IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

NaturalVLM: Leveraging Fine-grained Natural Language for Affordance-Guided Visual Manipulation

Ran Xu*, Yan Shen*, Xiaoqi Li, Ruihai Wu, Hao Dong

***Abstract*—Enabling home-assistant robots to perceive and manipulate a diverse range of 3D objects based on human language instructions is a pivotal challenge. Prior research has predominantly focused on simplistic and task-oriented instructions, *i.e.*, "Slide the top drawer open". However, many real-world tasks demand intricate multi-step reasoning, and without human instructions, these will become extremely difficult for robot manipulation. To address these challenges, we introduce a comprehensive benchmark, NrVLM, comprising 15 distinct manipulation tasks, containing over 4500 episodes meticulously annotated with fine-grained language instructions. We split the long-term task process into several steps, with each step having a natural language instruction. Moreover, we propose a novel learning framework that completes the manipulation task step-by-step according to the fine-grained instructions. Specifically, we first identify the instruction to execute, taking into account visual observations and the end-effector's current state. Subsequently, our approach facilitates explicit learning through action-prompts and perception-prompts to promote manipulation-aware cross-modality alignment. Leveraging both visual observations and linguistic guidance, our model outputs a sequence of actionable predictions for manipulation, including contact points and end-effector poses. We evaluate our method and baselines using the proposed benchmark NrVLM. The experimental results demonstrate the effectiveness of our approach. For additional details, please refer to https://sites.google.com/view/naturalvlm.**

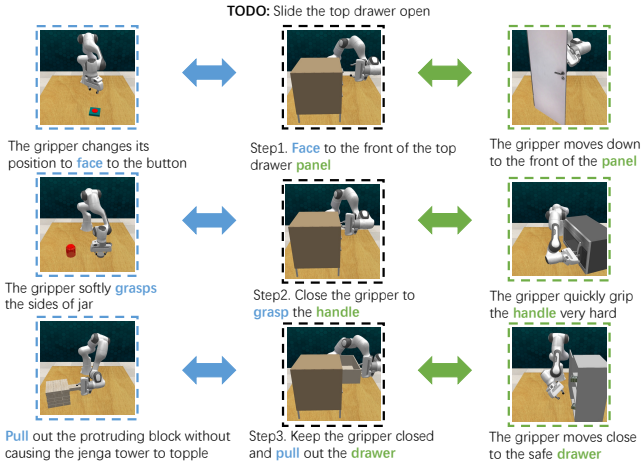

Fig. 1. Illustration on the fine-grained instructions. The leftmost and rightmost pairs represent action-prompt and perception-prompt bases respectively. In the center column are the manipulation steps for "Slide the top drawer open," each accompanied by fine-grained language instructions. If the current task's manipulation step shares the same action or noun phrase as another task's manipulation step in the fine-grained language instruction, cross-modal alignment will be conducted using the features of the action-prompt base and the perception-prompt base.

I. INTRODUCTION

Language serves as a crucial means for robots to engage with the world [9], [23], [25]. Robots must not only comprehend language instructions but also integrate them with real-time visual observations to make informed predictions for manipulation tasks. The existing benchmark VLMbench [30], provides high-level language instructions to guide robot agents, such as "pick up the red plate". However, relying solely on high-level instructions presents challenges, particularly for complex or unfamiliar tasks. Without the inclusion of low-level language instructions for guiding robots through each step of a task, successful task completion becomes exceedingly difficult. While some previous efforts have harnessed large language models [5], [11] to generate low-level instructions, these instructions tend to lack diversity in language style and often fail to accurately describe end-effector commands. The detailed illustration is shown in Sec. V-C.

Manuscript received: March 12, 2024; Revised: July 23, 2024; Accepted: September 25, 2024.

This paper was recommended for publication by Editor Aleksandra Faust upon evaluation of the Associate Editor and Reviewers' comments.

All authors are with School of CS, Peking University and National Key Laboratory for Multimedia Information Processing. Xiaoqi Li is also with Beijing Academy of Artificial Intelligence (BAAI).

* Equal Contribution. Author ordering determined alphabetically.

Corresponding to hao.dong@pku.edu.cn

Digital Object Identifier (DOI): see top of this page.

In this work, we aim to address existing limitations by introducing a novel task: low-level visual language manipulation. This task involves employing natural language to provide step-by-step instructions for visual manipulation tasks. To enable this, we propose the NrVLM dataset, which offers a collection of natural language instructions meticulously paired with manipulation episodes collected from V-REP simulation. Our dataset comprises an extensive dataset of 4500 episodes featuring 82 distinct variations across 15 tasks. We split the robot's manipulation task into discrete steps, each thoughtfully annotated in natural language. In Fig.1, we illustrate the fine-grained instructions for "Slide the top drawer open". These instructions encompass crucial details, *i.e.*, the required robot actions, designated objects for interaction, and the necessary end-effector state for each step. The purpose of these language instructions is to guide the agent in successfully executing manipulation tasks. To the best of our knowledge, the NrVLM dataset we present stands as a pioneering benchmark, uniquely combining manipulation trajectories with free-form instructional language, offering an invaluable resource for advancing research in this field.

Along with the benchmark, we further devise a framework that enables the agent to follow the instructions and execute step by step. This framework effectively utilizes a multi-modal approach, incorporating various information sources including visual observations of the current scene, the present state of the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
COMPARISON BETWEEN DIFFERENT BENCHMARKS.

Dataset	diverse objects	high-level task description	fine-grained instructions	different evaluation settings
Metaworld [27]	✗	✓	✗	✗
ManipulaTHOR [2]	✓	✗	✗	✗
RLBench [8]	✗	✓	✗	✗
CALVIN [16]	✗	✓	✗	✗
VLMBench [30]	✓	✓	✗	✗
NrVLM	✓	✓	✓	✓

end-effector, high-level language instructions, and fine-grained language instructions to generate actions for each individual step. Concretely, when presented with a sequence of low-level instructions, our model initially assesses the required execution step by analyzing the current visual scene, the end-effector status, and the high-level linguistic instruction. Subsequently, in order to fully exploit information from multiple modalities and predict reliable manipulation actions, we facilitate explicit learning through manipulation-aware cross-modality feature alignment as shown in Fig. 1. To enhance this alignment process, we establish both action-prompt and perception-prompt bases in advance since the features associated with actions and objects remain consistent across different tasks. The action-prompt base pairs action phrases (e.g., "pull", "grasp") with their corresponding action features, while the perception-prompt base pairs noun phrases (e.g., "drawer", "handle") with their corresponding object features. These pre-established priors significantly aid in aligning the respective multi-modal features, resulting in better predictions for manipulation actions.

To investigate the difficulty of the benchmark NrVLM and evaluate the performance of our method, we conduct extensive evaluations with four competitive visual language approaches and an ablated version. The results demonstrate the effectiveness of our framework against other methods.

In summary, we make the following contributions :

- We present the NrVLM, a comprehensive benchmark that combines diverse manipulation trajectories with fine-grained natural instructions, facilitating the agents in executing complex tasks sequentially.
- We propose a novel framework that enables the agent to utilize fine-grained instructions and acquire the manipulation-aware multi-modality alignment.
- Experimental results against four baselines validate the effectiveness and superiority of the proposed approach.

II. RELATED WORK

A. Robotic Manipulation Benchmarks

There are plenty of benchmarks related to visual-language robotic tasks [27], [2], [24], [8], [30]. ALFRED [24] is proposed for vision-and-language navigation and virtual object rearrangement tasks between different room-scale locations. MetaWorld [27] collected 50 translation-only tasks with demonstrations for reinforcement learning. ManipulaTHOR [2] is the first testing framework to study robot manipulation problems in more than 100 visually enriched, physicalized virtual room scenarios. RLBench [8] provides a benchmark and learning environment for both 'robot learning' and 'traditional' methods, with 100 completely unique, hand-designed tasks. CALVIN [16] encompasses a total of 34 distinct tasks and provides natural language instructions for long-horizon manipulation tasks. Following RLBench, VLM-Bench [30] collects a robot manipulation benchmark on 3D tasks with visual observation and compositional language instructions. As shown in Table I, different from previous work, our NrVLM benchmark offers 82 distinct variations across 15 tasks and 4500 low-level natural language instructions that annotate the robotic manipulation trajectory step by step. Such instructions aim to guide the agent to accomplish long-term and intricate tasks. In addition, to increase the challenge of the benchmark, we also provide different evaluation settings, including assessment on both training tasks and novel tasks.

B. Vision-and-Language Manipulation

Among visual language manipulation approaches [28], [22], [9], [30], [4], [10], BC-Z [9] develops imitation learning system to enable a vision-based robotic manipulation system. It aims to generalize to novel tasks and address a long-standing challenge in robot learning. Furthermore, PERACT [23] is an innovative behavior-cloning agent that utilizes a Perceiver Transformer [6] to encode both language and voxel scenes. The model exhibits strong performance in multi-task manipulation. In contrast, our framework focuses on learning to follow low-level instructions to perform step-by-step manipulation actions. Our approach adopts an action and perception prompt-based system that enables a thorough, manipulation-aware understanding of multiple modalities.

III. FINE-GRAINED INSTRUCTED MANIPULATION

A. Problem definition

In the low-level visual language manipulation task, the agent is provided with fine-grained language instructions to follow in order to complete the manipulation task. Specifically, to begin with, the agent is provided with a sequence of natural language instructions $\mathcal{L} = \{l_1, l_2, , , l_n\}$ (n denotes the maximum length of the language instructions) and current visual observations, including multi-view RGB images, depth images, and segmentation information. Given the initial state of the robot, the agent needs to predict an executable action command for the robot based on linguistic and visual information. After executing the action, the agent obtains a new set of visual observations and predicts more actions until it completes the sequence of low-level instructions.

B. Data Collection

Drawing inspiration from RLBench [8], we conduct data collection in V-REP simulation [21], interfaced with PyRep [7]. We collect data on 15 distinct manipulation tasks, encompassing a spectrum from simple tasks (*e.g.*, reach target) to more challenging ones (*e.g.*, stack cups). The number of manipulation steps required ranges from 1 to 20. Fig 2 visually showcases some of these tasks included in our benchmark.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Fig. 2. We introduce NrVLM, a comprehensive benchmark comprising multiple manipulation tasks annotated with fine-grained natural language instructions. Visualization of select tasks from the benchmark is presented in the top two rows. Additionally, we introduce difference task variations to enrich the diversity and complexity of the benchmark, as demonstrated in the bottom two rows.

This diverse array of task difficulties facilitates a comprehensive evaluation of the trained agents' performance.

To enhance the richness and complexity of our dataset, we introduce variations within each task. These variations involve introducing different object instances or altering shape geometries, thereby augmenting the diversity of challenges. Specifically, for manipulation tasks like "close-box" or "close-microwave", we select specific object instances from PartNet-Mobility dataset [17] that can replace the original objects provided in RLBench. For example, in the case of the task "close-microwave", the size, type, geometry, and color of the manipulated microwave vary in different task variations. These selected objects exhibit different properties and characteristics, contributing valuable variations to our dataset. We illustrate some of these task variations in the two bottom rows of Figure 2. Each task typically offers between 1 to 15 variations.

From these variations, we can generate an infinite number of manipulation episodes for training and evaluation. Within each variation, manipulation episodes differ in various aspects, such as the initial position and orientation of objects on the workbench, and the objects' relative spatial arrangements. Consequently, this results in diverse manipulation trajectories. This design enriches the diversity of manipulation trajectories required to accomplish each task, thereby bolstering the robustness of the trained agent.

All these tasks are categorized into three sets: a training set, a validation set, and a test set. The training set includes a total of 8 manipulation tasks with 46 variations, and both the validation and test sets include a total of 15 manipulation tasks (8 training tasks with an extra 7 novel tasks) with 82 variations. Each task comprises 300 manipulation episodes. The allocation of episodes between the training, validation, and test sets follows a ratio of 10:1:1. Our objective with this setup is to assess the agent's capability to successfully perform these novel tasks with the guidance of low-level natural language instructions. This evaluation helps us understand the agent's adaptability and generalization to new tasks and scenarios.

C. Fine-grained Instruction Annotation

In previous research on Visual Language Manipulation (VLM) tasks, language instructions primarily consisted of high-level descriptions. For example, an instruction for the "close-box" task might be as simple as "shut the lid of the box." The high-level descriptions remain totally the same for different manipulation attempts in the same task. While these high-level instructions convey the overall task objective, we have observed that most manipulation tasks should be accomplished by several steps and are naturally composed of several key actions. Therefore, these high-level instructions lack explicit guidance for the robot on how to perform these actions and accomplish the task step by step. To address this gap, we introduce the concept of fine-grained language instructions to guide the agent to complete the task step by step. Besides, the fine-grained language instructions are more precise and diverse compared to the unchanged high-level descriptions.

To generate fine-grained language instructions according to the sequential manipulation steps, we first split each expert manipulation demonstration into several slices. Following previous work [12], [23], we conducted keyframe action extraction to split the manipulation episode. The principle of keyframe action extraction is based on two criteria: (1) the velocity of the robot joints approaches zero, and (2) the gripper's open state changes. After splitting the demonstrations according to the keyframe actions, annotators can describe the precise robotic manipulation and provide fine-grained instructions for each step.

These annotated natural language instructions include action verbs, object noun phrases, and other details that explicitly outline how the end-effector should complete each step of the task. It is worth noting that the number of annotated natural instructions is closely related to the complexity of the task. Tasks of greater complexity, which demand a larger number of steps for completion, are accompanied by longer sets of fine-grained instructions. To maintain diversity and prevent the emergence of overly uniform linguistic styles in fine-grained instructions across different manipulation episodes, each task requires at least ten distinct annotators to produce the fine-grained natural instructions. Each annotator undergoes an independent annotation process without access to others' results. This approach significantly enhances the diversity of language instructions, ensuring a broader range of language styles and expressions.

IV. METHOD

A. Task Formulation

In our tasks, each demonstration is split into several steps, with each step paired with a fine-grained language instruction l and a keyframe action a. At each step, the agent receives visual observations, linguistic guidance, and agent state. The visual observation is the scene point cloud acquired from

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

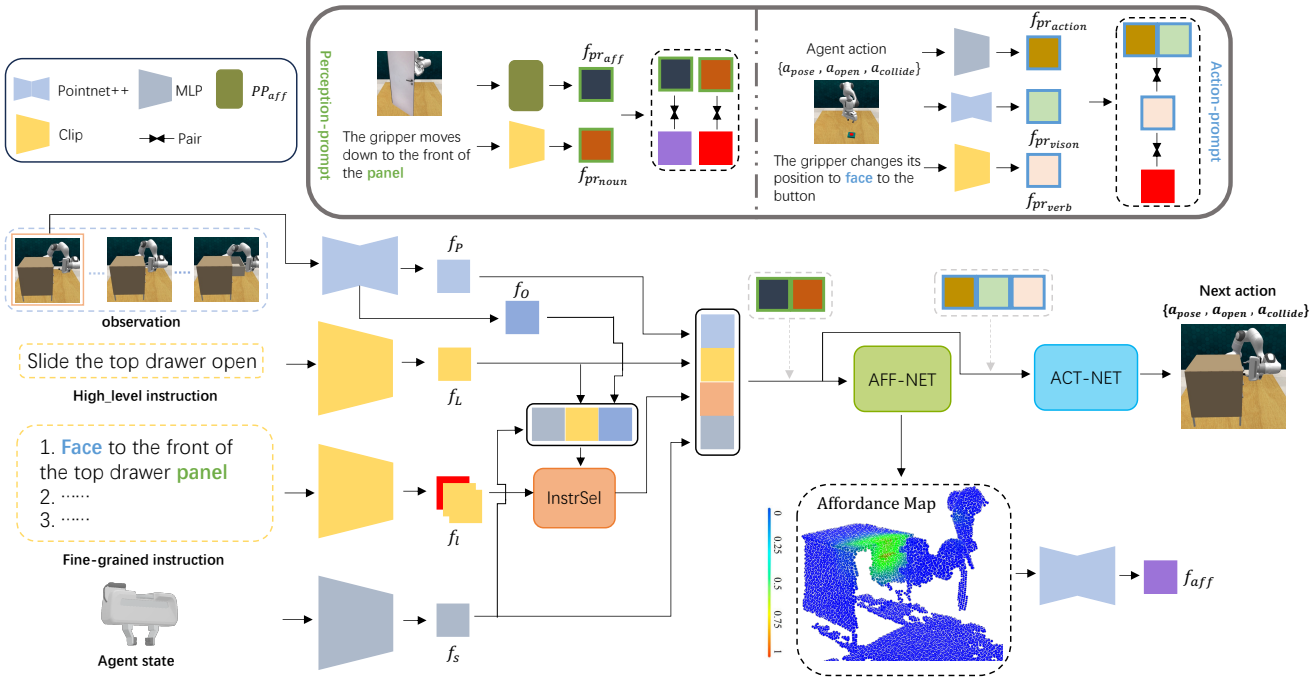

Fig. 3. **The overall framework.** The bottom part shows the manipulation process, where the Instruction Selection network (InstrSel) selects the appropriate fine-grained language instruction, the Affordance network (AFF-NET) predicts the object-centric affordance map, and the Actor network (ACT-NET) predicts the gripper action. The top part shows the alternative perception-prompt module and action-prompt modules, they enhance the Affordance and Actor networks by aligning the noun-related perception-prompt set and verb-related action-prompt set. The two dotted arrows before Affordance and Actor networks indicate that the prompt modules are optional. The entire method is trained in an end-to-end manner.

the RGB-D taken by depth cameras, while the linguistic information contains both high-level and fine-grained language instructions. The agent state is a 4-DoF vector that includes 1-DoF gripper open state and 3-DoF gripper position. Given these multi-modal information, the agent is expected to predict the gripper action $a = \{a_{pose}, a_{open}, a_{collide}\}$, which includes a 6-DoF pose, a 1-DoF gripper open state, and a 1-DoF collision state that determines whether the motion-planner uses collision avoidance to reach an intermediate pose. By formulating this manipulation problem as a behavior-cloning problem, we adopt key action as expert action to supervise the agent's predicted action for each step, enabling the agent to learn from the expert demonstrations.

B. Framework Overview

Fig. 3 presents an overview of our proposed framework. We will begin by discussing the bottom part of the figure, followed by an explanation of the top section.

In the bottom part, at each step, given the current point cloud observation, agent state, high-level instruction, and a sequence of fine-grained language, we first utilize the *Instruction Selection network (InstrSel)* (Sec. IV-C) to process the multi-modal information and select the appropriate fine-grained instruction from the fine-grained instruction sequence. Next, we use *Affordance network* (AFF-NET) (Sec. IV-D) to predict an object-centric affordance map that highlights actionable areas (e.g., to open the door, the affordance map highlights the door handle). Based on the affordance map, we can filter out low-rated regions and select a contact point. Finally, we incorporate *Actor network* (ACT-NET) (Sec. IV-D) to predict the precise gripper position based on the selected contact point, along with gripper rotation, open state, and collision state.

In the top part of the figure, we aim to bridge the three modalities including vision, language, and manipulation. We introduce two prompt modules: the *perception-prompt module* and the *action-prompt module* (Sec. IV-E). The *perception-prompt module* enhances the *Affordance network* by aligning the noun-related perception-prompt set, while the *action-prompt module* improves the *Actor network* by aligning the verb-related action-prompt set. For example, in the instruction "grasp the drawer handle", the noun "handle" refers to a specific object part that the agent should interact with, while the verb "grasp" indicates the action should take. Thus, the instruction noun connects the language and vision modalities, while the verb connects the language and manipulation modalities. Note that the integration of prompt modules is optional, as illustrated in the bottom part of Fig. 3 where the two gray dashed arrows before the Affordance and Actor networks indicate the optional nature of prompt features.

C. Instruction Selection Network

Fine-grained language instructions provide the agent with a step-by-step understanding of the manipulation task. After receiving the sequence of fine-grained language instructions, the agent first selects the appropriate instruction based on the current scene. To process the multi-modal information, we use pre-trained CLIP [20] to extract the text feature f_{l_i} for each

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

fine-grained instruction and f_L for the high-level instruction. PointNet++ [19] is used to extract the global feature f_o from the point cloud observation, while an MLP network encodes the agent state feature f_s. Subsequently, our *Instruction Selection network*, implemented as an MLP, evaluates the similarity of each fine-grained language instruction f_{l_i} with the current scene information, incorporating f_o, f_L, and f_s. We then apply the softmax operation on the similarity score and obtain the normalized weight to select the most related instruction. By summing up the fine-grained language features f_{l_i} according to the normalized weight, we can obtain the weighted feature of the fine-grained language instruction f_l. The normalized weight is formulated as a one-hot vector and is supervised by ground-truth one-hot vector under NLL loss. This *Instruction Selection network* enables the agent to select the most related instruction by comprehending both the fine-grained instruction and the current scene.

D. Affordance Network and Actor Network

The *Affordance network* predicts affordance map for manipulation task based on multi-modal information, indicating the manipulation region for the agent. We use PointNet++ [19] to extract the per-point feature f_p. The Affordance Network, implemented as an MLP, takes as input the concatenation feature f_p, f_s, f_l, and f_L, and outputs the affordance score $\in [0, 1]$ for each point. Point-wise affordance score forms the affordance map, from which we select a contact point $p' \in \mathbb{R}^3$ with a high affordance score. Following [26], we obtain the ground-truth affordance by placing a 3D Gaussian on the ground-truth interaction location from the expert demonstration. The Affordance Network is supervised using binary cross-entropy loss between the predicted affordance map and the ground-truth.

The *Actor network* predicts the action based on the selected contact point. Note that the gripper' 3D position differs from the contact point. For instance, in the case of lifting, the contact point is located on the object, whereas the target gripper position must be above the contact point. Consequently, the movement offset is introduced since different primitives require varying offsets in different directions at the contact point. The Actor network predicts the movement $a_{move} \in \mathbb{R}^3$ based on the selected contact point p', and the gripper position is then calculated using the formula $a_{position} = p' + a_{move}$. Additionally, the network predicts the rotation a_{rot} in a quaternion format, the gripper open state a_{open}, and the collision state $a_{collide}$. The predicted gripper position $a_{position}$ and rotation a_{rot} are combined to form the gripper pose a_{pose}. The Actor network includes one encoder that takes the concatenated features of $f_{p'}$, f_s, f_l, and f_L, and four decoders that predict a_{move}, a_{rot}, a_{open}, and $a_{collide}$, respectively. Both the encoder and decoders are implemented using an MLP architecture. To supervise the predicted gripper action, we use L1 loss for a_{move}, quaternion distance loss for a_{rot}, and cross-entropy loss for a_{open} and $a_{collide}$. The quaternion distance loss is designed to minimize the difference between the prediction q_{pred} and the ground-truth q_{GT}:

$$L(q_{pred}, q_{GT}) = 1 - \frac{1}{2}(q_{pred} * q_{GT} + q_{GT} * q_{pred}). \quad (1)$$

E. Prompt module

The three modalities of vision, language, and manipulation are intricately linked in Vision-and-Language Manipulation tasks. Concretely, in fine-grained instruction, the verb focuses on action knowledge, which connects the language and manipulation modalities, while the noun focuses on perceptual knowledge, which connects the language and vision modalities. Moreover, similar nouns in different instructions can indicate similar target parts, while similar verbs can indicate similar manipulation actions. To better leverage this correspondence between the modalities for learning perceptual and action knowledge, inspired by [13], we introduce the Prompt module to explore multi-modal alignment knowledge and help the agent better understand and execute the task.

To begin with, there are two types of prompts in our tasks: perception-prompts, includes a fine-grained instruction and the corresponding affordance map related to the noun, and action-prompts, which combines a fine-grained instruction and the corresponding action denoted by the verb. Since the action cannot be isolated from a specific scene, we add visual observation in action prompt pairs. Therefore, we can build two prompt bases for perception and action based on the nouns and verbs. The prompt bases can be thought of as a large dictionary, where the key is the important nouns or verbs and the value is the corresponding perception-prompt or action-prompt.

The Prompt module is designed to learn multi-modal alignment knowledge and consists of two sub-modules: the *perception-prompt module* and the *action-prompt module*. The *perception-prompt module* is designed to improve the Affordance network. During training, the noun-related perception-prompt set (the instruction and the corresponding observation) is retrieved from the pre-built perception-prompt base and fed into the perception-prompt module. This module uses the pre-trained CLIP to extract the text feature $f_{pr_{noun}}$, and uses the perception-prompt affordance network PP_{aff} to extract the affordance feature $f_{pr_{aff}}$. The PP_{aff} consists of the same Affordance network from the bottom part of Fig. 3 to predict the affordance map, and PointNet++ to encode the global feature $f_{pr_{aff}}$ of the affordance map. Finally, these noun-related prompt feature $(f_{pr_{aff}}, f_{pr_{noun}})$ are concatenated with the original input features and fed into the Affordance network.

Similarly, the *action-prompt module* retrieves the verb-related action-prompt set (the instruction, the corresponding action, and the scene observation) from the pre-built action-prompt base. The module uses the pre-trained CLIP to extract the text feature $f_{pr_{verb}}$, the PointNet++ to extract the observation feature $f_{pr_{vision}}$, and an MLP to extract the action feature $f_{pr_{action}}$. These prompt features $(f_{pr_{action}}, f_{pr_{vision}}, f_{pr_{verb}})$ are then concatenated with the original input feature and fed into the Actor network.

To improve the performance of the frozen CLIP model on our task, we add a shared soft-prompt [15], a vector of 20-length learnable parameters, to each fine-grained instruction's tokens. The soft-prompt parameters are updated during training, enabling the text features to better align with our task without requiring fine-tuning of the CLIP model.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II
WE PRESENT A COMPARATIVE ANALYSIS OF OUR METHOD AGAINST BASELINE METHODS, REPORTING TASK SUCCESS RATES (BEFORE SLASH) AND INSTRUCTION FOLLOWING RATES (AFTER SLASH) ON TRAIN TASKS (ABOVE) AND NOVEL TASKS (BOTTOM).

Train Tasks	close box	close door	open drawer	push button	slide cabinet	turn tap	take umbrella	open bottle
PERACT	0.36 / -	0.72 / -	**0.44** / -	0.40 / -	0.24 / -	0.76 / -	0.28 / -	0.52 / -
BC-Z	0.44 / -	0.84 / -	0.00 / -	0.08 / -	0.20 / -	0.00 / -	0.16 / -	0.08 / -
PERACT + F	0.36 / -	0.84 / -	**0.44** / -	**0.44** / -	**0.44** / -	0.68 / -	0.48 / -	0.56 / -
BC-Z + F	0.48 / 0.37	**0.92** / 0.45	0.12 / 0.45	0.20 / 0.44	0.08 / 0.33	0.00 / 0.23	0.28 / 0.58	0.12 / 0.42
Our w/o Pr	0.60 / **0.65**	0.76 / 0.94	0.16 / 0.94	0.20 / **0.98**	0.08 / **0.90**	0.40 / 0.85	0.24 / 0.99	0.60 / 0.65
Ours	**0.68** / 0.62	**0.92** / **1.00**	0.24 / **0.96**	0.28 / **0.98**	0.20 / 0.72	**0.80** / **0.94**	**0.52** / **1.00**	**0.80** / **0.82**
Novel Tasks	close drawer	close laptop	close microwave	lamp on	open door	open grill	fetch drawer item	Average
PERACT	0.60 / -	0.40 / -	0.52 / -	**0.08** / -	0.08 / -	0.28 / -	0.00 / -	0.38 / -
BC-Z	0.88 / -	0.04 / -	0.32 / -	0.00 / -	0.04 / -	0.00 / -	0.00 / -	0.21 / -
PERACT + F	0.68 / -	0.40 / -	**0.60** / -	0.04 / -	0.08 / -	0.32 / -	0.00 / -	0.42 / -
BC-Z + F	0.88 / 0.44	0.16 / 0.29	0.36 / 0.48	0.04 / 0.43	0.16 / 0.47	0.08 / 0.35	0.00 / 0.09	0.26 / 0.39
Our w/o Pr	0.88 / 0.58	0.28 / **0.59**	0.28 / 0.74	0.04 / **0.69**	**0.20** / 0.90	**0.44** / 0.77	0.00 / 0.22	0.34 / 0.76
Ours	**0.92** / **0.86**	**0.44** / 0.55	0.40 / **0.78**	0.04 / 0.66	**0.20** / **0.92**	**0.44** / **0.91**	0.00 / **0.23**	**0.46** / **0.80**

By utilizing the perception and action-prompts, the agent can better leverage cross-modal action knowledge, which is beneficial for guiding correct interaction.

To make each fine-grained language instruction have a closer connection with the corresponding affordance maps and actions, following [13], we introduce the multi-modal alignment loss $\mathcal{L}_{mm}$. Following InfoNCE [3] loss, $\mathcal{L}_{mm}$ aligns the concatenation of the action feature and visual scene feature $(f_{pr_{action}}, f_{pr_{vision}})$ with the instruction feature $f_{pr_{verb}}$. This alignment encourages a multi-modal correspondence between paired manipulation actions and language:

$$\mathcal{L}_{mm} = -log(\frac{exp(f_{pr_{action\&vision}} \cdot f_{pr_{verb_+}}/\tau)}{\sum_{k=0}^{K} exp(f_{pr_{action\&vision}} \cdot f_{pr_{verb_k}}/\tau)}) \quad (2)$$

, where $f_{pr_{action\&vision}}$ donates the concatenation of $f_{pr_{action}}$ and $f_{pr_{vision}}$, $+$ denotes positive pair, k denotes the number of all samples.

Besides, two fine-grained instructions with the same noun or verb usually lead to similar manipulation, either similar contact regions or similar actions. For example, to "grasp" an umbrella handle or to "grasp" an item in the drawer, the agent should take a similar action: move forward until it reaches the target point and then close the two gripper fingers. Therefore, to encourage the agent to focus on related actions or target object parts, we introduce the consistency loss $\mathcal{L}_{C_{verb}}$ and $\mathcal{L}_{C_{noun}}$. The $\mathcal{L}_{C_{verb}}$ aims to make the features $f_{pr_{verb1}}$, $f_{pr_{verb2}}$ of two instructions with the same verb closer:

$$\mathcal{L}_{C_{verb}} = |f_{pr_{verb1}} - f_{pr_{verb2}}| . \quad (3)$$

The $\mathcal{L}_{C_{noun}}$ aims to make the features of two instructions with the same noun closer, and the loss function is similar to $\mathcal{L}_{C_{verb}}$. If two fine-grained instructions share the same noun, the two predicted affordances f_{aff1} and f_{aff2} are supposed to highlight similar regions. For example, to grasp the door handle and to grasp the drawer handle, though the two handles probably have different geometric shapes, the two affordance maps should both highlight the two handles. To achieve this, we use the consistency loss $\mathcal{L}_{C_{aff}}$ to make the two affordance map features closer:

$$\mathcal{L}_{C_{aff}} = |f_{aff1} - f_{aff2}| . \quad (4)$$

By using these loss functions, we can effectively align the different modalities and improve the agent's understanding and execution of the task.

V. EXPERIMENTS

A. Experiment details

We conduct our experiments in the simulation environment V-REP [21], interfaced with PyRep [7], and use a Franka Panda robotic arm equipped with a parallel gripper. In the benchmark, for each task, we offer comprehensive scene information along with demonstration waypoints and a script to link the scene objects to the RLBench backend. It implements task variations, specifying success criteria, and incorporating additional intricate task behaviors. Input observations are acquired from four RGB-D cameras strategically positioned: one at the robot's front, one at the left shoulder, one at the right shoulder, and another on the robot's wrist. It is noteworthy that all these cameras are devoid of any noise and boast a resolution of 128 × 128 pixels.

B. Baseline Comparision and Ablation Study

a) Evaluation Metric: We evaluate each multi-task agent independently on 8 training tasks and 7 novel tasks, with each task consisting of 25 evaluation episodes, totaling 375 episodes. We introduce two metrics to quantitatively evaluate the methods. (1) To evaluate the quality of the action proposals, we report the average manipulation success rates per task. During evaluation, the agent continues taking actions until an oracle indicates task completion or until it reaches a maximum of 25 steps. (2) To evaluate the effectiveness of the Instruction Selection network in understanding the fine-grained instruction sequence based on the current observation, we introduce the metric Instruction Following. During inference, we feed all steps in those 350 testing episodes to the Instruction Selection network and calculate the accuracy of selecting the correct fine-grained instruction.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

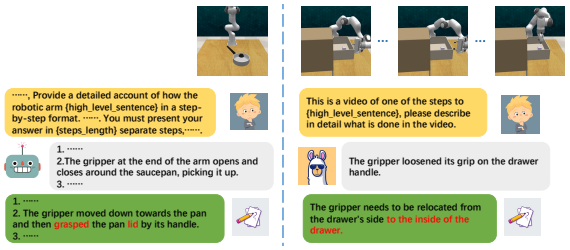

Fig. 4. Instruction generating process for two large models (Minigpt4 on the left). The red text in the green box is the important element missed by the large models, "high_level_sentence" is the high-level instruction of the current task, and "steps_length" is the total number of steps.

b) Baselines: We compare our approach with four baselines and one ablated version:

- **I-BC (Image-BC)**, an image-to-action agent similar to BC-Z [9]. Following BC-Z, we use FiLM [18] for conditioning with CLIP [20] language features. However, the vision encoders take in RGB-D images instead of RGB, and the vision encoder is implemented as a CNN. This baseline only takes in high-level instructions without fine-grained ones.
- **PERACT**, a behavior-cloning agent for multi-task 6-DoF manipulation. It encodes language goals and RGB-D voxel observations with a Perceiver Transformer. This baseline also only takes in high-level instructions without fine-grained ones.
- **I-BC + F**, which is based on I-BC. This baseline takes in additional fine-grained instructions, and we build an Instruction Selection network for it to select the suitable instruction according to the current scene.
- **PERACT + F**, which is based on PerAct. This baseline also takes in additional fine-grained instructions.
- **Our w/o Pr**: an ablated version of our method that removes the Prompt Module.

Table. II shows the task success rates (before slashes). We observe that our method outperforms all counterparts. Compared to pure high-level instructions, the fine-grained instructions can improve the manipulation accuracy since they guide the agent to complete the overall tasks step by step. Additionally, the ablation study demonstrates that the introduction of the Prompt module further boosts performance by aligning features between different modalities and providing perception and action knowledge.

Table. II also shows the Instruction Following numbers (after slashes). We find that our method can choose the right fine-grained instruction based on the current scene, which demonstrates the effectiveness of our approach in understanding and executing fine-grained instructions.

C. Instruction Generation

Large language models have revolutionized text annotation by automating labor-intensive and repetitive tasks, significantly reducing human effort. This innovation has accelerated the traditionally time-consuming process, making it more efficient. These models are also highly adaptable to specific tasks without requiring retraining, showcasing their versatility across various domains. Moreover, they excel in minimizing annotation errors, thereby elevating data quality and boosting the performance of machine learning and natural language processing applications.

TABLE III
QUANTITATIVE RESULTS OF THE LANGUAGE INSTRUCTIONS GENERATED BY LARGE LANGUAGE MODELS, DEMONSTRATING THE VALUE OF HUMAN-ANNOTATED FINE-GRAINED INSTRUCTIONS.

Method	Bleu score	Rouge score
Minigpt4	11.94	30.48
Video-LLaMA	7.11	36.74
Video-LLaMA (fine-tune)	15.07	34.74

a) Models: To evaluate the effectiveness of large language models in offering precise instructions for our NrVLM dataset, we conduct experiments employing two distinct models: Minigpt4 [31] and Video-LLaMA [29]. Specifically, Minigpt4 is utilized for tasks involving image-based question and answer tasks. For this, we select an RGB image captured from a frontal perspective of each manipulation episode, with the corresponding question template depicted in Fig. 4. On the other hand, the Video-LLaMA model adopts video-based question and answer tasks. Here, we input the video corresponding to each manipulation step to derive answers based on the question templates showcased in Fig. 4. To enhance the Video-LLaMA model's suitability for manipulation tasks, we fine-tune it using the training set tasks from our benchmark.

b) Metrics: We adopt two primary metrics in natural language processing, namely Bleu (Bilingual Evaluation Understudy) [1] and Rouge (Recall-Oriented Understudy for Gisting Evaluation) [14]. The Bleu score plays a pivotal role in assessing the accuracy of machine-generated translations, while the Rouge score quantifies the overlap between machine-generated summaries and human reference summaries. In our experiments, manually labeled language instructions serve as ground truth, and these two metrics are employed to assess the quality of the results generated by the two large language models.

c) Analysis: The Bleu and Rouge scores are both in the range of 0 to 100, and the results in Table.III indicate that both Minigpt4 and Video-LLaMA's generated results perform very poorly in terms of accuracy and recall. After fine-tuning on the manipulation tasks in our training set, Video-LLaMA's generated results can achieve a little gain on Bleu compared to the original Video-LLaMA, but the overall quality of the generation is still very low. These outcomes clearly emphasize that although the large language model demonstrates a degree of understanding when presented with input images or videos, its accuracy notably lags behind in generating manipulation outputs, let alone serving as instructions for guiding agent movement. As a result, the manually curated fine-grained natural instructions in our NrVLM benchmark hold significant value and necessity, particularly in tasks that involve intricate planning and precise manipulation.

VI. CONCLUSIONS

In this paper, we introduce the NrVLM benchmark, offering detailed natural instructions to assist agents in executing complex tasks sequentially. Alongside this benchmark, we present a framework for instructions following and establishing a manipulation-aware multi-modality alignment, improving the accuracy of manipulation. Our work shows that fine-grained instructions significantly enhance the learning and performance of robot policies. We recommend fine-grained instructions inclusion in future robotic datasets to support more research in training agents.

VII. ACKNOWLEDGEMENTS.

This project was supported the National Youth Talent Support Program (8200800081) and National Natural Science Foundation of China (No. 62136001).

REFERENCES

[1] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Frederick Jelinek, John Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85, 1990.

[2] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Manipulathor: A framework for visual object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4497–4506, 2021.

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[4] Siyuan Huang, Zhengkai Jiang, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model. *arXiv preprint arXiv:2305.11176*, 2023.

[5] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[6] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[7] Stephen James, Marc Freese, and Andrew J Davison. Pyrep: Bringing v-rep to deep robot learning. *arXiv preprint arXiv:1906.11176*, 2019.

[8] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[9] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[10] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[11] Chuhao Jin, Wenhui Tan, Jiange Yang, Bei Liu, Ruihua Song, Limin Wang, and Jianlong Fu. Alphablock: Embodied finetuning for vision-language reasoning in robot manipulation. *arXiv preprint arXiv:2305.18898*, 2023.

[12] Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 4613–4619. IEEE, 2021.

[13] Bingqian Lin, Yi Zhu, Zicong Chen, Xiwen Liang, Jianzhuang Liu, and Xiaodan Liang. Adapt: Vision-language navigation with modality-aligned action prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15396–15406, 2022.

[14] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[15] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[16] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[17] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019.

[18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[21] Eric Rohmer, Surya PN Singh, and Marc Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pages 1321–1326. IEEE, 2013.

[22] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[23] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[24] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter 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.

[25] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022.

[26] Thomas Weng, Sujay Man Bajracharya, Yufei Wang, Khush Agrawal, and David Held. Fabricflownet: Bimanual cloth manipulation with a flow-based policy. In *Conference on Robot Learning*, pages 192–202. PMLR, 2022.

[27] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[28] Hanbo Zhang, Yunfan Lu, Cunjun Yu, David Hsu, Xuguang La, and Nanning Zheng. Invigorate: Interactive visual grounding and grasping in clutter. *arXiv preprint arXiv:2108.11092*, 2021.

[29] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[30] Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Wang. Vlmbench: A compositional benchmark for vision-and-language manipulation. *Advances in Neural Information Processing Systems*, 35:665–678, 2022.

[31] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.