ACTLLM: Action Consistency Tuned Large Language Model

Jing Bi[1], Lianggong Bruce Wen[2], Zhang Liu[2], Chenliang Xu[1]

***Abstract*— This paper introduces ACTLLM (Action Consistency Tuned Large Language Model), a novel approach for robot manipulation in dynamic environments. Traditional vision-based systems often struggle to learn visual representations that excel in both task execution and spatial reasoning, thereby limiting their adaptability in dynamic environments. ACTLLM addresses these challenges by harnessing language to craft structured scene descriptors, providing a uniform interface for both spatial understanding and task performance through flexible language instructions. Moreover, we introduce a novel action consistency constraint that aligns visual perception with corresponding actions, thereby enhancing the learning of actionable visual representations. Additionally, we have reformulated the Markov decision process for manipulation tasks into a multi-turn visual dialogue framework. This approach enables the modeling of long-term task execution with enhanced contextual relevance derived from the history of task execution. During our evaluation, ACTLLM excels in diverse scenarios, proving its effectiveness on challenging vision-based robot manipulation tasks.**

I. Introduction

Adapting to diverse and dynamic environments, while executing flexible task specifications, remains a significant challenge in robot manipulation methods. Language-based vision manipulation systems offer a promising solution by leveraging rich visual information and language to better comprehend varying environmental conditions and task specifications [1], [2], [3], [4]. These systems typically comprise two key components: a vision component that associates concepts from language instructions with visual information, and a policy module that generates actions based on the outputs from the vision component. Recent end-to-end frameworks [2], [5], [6], [7], [8] attempt to leverage affordances to integrate semantic meanings with visual information. This integration helps robots identify feasible actions within specific physical contexts, addressing the question of which actions are possible and where they can be performed in a given scene by merging linguistic context with visual cues. In contrast, another line of research, exemplified by CALVIN[4] advocates for a transition towards adaptable, task-agnostic manipulation policies that employ general, unstructured language to define tasks. This approach allows for more generalized policies as the broad language specifications enable versatility in task execution.

However, existing research frequently separates the learning processes of vision models and action policies, with each component being optimized independently. This division often results in perception models often provides representations that are not optimized to be effectively used by the policy models. Moreover, the action policies face the challenge of learning a broad spectrum of skills. This complexity arises from the necessity to interpret open-ended language instructions and effectively leverage visual representations, thus demanding more sophisticated and integrated learning approaches. Therefore, achieving more adaptable policies that associate visual observations with linguistic concepts for robot manipulation presents a two-stage challenge: (i) Enhancing the alignment between the vision and language concepts. (ii) Optimizing the integrated information from both modalities for action policy.

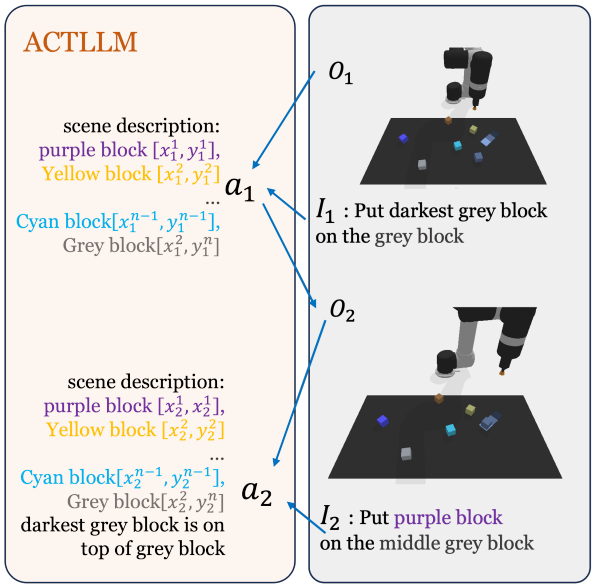

Fig. 1. We reformulated the Markov decision process for manipulation tasks into a multi-turn visual dialogue framework, wherein the model generates descriptions of the current view and potential future states, based on given instructions and history observations. From these descriptions, actions are generated to accomplish the tasks. This approach ensures that the actions generated between scene descriptions are consistent with the changes in the scene, empowering Large Language Models (LLMs) to analyze spatial and temporal relationships within the manipulation trajectory. This facilitates the efficient execution of complex tasks.

Due to the strong zero-shot performance of foundation models, several works have integrated these models into robot applications to overcome the first challenge. These methods encompass various tasks ranging from generating natural language scene descriptions [9], leveraging pretrained representation for reactive control [10], grounding actions to symbolic representation for test-time adaptation[11], [12], [13] to automating dense reward generation [14], [15], [16],

This work is a work in progress.

[1]University of Rochester, Rochester, NY, USA. Email: jing.bi@rochester.edu, chenliang.xu@rochester.edu

[2]Corning Inc., Corning, NY, USA. Email: {wenl, liuzh3}@corning.com

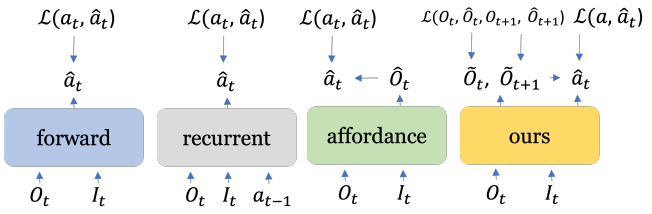

Fig. 2. In comparison to previous models: (a) Traditional forward models use current observations and instructions to output actions, with potential enhancements from additional historical information. (b) Affordance methods differ by translating observations into heatmaps for simpler action computation via argmax. (c) Our approach advances these concepts by generating actions from LLMs, utilizing text-based scene descriptions to regularize and improve the accuracy of the action embeddings.

[17], [14]. Moreover, several methods have adapted foundational model architectures to address the second challenge. For instance, the ReAct [18] planner generates conditional sequences that guide low-level policy actions. PaLM-E [6] integrates multimodal inputs—including vision, text, and state estimation—to produce low-level instructional texts to drive the controller. However, previous methods often treat task specification and environment understanding as distinct challenges, a separation that can hinder the integration of visual concepts with their semantic meanings, thereby creating barriers to effectively instructing robots. Moreover, visual observations are frequently processed into a hidden representation, which is not only difficult to interpret but also challenging to be optimized for policy.

To overcome these challenges, we introduce ACTLLM, a method that unifies the interpretation of visual information with policy learning. Our approach leverages structured scene descriptions that are not only more accessible for humans but also allow us to construct a novel loss functions to optimize the model more effectively. By incorporating an action consistency constraint loss, we jointly optimize the action policy with scene description generation, significantly enhancing the fusion of these elements and addressing the aforementioned issues. Furthermore, following proposed method, we are able to reformulated the Markov decision process for manipulation tasks into a multi-turn visual dialogue framework to help long-term task learning. To summarize, our contributions lay in three folds:

Structured scene description: By explicitly representing observations as structured scene descriptions, we can integrate spatial reasoning with instruction understanding, creating actionable representations that merge language comprehension with visual features.

Noval Approach to policy optimization: Building on the foundation of structured scene descriptions, we propose a novel loss function. This enhances the integration of information from both textual instructions and visual observations, optimizing policy learning in dynamic environments.

Enhancing LLM tuning for manipulation ACTLLM transforms traditional Markov Decision Processes into a visual multi-turn dialogue framework. This enables a novel method for enhancing LLM fine-tuning, incorporating manipulation control data to refine decision-making processes.

II. Related work

A. Vision for Manipulation

Traditional methods for robot perception have primarily relied on explicit 'object' representations, such as instance segmentation, object classes, and poses [19], [20]. However, these methods encounter challenges when dealing with deformable and granular items like clothing and beans, which are difficult to characterize using geometric models or segmentations. To overcome these limitations, recent approaches [21], [22] have begun to make fewer assumptions about objects and tasks, often framing the problem as an image-to-action prediction task. Nevertheless, direct training on RGB images for tasks with 6 Degrees of Freedom (6-DoF) tends to be inefficient, typically necessitating numerous demonstrations or episodes to acquire basic skills like object rearrangement. In response to these challenges, several methods have emerged. For example, in 3D environments, C2FARM [23] presents an action-centric reinforcement learning (RL) agent with a coarse-to-fine-grain 3D-UNet backbone. However, this approach has a limited receptive field at the finest level, preventing it from encompassing the entire scene. Another line of research, exemplified by approaches like [24], [2], [25], focuses on learning action-centric representations with affordances, emphasizing the interaction aspects of objects. In contrast to these methods, our model relies solely on RGB images without the need for special modeling of "objects," as in previous works.

B. Language empowered Robotics

Instruction-based policies have been a popular area of research in robotics [26], [27], [28]. A notable development is CLIPORT [29], which enhances Transporter [24] with semantic understanding and object manipulation abilities via CLIP text encoding [30]. This concept was further expanded to 3D environments in the Perceiver-Actor model, utilizing voxelized observation and 3D action spaces [31]. Another innovative application of LLMs in robotics involves code generation for action policy[32], enabling robots to utilize vision APIs for tasks like segmentation, detection, and internet access. Instruct2Act [9] exemplifies this by integrating robotic skills with LLMs, striking a balance between flexibility and expertise, and demonstrating high performance in zero-shot settings. Language also plays a crucial role in high-level robotic planning [33], [34], [35] and low-level policy development, with model-based approaches gaining traction [36]. Notably, CaP [32] and Socratic Models [37] have made significant strides by generating detailed policy codes and incorporating perceptual data into LLMs, respectively. Additionally, LLMs are being utilized as a source of reward or feedback in robotic systems, with methods like those by [34] and [38] enhancing robotic operations through closed-loop feedback and reinforcement learning. PAFF [11] tackles the challenge by utilizing feedback from foundation models via a Hindsight Experience Replay process, enabling the model to respond to randomly generated instructions in unfamiliar environments, with actions subsequently collected

and relabeled to fine-tune the foundation model for test-time adaptation. VIMA [1] proposes the VisuoMotor Attention agent to solve robot manipulation from multimodal prompts with a Transformer Encoder-Decoder architecture.

III. Method

A. Preliminary

We have access to a dataset $\mathscr{D} = \{\zeta_1, \zeta_2, \ldots, \zeta_n\}$ of n expert demonstrations trajectory ζ with associated discrete-time input-action pairs $\zeta_i = \{(o_1, I_1, a_1), (o_2, I_2, a_2), \ldots\}$. Notably, the instruction I for each step may consist of text only, images only, or a combination of both as shown in Figure 3 The action space consists of primitive motor skills, including actions such as *pick and place*, *wipe*, *push* [24]. Additionally, for each action, two positional vectors indicate the initial and target poses, denoted as $(p_{\text{initial}}, p_{\text{target}})$.

Our goal is to learn a policy model π that can effectively generate an action a_t to accomplish provided multimodal instruction I based on the current execution history. Inspired by recent advancements in imitation learning literacy [39], we designed our policy as $\hat{a} = \pi_\theta(x_t, x_{t+1})$. This all us to deduce the most likely action that transitions the agent from its current state to a subsequent state, where x_t and x_{t+1} represent the states extracted from observations. For a one-step trajectory, represented as (o_t, a_t, o_{t+1}, I_t), we will map observations to their respective state representations through an observation model, yielding $x_t = \mathscr{M}(o_t), x_{t+1} = \mathscr{M}(o_{t+1})$. During inference, to enable the policy leverage the future state x_{t+1}, we need to incorporate a forward model for forecasting future states given the current observation and instruction, formalized as $\tilde{x}_{t+1} = \mathscr{F}_\beta(o_t, I_t)$. The essence of this formulation lies in the focus on the desired final state rather than the specific steps taken to achieve it, as multiple paths can lead to the same outcome. In practical, this approach means that the model does not strictly penalize deviations from the ground-truth actions, as long as the alternative actions proposed lead to the same expected subsequent states.

B. Structure scene description

In optimizing our model, the key lies in the choice of state representation for both $\mathscr{M}$ and prediction $\mathscr{F}$ model. Our key insight for utilizing LLM as both a model $\mathscr{M}$ and a function $\mathscr{F}$, which not only enhances model optimization but also possesses a stable ground-truth, as previous approaches using hidden vectors for state representation can be unstable, particularly during the early training stages.

To enable LLM to provide structured representations rather than unstructured language, we leveraged predefined JSON schema to manipulate the decoding phase. During this phase, we pre-fill the fixed tokens of the data schema and delegate the generation of content tokens solely to the language model. This approach ensures that the structure remains consistent while the model dynamically fill the content. Therefore, we define the state representation as a set of $\{(\mathfrak{o}_1, \mathfrak{c}_1, \mathbf{p}_1), (\mathfrak{o}_2, \mathfrak{c}_2, \mathbf{p}_2), (\mathfrak{o}_3, \mathfrak{c}_3, \mathbf{p}_3), \ldots\}$, where $\mathfrak{o}$, $\mathfrak{c}$ and $\mathbf{p}$ represent the object, color, and coordinates, respectively.

This design offers several benefits: (i) interpretability: this approach simplifies the understanding and validation of the model's decision-making process, making it more transparent how and why certain conclusions were reached. (ii) clarity: we can formulate a clear loss function based on the state representation, rather than relying on ambiguous regression losses associated with hidden states. (iii) stable ground-truth: by leveraging access to the simulator's internal state, we can precisely obtain the condition of all objects. To enable the LLMs to describe the scene with coordinates, we adopt the center position of the bounding box to denote the object's location. Coordinates are normalized within a [0, 1] range and maintained to two decimal places; [0.50,0.50] indicates that the object is at the center of the view. Since we would like to enlarge the token space of the LLM, we integrate the Adapter mechanism [40] to model the spatial tokens. Once we have the scene decription, we can compute the loss as $\mathscr{L}(x_t, x_{t+1}, \tilde{x}_{t+1}, \tilde{x}_t)$

We adopt the approach from GLIP [41], employing contrastive loss to compare predicted objects with ground truth text tokens. This involves directly matching predicted objects with their corresponding textual descriptors. The process includes: (i) For each predicted object (query), we compute a dot product with the feature vectors of the text descriptions to produce logits corresponding to each text token, followed by applying focal loss to each logit. (ii) Since the set of objects lacks inherent order, we use box regression and classification costs to perform bipartite matching between the predictions and the ground truth data. (iii) Finally, we calculate the loss values for the matched predictions and their corresponding ground truths, ensuring precise alignment and accuracy.

By aligning both object properties and their locations with language tokens, we can enhance the model's reliability and accuracy in scene prediction tasks.

C. Action consistency loss

We now focus on how to generate the action $a_t = \pi_\theta(x_t, x_{t+1})$ based on the state representation. Concatenating another language model to predict actions based on generated or ground-truth state representations seems intuitive. However, it could introduce more drawbacks than benefits. This is primarily due to potential inaccuracies in scene descriptions during initial training stages, leading to noise in policy gradient updates. Therefore, we propose constructing the policy by aggregating the sequence of token embeddings of the state representation using a learnable parameter W with an attention module

$$Q = EW, \quad x_{\text{agg}} = \text{softmax}(QQ^T)Q$$
$$\hat{a}_t = \text{MLP}(x_{\text{agg}})$$

where E represents the sequence of token embeddings of x_t, x_{t+1}, $E = [e_1, e_2, \ldots, e_n]$, and n is the number of tokens in the sequence. Through this method, actions are derived from an aggregated token that adeptly encapsulates a sentence-level representation of the state. Moreover, due to its lightweight design, the model exhibits better adaptability,

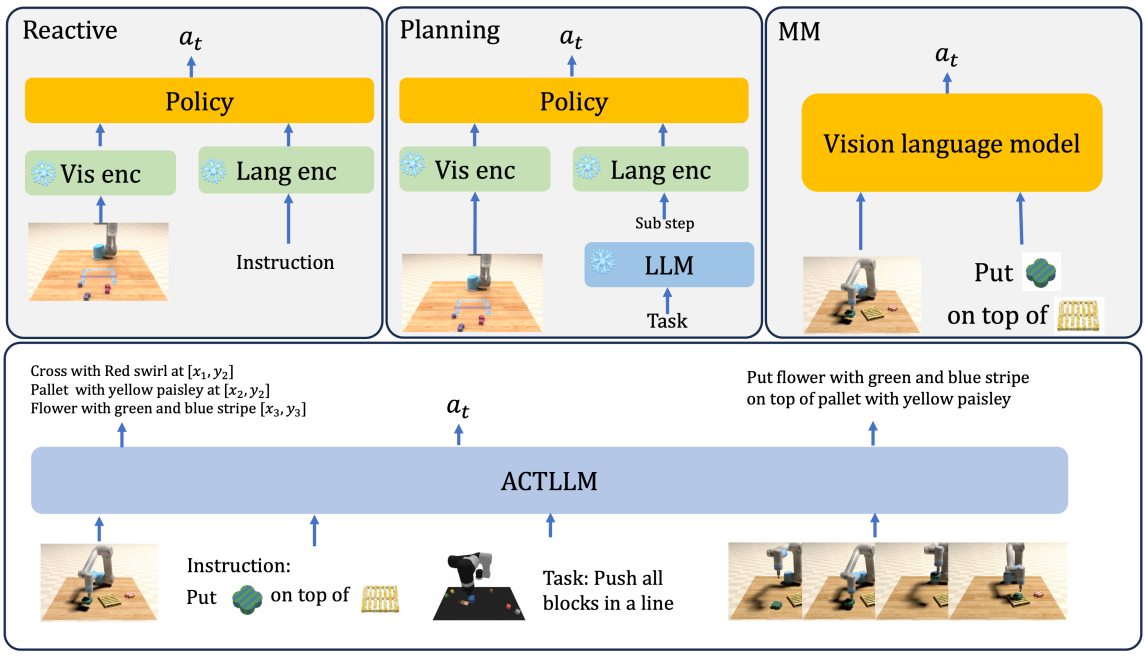

Fig. 3. Comparison among our vision-language manipulation approach and existing solutions: **Reactive** indicates utilizing foundation models to extract features for following action generation [10]. In contrast, **Planning** strategies typically employ LLM to decompose tasks into sub-steps and solve them. **MM** means works directly fine-tuning vision-language models for robotic manipulation, as seen in [1], [3] directly fine-tune a vision-language model for robot manipulation. Our model can process both visual and language tokens as task-specific inputs, allowing us to generate actions that are more consistent and accurately correspond to the scene description.

which is crucial to smoothly accommodate improvements in state representations over time.

Once we have one step tuple (s_t, a_t, s_{t+1}, I_t), we can design the optimized function to align the action with the state transition. The joint objective for training is formalized as:

$$\begin{aligned} \min_{\theta,\beta} \; & \mathscr{L}(x_t, x_{t+1}, \tilde{x}_{t+1}, \tilde{x}_t) + \mathscr{L}(a_t, \hat{a}_t), \\ \text{s.t.} \quad & \tilde{x}_t, \tilde{x}_{t+1} = \mathscr{F}_\beta(o_t, I_t) \\ & \hat{a}_t = \pi_\theta(x_t, x_{t+1}) \end{aligned} \tag{1}$$

where the action loss includes both the classification loss of primitive skill and the loss of the scene description. As shown in Figure 4, the action consistency loss effectively harnesses additional information to more accurately align action and visual data together. The advantage of this method of action generation is that it tightly integrates the action with the scene description, ensuring that the action is more closely associated with the context within the task and scene. Moreover, by employing multi-turn tuning as detailed below, the action benefits from its correspondence to scene changes. Consequently, changes within the scene also influence how the action is generated.

D. Markov decision making as visual dialogue

Recent advancements, as highlighted in vicuna fintuning [42], demonstrate that multi-turn interactions significantly enhance Large Language Models (LLMs) by enabling more complex and context-rich dialogues. Unlike single-turn dialogues, which generate responses without considering past interactions, multi-turn conversations draw upon previous exchanges, allowing the model to gain a comprehensive understanding of the context and the user's goals.

Expanding on the previously introduced concept, we propose a straightforward extension of our one-step optimization to a variable-length sequence of actions as shown in Figure 1 This framework involves multi-turn dialogue tuning, where each step includes a current observation and instruction, leading to a unique state. The model needs to select the most suitable action based on the current state and overall task goal, incorporating the historical context of predicted states.

The joint loss is calculated at each time step and is optimized alongside the action prediction loss across the entire trajectory. The final multi-step objective with feature space dynamics is defined as follows:

$$\begin{aligned} \min_{\theta,\beta} \; & \sum_{t=1}^{T-1} \Big(\mathscr{L}(x_t, x_{t+1}, \tilde{x}_{t+1}, \tilde{x}_t) + \mathscr{L}(a_t, \hat{a}_t) \Big), \\ \text{s.t.} \quad & \tilde{x}_t, \tilde{x}_{t+1} = \mathscr{F}_\beta(o_t, I_t) \\ & \hat{a}_t = \pi_\theta(x_t, x_{t+1}) \end{aligned} \tag{2}$$

This method allows the model to learn and adapt over time, fostering a more advanced and user-focused conversational experience. It enables users to switch tasks or offer prompt interventions in between, enhancing interaction quality and effectiveness. As illustrated in Figure 3, compared with other methods, our proposed model can process various types of information and generate essential output for robot manipulation. One advantage of having a long context for the trajectory is that it ensures better alignment between the action and scene descriptions. Specifically, when tuning on the

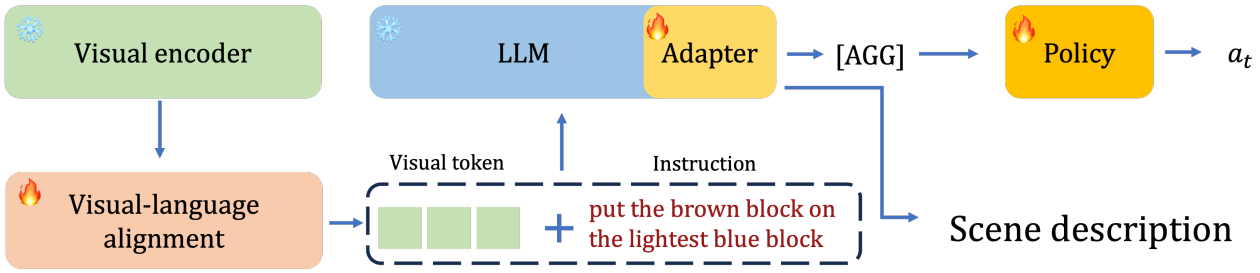

Fig. 4. The illustration of our model. Our model consists of three main components: a visual encoder, a visual-language alignment layer, and a decoder-only LLM. The coordinates of a bounding box are converted into texts in a specific format. During training, we freeze the visual encoder and LLM and only update the Adapters and the alignment layer. Upon receiving an instruction, the LLM initially produces an aggregated token. This token informs the policy to generate the corresponding action for this step and also aids description of the environment.

step tuple (s_t, a_t, s_{t+1}, I_t), the action and scene descriptions are closely associated with the current state. However, when optimizing for multi-turn interactions, the action generation can leverage the extended context as cues. This allows for actions that consider a more comprehensive history.

E. Implementation details

Our model architecture is composed of two components (i) a MLLM serves as both $\mathscr{F}$ and $\mathscr{M}$. This component encompasses a visual encoder Φ_V, a projection layer Φ_P, and a Language Model Φ_L. (ii) a policy head π, which includes an attention module and an MLP. The visual encoder is utilized to transform all images found in instructions or observations into a sequence of visual tokens, denoted as $Z_V = \Phi_V(I)$. To adapt this representation for use in the LLM, a linear layer Φ_P is employed, converting Z_V into the LLM's input space, leading to $Z_T = \Phi_P(Z_V)$. Subsequently, both Z_T and Q_T are concatenated and passed through Φ_L to produce the scene description x_t, x_{t+1}. Finally, the policy head π processes the scene description tokens and translates them into actions. As discussed in [43], fine-tuning the visual encoder, even with a modest visual instruction tuning dataset, can lead to semantic loss. This loss adversely affects the image representation capabilities of the visual encoder. Consequently, we opt to keep the visual encoder frozen, especially considering that simulations might exacerbate semantic loss. We incorporate the LLAMA3-8B [44] model, as our LLM, specifically tuned for following instructions. To bridge visual tokens with the static LLM, we utilize an alignment layer. This is supplemented by an action projection module, which is crucial for converting model insights into actionable outputs. AdamW is selected as the optimizer. Further details will be provided in the supplementary material.

IV. Experiment

Our experiments focus on two benchmark (i) VIMA-BENCH: A newly introduced task suite and benchmark designed to facilitate the learning of general robot manipulation through multimodal prompts. (ii) CLIPORT: This dataset offers step-level text instructions paired with top-down RGB-D observations, serving as a platform for learning multi-task manipulation. We carefully design evaluations to evaluate (i) Compositional Generalization which assesses the model's proficiency in handling objects with new shapes, colors, and entirely novel objects. (ii) In-context Generalization, where the model is evaluated with the novel tasks Note that during the testing stage, we do not provide a state representation as input to the model. Therefore, we do not need to rely on any object detector to provide a structured screen description, as it is generated by the model.

A. Compositional Generalization

Compositional generalization evaluates a model's ability to apply learned knowledge to new combinations of familiar elements or concepts. We demonstrate this in the both CLIPORT and VIMA-BENCH environment.

In the CLIPORT environment, we follow the PAFF framework [11]. We report the task success rate across 100 evaluation instances in 10 diverse scenes. These scenes feature various blocks, objects, and bowls, thereby testing the model's capacity to accurately place objects. We adopted two evaluation protocols as follows: 1) `pack-unseen-objects`: In this protocol, we train a policy to pack objects of different shapes into a brown box ('pack-shapes') and then evaluate its ability to 'pack-unseen-objects.' 2) `put-shapes-in-bowls` Another evaluation involves placing blocks of different colors into bowls of different colors ('put-blocks-in-bowls') then we ask the model to put objects of different shapes into bowls of different colors. We included all baseline models from PAFF, where MdetrORT [45] is a variation of CLIPORT [2] by replacing the visual and language encoder, and AugORT [46] include more data augmentation to the MdetrORT.

TABLE I

Results from the compositional and out-of-distribution generalization evaluations on the CLIPORT platform are presented. The primary metric for evaluation is the success rate. Each step is provided with a new instruction in the left column; a subsequent instruction is issued only after the previous one has been successfully executed.

Method	put-shapes-in-bowls		pack-unseen-objects	
CLIPORT	28.0%	16.8%	58.9%	46.1%
MdetrORT	33.8%	17.8%	62.0%	48.4%
AugORT	34.4%	18.9%	63.1%	49.0%
PAFF	**51.0%**	**35.0%**	**72.8%**	**63.8%**
ACTLLM	**64.0%**	**66.2%**	**85.8%**	**79.6%**

TABLE II

WE CONDUCTED A COMPARATIVE ANALYSIS OF OUR METHODS USING THE VIMA-BENCH EVALUATION ACROSS FOUR DISTINCT LEVELS. THE 'AVG' REPRESENTS THE AVERAGE SUCCESS RATE FOR ALL TASKS WITHIN EACH LEVEL. TO ASCERTAIN THE SUCCESS RATE FOR EACH METHOD, WE SAMPLED 200 EPISODES FROM EACH TASK. OUR METHODS DEMONSTRATED SIGNIFICANT IMPROVEMENTS OVER BASELINE APPROACHES.

	L1					L2					L3					L4	
Method	Avg	T5	T9	T16	T17	Avg	T5	T9	T16	T17	Avg	T5	T9	T16	T17	Avg	T10
Gato	57.0	44.5	14.0	43.0	1.5	53.9	46.0	10.5	42.0	1.0	45.6	36.0	17.0	41.5	0.0	13.5	0.0
Flamingo	47.2	41.0	3.0	38.0	2.0	47.1	43.0	4.5	40.0	1.0	42.1	36.5	6.0	45.5	0.5	11.1	0.0
GPT	47.9	45.0	8.0	33.0	1.0	47.4	43.0	10.5	34.0	3.0	42.6	32.0	5.0	37.5	0.0	12.1	0.5
VIMA	87.2	65.0	13.5	88.0	77.0	87.0	61.0	12.5	87.5	77.5	84.0	63.0	12.0	58.5	78.0	49.6	0.0
ACTLLM	**90.5**	**78.3**	**65**	**96.0**	**86.0**	**90.9**	**78.2**	**61.5**	**93.0**	**84.3**	**93.4**	**84.0**	**70.9**	**88.2**	**87.0**	**64.8**	**12.1**

We present the evaluation results of compositional generalization in Table I. Our method significantly outperforms the baseline across both evaluation protocols by a substantial margin, demonstrating the efficacy of our scene representation model in achieving superior compositional generalization. Unlike PAFF, ACTLLM takes a comprehensive approach that goes beyond focusing solely on task-relevant objects; it considers all objects present. This broader perspective equips it to excel in tasks involving novel objects and shapes. Furthermore, ACTLLM can accurately generate scene descriptions containing various concepts, such as objects and containers, within a compositional setting. Additionally, it can learn object-aware representations without the need for specific objects and associated bounding boxes.

In VIMA-BENCH, the compositional generalization evaluation is more challenging can be broken down into three levels: 1) **Placement:** During training, all prompts are encountered, and only the placement of objects on the tabletop is randomized at testing. 2) **Combinatorial:** In the training phase, all textures and objects are familiar, but during testing, new combinations of these elements are introduced. 3) **Novel Object:** Both the test prompts and the simulated workspace feature novel textures and objects that were not encountered during training. We include results for models such as Gato [47], Flamingo [48], and GPT [1] as reported directly within the VIMA paper. The results of the evaluation for various levels of evaluation protocols are presented in Table II. Since we have limited space, we only report the success rates of representative tasks for which different methods show significant performance differences. The average (Avg) indicates the success rate of all tasks at a particular evaluation level. We can see that our methods outperform all baseline methods in the first three levels, which require the learning of new combinations of familiar elements or concepts.

B. In-context Generalization

This challenge is newly introduced in VIMA, where new tasks are defined by novel prompt templates during test phases. These templates encompass not only novel actions but also novel objects that were not present in the training data. Our evaluation follows a two-fold approach: first, we follow the VIMA protocol, which involves directly executing Level 4 tasks, as shown in Table II; second, we use in-context demonstration videos that convey the essence of the tasks during testing. The inclusion of in-context learning, specifically during testing phases, is aimed at examining the model's zero-shot adaptation capabilities. This approach underscores the importance of providing models with dynamic, real-world contexts that enhance their learning and adaptation processes. Following VIMA, we save T9: Twist T10: Follow Motion T11: Follow Order as a novel task. We then conducted an ablation study to assess the effectiveness of various modules.

TABLE III

EVALUATING THE IN-CONTEXT GENERALIZATION CAPABILITY OF THE ACTLLM WITH *Twist* AND *Follow Order* AS NOVEL TESTING TASKS.

Task	T9	T10	T11	Overall
Our Method	**20.1**%	**63.4**%	**10.0**%	**36.2**%
w/o task tuning	14.5%	38.9 %	6%	26.7%
w/o future state	12.4%	23.9 %	4.7%	22.3%
w/o multi-turn	16.1%	29.9 %	3.5%	24.3%

As shown in table III, removing task tuning significantly reduces performance across all tasks, leading to a marked decrease in the accuracy of in-context learning. This suggests the model struggles to interpret tasks from videos, a foreseeable outcome since the training primarily incorporates multimodal prompts rather than video content. Such findings underscore the critical role of task tuning in enhancing the model's ability to generalize. The omission of the future state prediction module severely impacts performance, particularly in task T10, underscoring the importance of the model's predictive capabilities for effective action planning. Additionally, removing multi-turn tuning also reduces effectiveness, though less so than the absence of future state prediction, which likely serves as a regularization mechanism.

V. CONCLUSION, FUTURE WORK AND LIMITATION

Our method, ACTLLM, demonstrates superior performance on a benchmark dataset by incorporating an action consistency loss, notably improving compositional and zero-shot generalization. Our research advances the development of intelligent robotic systems that intuitively understand and execute human language commands, improving human-robot interaction. Future research could explore real-world applications and in-context learning capabilities, broadening the deployment of intelligent robotic systems.

While our model shows promising potential, a key challenge remains: 2D images often fail to capture the precise 3D locations and desired states essential for effective robot control.

REFERENCES

[1] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, 2022.

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

[3] J. Li, Q. Gao, M. Johnston, X. Gao, X. He, S. Shakiah, H. Shi, R. Ghanadan, and W. Y. Wang, "Mastering robot manipulation with multimodal prompts through pretraining and multi-task fine-tuning," *arXiv preprint arXiv:2310.09676*, 2023.

[4] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327–7334, 2022.

[5] 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.

[6] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[8] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011, pp. 1507–1514.

[9] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, "Instruct2act: Mapping multi-modality instructions to robotic actions with large language model," *arXiv preprint arXiv:2305.11176*, 2023.

[10] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[11] Y. Ge, A. Macaluso, L. E. Li, P. Luo, and X. Wang, "Policy adaptation from foundation model feedback," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 19 059–19 069.

[12] R. Wang, J. Mao, J. Hsu, H. Zhao, J. Wu, and Y. Gao, "Programmatically grounded, compositionally generalizable robotic manipulation," *arXiv preprint arXiv:2304.13826*, 2023.

[13] J. Zhang, K. Pertsch, J. Zhang, and J. J. Lim, "Sprint: Scalable policy pre-training via language instruction relabeling," *arXiv preprint arXiv:2306.11886*, 2023.

[14] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023.

[15] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," in *The 11th International Conference on Learning Representations*, 2023.

[16] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, *et al.*, "Language to rewards for robotic skill synthesis," *arXiv preprint arXiv:2306.08647*, 2023.

[17] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," *The International Journal of Robotics Research*, vol. 41, no. 1, pp. 45–67, 2022.

[18] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[19] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3665–3671.

[20] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation," in *Conference on robot learning*. PMLR, 2020, pp. 1369–1378.

[21] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[22] K. Kang, C. Xie, C. He, M. Yi, M. Gu, Z. Chen, K. Zhou, and H. Wu, "Learning efficient illumination multiplexing for joint capture of reflectance and shape." *ACM Trans. Graph.*, vol. 38, no. 6, pp. 165–1, 2019.

[23] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 739–13 748.

[24] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[25] E. Stengel-Eskin, A. Hundt, Z. He, A. Murali, N. Gopalan, M. Gombolay, and G. Hager, "Guiding multi-step rearrangement tasks with natural language instructions," in *Conference on Robot Learning*. PMLR, 2022, pp. 1486–1501.

[26] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning*. PMLR, 2023, pp. 287–318.

[27] 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.

[28] M. Shridhar and D. Hsu, "Interactive visual grounding of referring expressions for human-robot interaction," *arXiv preprint arXiv:1806.03831*, 2018.

[29] 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.

[30] 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.

[31] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[32] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," *arXiv preprint arXiv:2209.07753*, 2022.

[33] 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.

[34] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[35] 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.

[36] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn, *et al.*, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *Conference on Robot Learning*. PMLR, 2022, pp. 1303–1315.

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

[38] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," *arXiv preprint arXiv:2303.00001*, 2023.

[39] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," 2023.

[40] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," 2019.

[41] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, "Grounded language-image pre-training," in

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2022. [Online]. Available: http://dx.doi.org/10.1109/CVPR52688.2022.01069

[42] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[43] G. Wang, Y. Ge, X. Ding, M. Kankanhalli, and Y. Shan, "What makes for good visual tokenizers for large language models?" *arXiv preprint arXiv:2305.12223*, 2023.

[44] A. Dubey, A. Jauhri, and A. e. a. Pandey, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[45] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion, "Mdetr-modulated detection for end-to-end multi-modal understanding," in *ICCV*, 2021, pp. 1780–1790.

[46] A. Pashevich, R. Strudel, I. Kalevatykh, I. Laptev, and C. Schmid, "Learning to augment synthetic images for sim2real policy transfer," in *IROS)*. IEEE, 2019, pp. 2651–2657.

[47] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A generalist agent," 2022.

[48] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a visual language model for few-shot learning," 2022.