sub-goals with the same action and relation, we assumed the action happened simultaneously and that changing the order would not harm the cooking process. Using these augmentation methods, we generated 1,000 recipe samples for each recipe.

We used 18k samples as a training set and 2k samples as a testing set. To show the effectiveness of our model in treating unseen objects, the test set contains more than one ingredient that is unseen from the training set. While these unseen ingredients are unable to be treated in the conventional task planning methods that employ predefined domains, the proposed method may effectively treat various objects from the real world due to its capacity to interpret and adapt based on linguistic features extracted from a pretrained large language model.

For generating the pseudo dataset as described in Section III-C, we undertook the following procedures. Initially, we selected 50 recipe names from the Recipe1M+ dataset that were not part of the original training dataset. For each recipe, we gathered their natural language descriptions from the dataset, matched them with well-crafted prompts, and used GPT-3 to transform them into object-oriented structures. This procedure was repeated to obtain 10 detailed object-oriented structured datasets for each recipe. We then applied an object replacement strategy to the pseudo-generated recipes, which increased our dataset by adding 1,708 new samples. Of these, 1,000 were allocated for training, and the remaining 708 were allocated for evaluation. To guarantee the integrity of the evaluation, we manually re-labeled the evaluation sets to verify that each data represents a complete cooking recipe.

For model training details, we use the Adam optimizer with an initial learning rate of 1e-4. The encoder and decoder each consist of a transformer with two fully connected layers, each with a hidden dimension size of 512, the GELU activation function, layer normalization, and four attention heads.

B. Evaluation Metric

Our objective is to assess the quality of the generated sub-goals for complex cooking tasks, which often involve multiple concurrent actions. This complexity makes it challenging to compare the sub-goals to a fixed ground truth reference directly. To effectively validate the quality of the generated sub-goals, we used a strategy known as the progress line [37]. In cooking, ingredients undergo various state changes, including changes in actions, objects, and the relationships between objects. We refer to these sequences of state changes as progress lines. To evaluate the accuracy of the generated sub-goals, we align each sub-goal with the corresponding progress line in the ground truth data. A generated sub-goal was considered successful if it was correctly aligned with its corresponding progress line. However, if a generated sub-goal is partially incorrect (e.g., involving an incorrect action or object), we classified it strictly as a failure. We calculated the success rate by determining the percentage of generated sub-goals that successfully align with the ground truth progress lines.

	FOON	Unseen	Pseudo	Average
RNN	91.84	31.85	60.85	61.51
Trnsf	94.95	22.57	61.32	59.61
Mem	93.45	30.88	61.30	61.88
Ours	**95.40**	**35.07**	**62.12**	**64.20**

TABLE I: Success rate of different architectures and datasets.

C. Real Experiments

To validate the practical applicability of the generated recipes in real-world scenarios, we conducted experiments using the 6-DOF UR5e robot arm equipped with a Robotiq-2F-85 gripper. We note that our study did not consider motion planning as this work primarily focuses on long-horizon task planning. Instead, we executed robot movements based on predefined actions and object positions.

We provide snapshots of two distinct processes in Fig 4, both targeting a recipe 'chocolate brownie' but with varying sets of ingredients. In the upper figure, the environment contains an additional ingredient, 'vanilla extract', which was not part of the training process. Guided by the goal of creating a 'chocolate brownie' and the list of ingredients, our task planner adaptively generated a sequence of sub-goals. The experiments resulted in the successful picking and placement of butter, a cup of water, and brownie mix in a small bowl and stirring the ingredients. Afterward, ingredients such as vanilla extract, egg yolk, and sugar were added and stirred once more before being placed into the oven. Conversely, in the lower figure, we explored the scenario of substituting 'vanilla extract' with 'lemon'. It is important to emphasize that 'lemon' was not part of the training data for a 'chocolate brownie'. Nonetheless, leveraging the features extracted from CLIP, our task planner adaptively generated a sequence of sub-goals accommodating the new ingredient. These examples underscore the capability of our proposed framework to accommodate and effectively manage previously unseen ingredients.

D. Architectural Validation

1) Encoder-decoder evaluation: To assess the effectiveness of our framework, we employed two primary architectures widely used for task planning, recurrent neural network (RNN) [1] and transformers (Trnsf) [38], [39], alongside a variant that incorporates external memory into the conventional transformer framework which we call Mem. The results of these approaches are summarized in Table I.

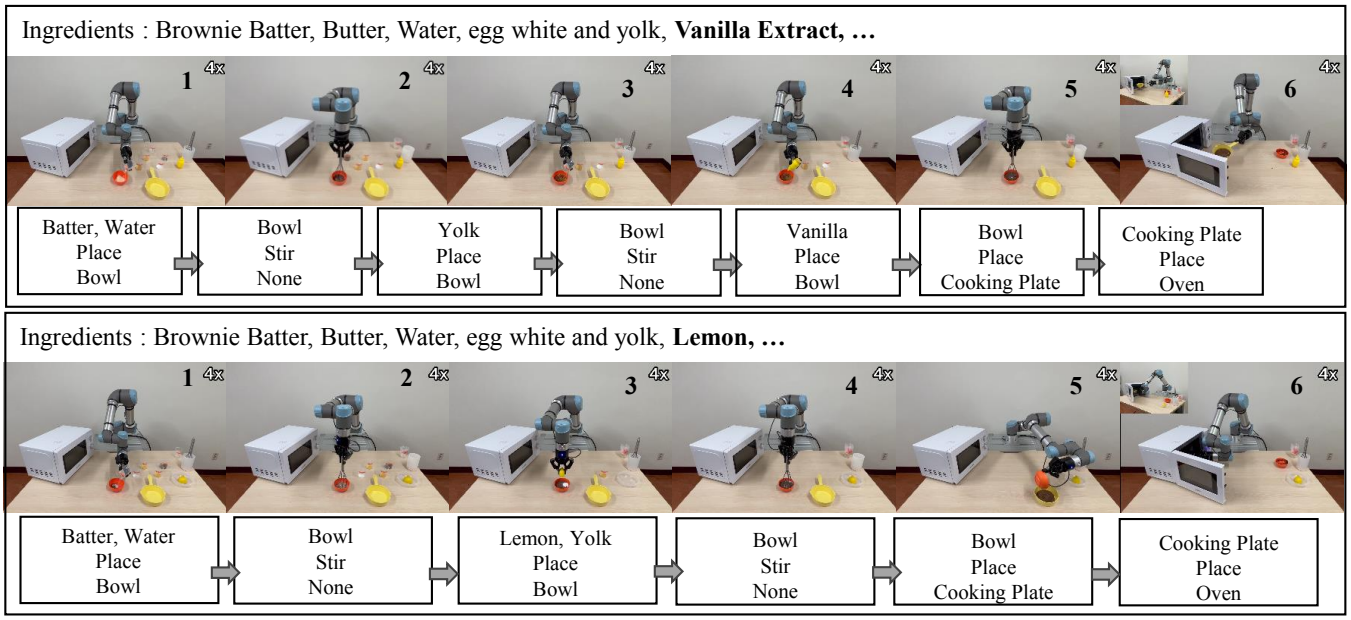

Fig. 4: An example of a real-world setup for two different sets of ingredients with the same target recipe. Each entity in the box denotes objects (top), actions (middle), and related objects (bottom) in an object-oriented structure.

We categorized the results into three groups: the results of supervised learning using FOONets as FOON, the results of unsupervised learning with unseen data as Unseen, and the results from retraining with pseudo-generated dataset as Pseudo. In supervised learning, the RNN method showed the lowest performance, underscoring the enhanced capability of transformer-based approaches to more effectively understand complex natural language features. In unsupervised learning, the basic transformer approach had the lowest success rate, which was significantly improved by 8.31% upon integrating the external memory. This demonstrates the benefits of an external memory structure for enhanced generalization in task sequence creation. Notably, our approach achieved the highest success rates across all cases, demonstrating that our framework effectively leverages the benefits of both RNNs and transformers in creating detailed task plans sequentially.

2) Pseudo data validation: Success rates for testing 50 distinct recipes that were not previously encountered are shown in Table I. We assessed the capability of our framework to generate task plans for these unseen recipes through unsupervised learning. Subsequently, we created a pseudo dataset tailored to these target recipes to retrain our framework. The retraining significantly enhanced the success rate of our framework by 27.05%. Consequently, it improved success rates across all scenarios, with an average increase of 31.30%. This constant enhancement in success rates across different models underscores the effectiveness of our method in adapting previously unknown information without the need for manual annotations.

3) Augmented size of dataset: We assess the model's success rate by varying the dataset size to 2k, 6k, 10k, and 18k samples. For each size, we augment it with an additional 100, 300, and 500 samples per recipe, respectively. The success rate for each dataset configuration is evaluated using the same test set comprising 2k non-overlapping samples. Specifically, with training dataset sizes of 2k, 6k, 10k and 18k, we achieve success rates of 52.75%, 91.76%, 92.49%, and 95.40%, respectively. Notably, increasing the dataset size results in significant improvements, highlighting the effectiveness of linguistic augmentation.

4) Large Language Models: We employed three different LLM: BERT [32], DistilBERT [40], and CLIP [23]. While BERT and DistillBERT focus on linguistic features, CLIP is trained to learn both linguistic and visual features in a multi-modal manner. We achieved 95.06%, 94.87%, and 95.40% with BERT, DistillBERT, and CLIP, respectively. Although the performance differences are not substantial, it is noteworthy that CLIP, which leverages multi-modal pretraining, exhibited the best results. This suggests that LLMs pretrained with a multi-modal approach enhance feature extraction capabilities for generating real-world tasks, possibly benefiting from including visual cues within the linguistic features.

V. Conclusion

In this work, we have leveraged pretrained Large Language Models (LLMs) to propose a language-based learnable task planner. The proposed task planner understands the environmental attributes in linguistic terms, enabling robots to work with unseen objects and targets in long-horizon cooking tasks. Furthermore, we have presented a method for additionally training task planners in an unsupervised manner, using an LLM to produce a pseudo dataset from language descriptions automatically. Our task planner achieved a success rate of 95.4% for cooking tasks involving unseen objects. We believe that the LLM-based architecture of our task planner could be broadly applied to various real-world tasks as well.

REFERENCES

[1] D. Xu, R. Martín-Martín, D.-A. Huang, Y. Zhu, S. Savarese, and L. F. Fei-Fei, "Regression planning networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[2] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *arXiv preprint arXiv:2003.06085*, 2020.

[3] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[4] D. Paulius, K. S. P. Dong, and Y. Sun, "Task planning with a weighted functional object-oriented network," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3904–3910.

[5] D. Paulius, A. B. Jelodar, and Y. Sun, "Functional object-oriented network: Construction & expansion," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5935–5941.

[6] J. Hoffmann, "Ff: The fast-forward planning system," *AI magazine*, vol. 22, no. 3, pp. 57–57, 2001.

[7] J. A. Baier, F. Bacchus, and S. A. McIlraith, "A heuristic search approach to planning with temporally extended preferences," *Artificial Intelligence*, vol. 173, no. 5-6, pp. 593–618, 2009.

[8] M. Helmert, "The fast downward planning system," *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.

[9] R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," *Artificial intelligence*, vol. 2, no. 3-4, pp. 189–208, 1971.

[10] C. Aeronautiques, A. Howe, C. Knoblock, I. D. McDermott, A. Ram, M. Veloso, D. Weld, D. W. SRI, A. Barrett, D. Christianson, *et al.*, "Pddl— the planning domain definition language," *Technical Report, Tech. Rep.*, 1998.

[11] 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.

[12] 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," *arXiv preprint arXiv:2212.04088*, 2022.

[13] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548.

[14] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese, "Neural task programming: Learning to generalize across hierarchical tasks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3795–3802.

[15] J. Fu, A. Korattikara, S. Levine, and S. Guadarrama, "From language to goals: Inverse reinforcement learning for vision-based instruction following," *arXiv preprint arXiv:1902.07742*, 2019.

[16] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, "Latte: Language trajectory transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7287–7294.

[17] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "A recurrent vision-and-language bert for navigation. arxiv 2021," *arXiv preprint arXiv:2011.13922*.

[18] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.

[19] L. P. Kaelbling and T. Lozano-Pérez, "Integrated task and motion planning in belief space," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1194–1227, 2013.

[20] D. Hadfield-Menell, E. Groshev, R. Chitnis, and P. Abbeel, "Modular task and motion planning in belief space," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4991–4998.

[21] C. Phiquepal and M. Toussaint, "Combined task and motion planning under partial observability: An optimization-based approach," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9000–9006.

[22] 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*, vol. 33, pp. 1877–1901, 2020.

[23] 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*. PMLR, 2021, pp. 8748–8763.

[24] B. Wally, J. Vyskočil, P. Novák, C. Huemer, R. Šindelár, P. Kadera, A. Mazak, and M. Wimmer, "Flexible production systems: Automated generation of operations plans based on isa-95 and pddl," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4062–4069, 2019.

[25] B. Eysenbach, R. R. Salakhutdinov, and S. Levine, "Search on the replay buffer: Bridging planning and reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[26] D. Shah, P. Xu, Y. Lu, T. Xiao, A. Toshev, S. Levine, and B. Ichter, "Value function spaces: Skill-centric state abstractions for long-horizon reasoning," *arXiv preprint arXiv:2111.03189*, 2021.

[27] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel, "Learning plannable representations with causal infogan," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[28] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning. in 2016 ieee," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2655–2662.

[29] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2robot: Learning manipulation concepts from instructions and human demonstrations," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1419–1434, 2021.

[30] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] 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.

[33] A. Zeng, M. Attarian, B. Ichter, K. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, *et al.*, "Socratic models: Composing zero-shot multimodal reasoning with language," *arXiv preprint arXiv:2204.00598*, 2022.

[34] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 523–11 530.

[35] 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*. PMLR, 2022, pp. 9118–9147.

[36] J. Marın, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 187–203, 2021.

[37] M. S. Sakib, D. Paulius, and Y. Sun, "Approximate task tree retrieval in a knowledge network for robotic cooking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 492–11 499, 2022.

[38] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 538–547.

[39] V. Jain, Y. Lin, E. Undersander, Y. Bisk, and A. Rai, "Transformers are adaptable task planners," in *Conference on Robot Learning*. PMLR, 2023, pp. 1011–1037.

[40] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.