2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

LACY: A Vision-Language Model-based Language-Action Cycle for Self-Improving Robotic Manipulation

Youngjin Hong*, Houjian Yu*, Mingen Li, and Changhyun Choi

***Abstract*— Learning generalizable policies for robotic manipulation increasingly relies on large-scale models that excel at mapping language instructions to actions (L2A). However, this unidirectional training paradigm often produces policies that can execute tasks without deeper contextual understanding, thereby limiting their ability to generalize and to explain their behavior. We argue that the complementary skill of mapping actions back to language (A2L) is essential for developing more holistic and robust grounding. An agent capable of both acting and explaining its actions can form richer internal representations and, critically, unlock new paradigms for self-supervised learning. In this paper, we introduce LACY (Language-Action CYcle), a unified framework that learns such bidirectional mappings within a single vision-language model. LACY is jointly trained on three synergistic tasks: generating parameterized actions from language (L2A), explaining observed actions in language (A2L), and verifying semantic consistency between two language descriptions (L2C). The framework enables a self-improving cycle that autonomously generates new training data by chaining the L2A and A2L modules in an L2A2L pipeline. The L2C module then filters this data using an active data augmentation strategy that selectively targets low-confidence cases, thereby improving the model efficiently without requiring additional human annotations. Extensive experiments on pick-and-place tasks in both simulation and the real world demonstrate that LACY substantially improves task success rates by 50.56 percentage points on average compared to baseline methods and yields more robust language-action grounding for robotic manipulation. For more details, please refer to our project page: https://vla2026.github.io/LACY/**

I. Introduction

Human actions and language are deeply and bidirectionally intertwined [1]–[3]. Humans can rapidly acquire new tasks by observing demonstrations, semantically interpreting the underlying task procedures, and describing the observed actions in natural language. For example, after watching the toy block manipulation shown in Fig. 1, one can readily articulate the task as a sequence of sentences. Conversely, when given a textual task description, humans can typically (1) interpret the intended actions (e.g., pick, place, grasp, put), (2) ground the mentioned objects in the environment (e.g., yellow block, blue cylinder), and (3) execute the described action sequence accordingly.

Learning from Demonstration (LfD) and imitation learning have been extensively studied for robotic manipulation tasks [4]–[8]. While these approaches have demonstrated strong performance, they typically require large amounts of high-quality demonstration data, which are costly and labor-intensive to collect [9]. Recently, the emergence of Vision-Language Models (VLMs) has driven substantial progress by leveraging their generalization capabilities and open-world reasoning. However, existing approaches, from early framework such as CLIPort [10] to large-scale generalist models like RT-2 [11] and OpenVLA [12], have predominantly focused on a single paradigm: learning a unidirectional mapping from *language-to-action* (L2A). Although these L2A approaches have achieved notable success, their effectiveness remains fundamentally limited by their dependence on massive, passively collected datasets, which constrains both scalability and data efficiency.

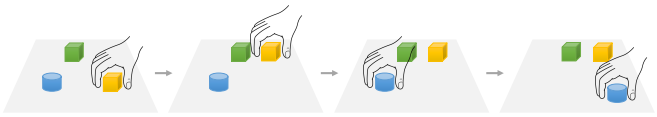

Fig. 1: **Human demonstration of toy object manipulation.** Humans can readily infer task procedures from a manipulation demonstration and express them in language (e.g., "pick up the yellow block" → "place it to the right of the green block" → "grasp the blue cylinder" → "put it on the bottom right of the table"). This linguistic description enables humans to accurately replicate the demonstrated action sequence.

To address this fundamental limitation, we advocate for an approach grounded in a more holistic, bidirectional integration of language and action. Inspired by findings in neuroscience [2], [3], we posit that the complementary ability to describe tasks by observing manipulation actions, referred to as *action-to-language* (A2L), is equally essential yet remains relatively underexplored in robot learning. An agent equipped with both L2A and A2L capabilities can develop richer and more robust internal representations of the world. More critically, this synergy enables a powerful self-supervised learning paradigm: the model can execute a robot action from a language command (L2A), then generate a new language description of that action based on its own perception (A2L), and finally assess whether the initial and generated descriptions are semantically consistent (L2C). This cycle allows the model to generate, evaluate, and refine its own data, thereby substantially reducing its reliance on external human supervision.

In this paper, we introduce **LACY** (Language-Action CYcle), a unified VLM framework that leverages the language-action cycle for robust and scalable learning. LACY is implemented by fine-tuning a single LLaVA-NeXT model [13] to

*Equal contribution. Author order is alphabetical. This work was supported in part by the Sony Research Award Program and NSF Award 2143730. The authors are with the Department of Electrical and Computer Engineering, Univ. of Minnesota, Minneapolis, USA {hong0745, yu000487, li002852, cchoi}@umn.edu

perform three complementary yet synergistic roles: (1) generating actions from language (L2A), (2) explaining actions in language (A2L), and (3) verifying semantic consistency between two language descriptions (L2C). We first train the L2A and A2L components on an initial dataset and then use the proposed language–action cycle to autonomously generate a large-scale, high-quality dataset for further training. A central innovation of our framework is a confidence-based active data augmentation strategy, in which new data is stochastically generated only for low-confidence scenarios, thereby directing learning toward more challenging cases. We evaluate LACY on a tabletop pick-and-place task in both simulation and real-world settings. Experimental results show that policies trained through LACY's self-improvement loop significantly outperform baselines trained solely on the initial dataset, yielding substantial gains in task success rates.

The main contributions of this work are as follows:

- **A unified VLM framework (LACY)** that is jointly trained to perform three complementary tasks: language-to-action generation (L2A), action-to-language explanation (A2L), and semantic consistency verification (L2C).
- **A self-improving data generation pipeline** that leverages the language-action cycle to autonomously produce new training data, which is subsequently filtered by the L2C module to ensure high-quality data.
- **A confidence-based active data augmentation strategy** that directs data generation toward low-confidence scenarios, thereby mitigating overfitting and improving model performance on challenging cases.

II. Related Work

A. Vision-Language Models for Robotic Manipulation

One of the challenges for generalist robot policies lies in grounding vision, language, and action, which exhibit significant abstraction gaps. Outside of robotics, there have been many successful attempts to bridge the vision and language feature spaces [14], [15]. Along with the success of vision-based imitation learning [4], [5], early attempts were made in language-grounded imitation learning [16], [17].

With the advent of state-of-the-art VLM models [18], [19], recent works tend to directly leverage the generalizability and reasoning capabilities of VLMs for robotic tasks. One dominant paradigm is end-to-end Vision-Language-Action (VLA) models [10], [12]. Given language instructions and image observations, VLA models directly generate robot actions. Thanks to large-scale robot datasets [20], [21], VLA models are built upon existing open-source VLM models by fine-tuning with robot datasets. Recent works also explore diverse approaches for faster and high-performance fine-tuning and inference of VLA models [22], [23]. However, due to the lack of robot data compared to open-world data from which VLMs were trained, these models must be trained on additional in-domain robot data, which are expensive and require significant time, equipment, and expertise to collect.

Recent approaches explore hierarchical VLA architectures that use VLMs to generate intermediate action representations and employ existing low-level policies [24], [25]. By using abstract robot action expressions, hierarchical models can fully leverage the reasoning capabilities of VLM models, either by fine-tuning with off-domain data [26], or even through zero-shot inference via off-the-shelf VLMs [27]. Our work has connections with these prior methods, as our L2A task generates a parameterized pick-and-place motion.

Prior works have mainly focused on how to collect or utilize large-scale data for robotic tasks. However, these models are almost exclusively trained for the unidirectional L2A task, while the complementary skill of observing an action and generating a linguistic description, A2L, remains largely unexplored. We argue that learning the A2L approach deepens understanding of robot tasks, and this can maximize learning efficiency from limited in-domain demonstrations through bidirectional grounding.

B. Data Generation and Self-Supervision in Robotics

The immense data requirements of VLA models have spurred research into data generation and self-supervision. Common strategies include using simulation to create large-scale datasets [21], [28], though this often introduces sim-to-real gaps. Reinforcement learning approaches have also been explored to enable autonomous data collection and policy improvement [29], [30], but these methods typically require extensive environment interaction and careful reward design. Alternative approaches focus on enabling models to learn from their own experience through self-supervision [31], [32], data augmentation technique [33], and cross-modal transfer [34]. However, a key challenge in self-supervised learning is ensuring the quality of self-generated data, as models may reinforce their own biases without external verification [35], [36].

LACY provides a novel form of self-supervision by framing the problem through the lens of cycle consistency, a powerful principle from generative modeling [37]. The core idea is that a forward mapping followed by a backward mapping should reconstruct the original input. This has been applied in robotics for tasks like cross-domain policy transfer [38], but its application to the language-action domain for data generation is new. Our language-action cycle ($\mathbf{l} \rightarrow \mathbf{a} \rightarrow \mathbf{l}'$) provides a principled self-supervisory signal as the semantic intent of the initial command $\mathbf{l}$ must be preserved in the reconstructed description $\mathbf{l}'$. The L2C module acts as the crucial verifier, providing the reliable, extrinsic-like feedback that naive self-correction lacks. By combining this with a confidence-based active data augmentation strategy, inspired by methods for calibrating LLM uncertainty [39], LACY focuses its learning on ambiguous cases, making the self-improvement process both robust and efficient.

III. Proposed Approach

A. Preliminary: Problem Definition and Notations

Following the notations in [9], [10], we assume that a set of n demonstrations $\mathcal{D} = \{\zeta_1, \zeta_2, \cdots, \zeta_n\}$ are given. Each demonstration $\zeta_t = (\mathbf{o}_t, \mathbf{l}_t, \mathbf{a}_t)$ is a triplet containing an image observation $\mathbf{o}_t$, a task description in human language

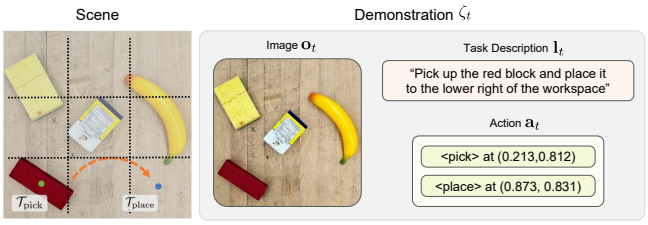

Fig. 2: **Notations.** Each demonstration ζ_t includes an image observation $\mathbf{o}_t$, a task description $\mathbf{l}_t$, and a pick-and-place action $\mathbf{a}_t$. The workspace is divided into a 3×3 grid. Coordinates (x, y) are normalized to $[0, 1]$, where $x, y \in [0, 1]$, with $(x, y) = (0, 0)$ at the left/top image border and $(x, y) = (1, 1)$ at the right/bottom border.

$\mathbf{l}_t$, and a pick-and-place action $\mathbf{a}_t = (\mathcal{T}_{\text{pick}}, \mathcal{T}_{\text{place}}) \in \mathbb{R}^2 \times \mathbb{R}^2$, as shown in Fig. 2.

From the demonstration set $\mathcal{D}$, a language-to-action (L2A) model learns a policy $\pi_{l \to a}$ that maps an observation $\mathbf{o}_t$ and a language instruction $\mathbf{l}_t$ to a predicted action $\hat{\mathbf{a}}_t$:

$$\pi_{l \to a} : \mathbf{o}_t, \mathbf{l}_t \to \hat{\mathbf{a}}_t. \tag{1}$$

Conversely, our proposed action-to-language (A2L) model learns a policy $\pi_{a \to l}$ that predicts a language description $\hat{\mathbf{l}}_t$ from an observation $\mathbf{o}_t$ and a demonstrated action $\mathbf{a}_t$:

$$\pi_{a \to l} : \mathbf{o}_t, \mathbf{a}_t \to \hat{\mathbf{l}}_t. \tag{2}$$

The language $\mathbf{l}_t$ of a pick-and-place action $\mathbf{a}_t$ includes which object to pick and where to place, for which absolute or relative spatial description is employed (see Section III-C for details).

Additionally, we introduce a language-to-consistency (L2C) model that estimates the consistency $c_t \in [0, 1]$ between two languages ($\mathbf{l}_t$ and $\hat{\mathbf{l}}_t$) conditioned on the observation $\mathbf{o}_t$:

$$\pi_{l \to c} : \mathbf{o}_t, \mathbf{l}_t, \hat{\mathbf{l}}_t \to c_t. \tag{3}$$

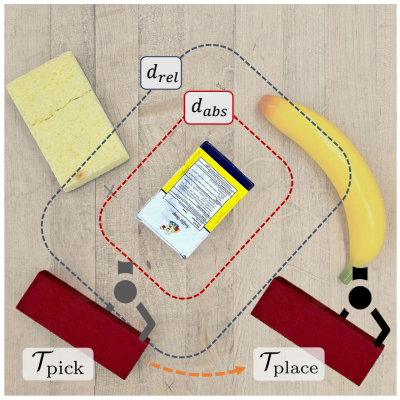

Fig. 3: **Spatial description types.** Task description for placing an object uses different forms of language descriptions (absolute or relative) based on the Euclidean distance to the placing location and the proximity to the outer contour of the nearest object.

B. System Overview

We introduce LACY (Language-Action CYcle), a framework built upon a single, powerful VLM (LLaVA-NeXT [13]) that is fine-tuned to serve three roles: an action generator (L2A), an action explainer (A2L), and a consistency verifier (L2C). Fig. 4 illustrates the overall architecture. The core of our framework is a closed-loop process where LACY uses its bidirectional capabilities to generate new, high-quality training data, which is then used to iteratively refine the model itself.

C. Explaining Robot Actions in Language via A2L

The main goal of the A2L model is to generate a language description by observing a manipulation action. Since we consider pick-and-place actions, the language needs to describe which object to grasp and where to place it, which necessitates spatial description. We designed A2L to generate two distinct types of naturalistic spatial descriptions: absolute and relative. Instead of using unconstrained free-form descriptions, we constructed a controlled set of linguistic templates to describe robot actions. This provides consistency across samples while maintaining sufficient linguistic diversity for generalization. From an image $\mathbf{o}_t$ and an action $\mathbf{a}_t$, we generate language based on the spatial context.

- ***Absolute Spatial Description***: Describes the placing motion relative to the entire workspace, which is divided into a 3×3 grid (e.g., "top left," "center"). This is used when the placement location is not near any other object. For example: "Pick the yellow block and place it in the middle left of the workspace."
- ***Relative Spatial Description***: Describes the placing motion relative to a nearby reference object. This is used when the placement is close to another object. For example: "Pick the yellow block and place it to the top right of the mustard bottle."

Humans naturally switch between these description styles. To mimic this, we generate descriptions based on the normalized distance between the placing location and the closest object, as shown in Fig. 3. If the distance is less than d_{rel}, we generate a relative description. If it is greater than d_{abs}, we generate an absolute one. In our experiments, we use $d_{abs} = 0.15$ and $d_{rel} = 0.3$ as we empirically found that these values tend to generate a balanced distribution between relative and absolute descriptions.

D. Unified Model via Two-Stage Fine-Tuning

A primary challenge in robot learning is the scarcity of robot-specific demonstration data compared to the abundance of general computer vision datasets. To address this, we propose a data-efficient, two-stage fine-tuning strategy that leverages a Chain-of-Thought (CoT) [40] reasoning process.

Stage 1: Object Grounding Pre-training. We first pre-train our VLM backbone on an object grounding task. Using a dataset of 8,000 images with object labels and object center locations, we teach the model to perform a foundational vision task: identifying all objects in an image and listing their names and center coordinates. This task involves outputting a set of detected objects $\mathcal{O} = \{(n_i, p_i)\}_{i=1}^{N}$, where n_i is the object name and p_i is the center coordinate for the detected object i. This pre-training endows the model with a

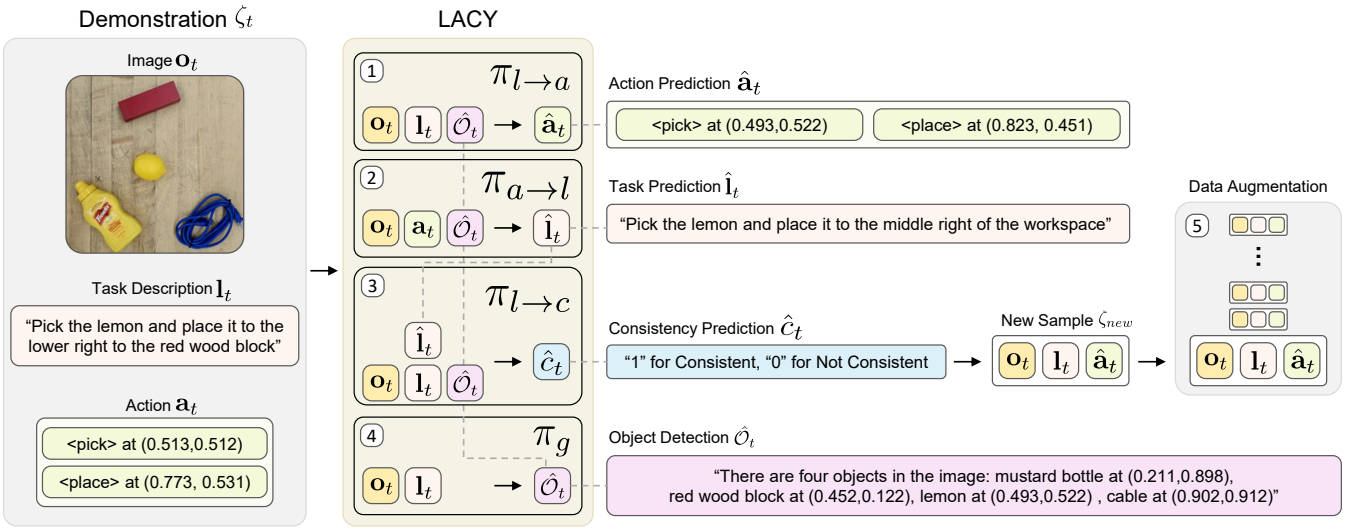

Fig. 4: **Overview of the LACY framework.** LACY (Language-Action CYcle) builds upon a single VLM [13] fine-tuned to serve three roles: (1) an action generator (L2A), (2) an action explainer (A2L), and (3) a consistency verifier (L2C). The framework operates as a closed-loop system, where these bidirectional capabilities enable LACY to generate new high-quality training data and iteratively refine itself. (4) Each task is framed as a chain-of-thought (CoT) process, where the model first performs object grounding to predict object names and locations ($\hat{\mathcal{O}}$) and then uses this contextual information to complete the target task. (5) As shown in Algorithm 1, new samples are generated and merged with the initial dataset.

robust visual understanding that serves as a strong prior for downstream manipulation tasks.

Stage 2: CoT-based Multi-task Fine-tuning. We then fine-tune the model on our smaller, robot-specific dataset of 1,000 demonstrations for the three synergistic tasks of LACY. Fig. 4 shows the input and output format for each model: L2A, A2L, and L2C. Crucially, all three tasks are formulated as a CoT process that explicitly utilizes the pre-trained grounding skill. The model is prompted to first perform the object grounding task to predict object name and location, $\hat{\mathcal{O}}$, and then use that as context to perform L2A, A2L, and L2C. This CoT approach not only makes the model's reasoning process more transparent and robust but also allows the grounding skill to be further refined with in-domain robotics data. This two-stage approach allows LACY to achieve high performance with significantly less robot-specific data. Throughout this process, we use Low-Rank Adaptation (LoRA) [41] for efficient fine-tuning.

E. Self-Improving Data Generation with L2A and A2L

The central mechanism of LACY is its ability to generate its own training data through a closed loop, which we term the L2A2L pipeline. This process begins with an input language command $\mathbf{l}$ from our training set, which is fed into L2A to generate an intermediate action $\hat{\mathbf{a}}$. This generated action $\hat{\mathbf{a}}$ is then immediately fed into A2L, along with the same observation $\mathbf{o}$, to produce a new, reconstructed language description $\hat{\mathbf{l}}$. This cycle yields a new complete triplet, $\zeta_{new} = (\mathbf{o}, \mathbf{l}, \hat{\mathbf{a}})$, which can be added to our training set if its quality and semantic consistency are verified by the L2C module.

F. Confidence-based Data Augmentation with L2C

A naive approach to data augmentation would be to generate many samples and filter them. However, this can lead to overfitting on tasks the model already performs well on, and could add redundant data if multiple outputs from a single sample are deemed valid. To address this, we propose a confidence-based active data augmentation strategy guided by L2C that incorporates a robust, voting-based filtering mechanism, as detailed in Algorithm 1.

The process begins with a single, deterministic pass through the L2A2L pipeline. For a given input language instruction $\mathbf{l}_t$, we obtain a generated action $\hat{\mathbf{a}}_t$ and a reconstructed language description $\hat{\mathbf{l}}_t$. Instead of having the VLM directly output a numerical value in text form, which can be unreliable [42], we task L2C with a binary classification problem. It outputs a consistency value "1" if $\mathbf{l}_t$ and $\hat{\mathbf{l}}_t$ are consistent, and "0" otherwise. To obtain a confidence value for binary classification, we directly extract the logits corresponding to the tokens "0" and "1" from the model output at the final decoding step, as shown in Fig. 5. Instead of selecting the discrete token, we apply a sigmoid function to the logit difference to obtain a continuous probability in the range of $[0, 1]$.

This technique of using logit differences to calibrate model confidence is inspired by similar methods in language and vision models [39], [43], [44]. The continuous consistency score c is calculated as a probability:

$$c = \sigma(z_1 - z_0) \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function. Note that this formulation is equivalent to the two-class softmax:

$$\sigma(z_1 - z_0) = \frac{\exp(z_1)}{\exp(z_0) + \exp(z_1)}. \tag{5}$$

- If the consistency score c is high (i.e., $c \geq \tau$), we consider this a high-confidence case that the model has already mastered. No additional data are generated for this sample, avoiding redundant computation.

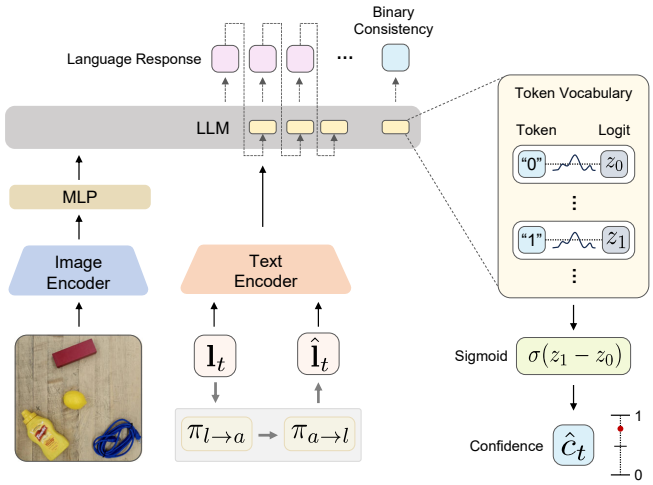

Fig. 5: **Binary confidence extraction from VLM outputs.** The logits z_0 and z_1 corresponding to the tokens "0" and "1" are used to compute a confidence score c.

For each candidate action $\mathbf{a}'_i \in \mathcal{A}_{\text{cand}}$, we then generate a corresponding group of N language descriptions ($\mathcal{L}_{\text{cand}}$) using A2L. Instead of accepting an action upon the first successful validation, we evaluate the entire group. An action $\hat{\mathbf{a}}_i$ is deemed valid and added to our new dataset only if a majority ($\geq \nu$) of its corresponding language descriptions are judged as consistent with the original instruction $\mathbf{l}_t$ by L2C. This majority-voting approach prevents the addition of duplicate $(\mathbf{l}, \hat{\mathbf{a}})$ pairs and ensures that only actions that are robustly and consistently explainable are used for retraining, filtering out potential flukes.

Finally, to prevent catastrophic forgetting, we re-initialize the model from the base model on a merged dataset including both original data and newly generated, high-quality data.

IV. Experiments

A. Experimental Setup

Our experiments are conducted in a simulated tabletop environment built with CoppeliaSim [45] and 32 YCB objects, and in a real-world setup with a Franka Emika Panda Robot and an Intel RealSense D415 camera using 12 real-world objects, as shown in Fig. 6.

Dataset: We use a training dataset comprising up to 4,000 successful pick-and-place demonstrations. For ablation studies, we employ a smaller subset of 1,000 demonstrations to evaluate the effectiveness of our proposed methods under data-scarce conditions, while the simulation test set includes 100 unseen scenarios. For the real-world environment, we trained the model with 212 demonstrations and tested with 50 unseen scenarios.

Implementation Details: The object grounding pre-training is conducted for 2 epochs. The second stage of the multi-task fine-tuning is performed for 5 epochs, as well as the multi-task re-fine-tuning using the augmented dataset including both original data and data sampled via the L2C module. All experiments are carried out on a single NVIDIA A40 GPU with 48 GB of memory.

Algorithm 1: Confidence-based Active Data Augmentation with Majority Voting

Input: Initial demonstration set $\mathcal{D}$, LACY model π, number of stochastic samples N, consistency threshold τ, voting threshold $\nu = 0.5$
Output: Augmented dataset $\mathcal{D}_{\text{aug}}$

```
D_new ← ∅;
for each (o_t, l_t, a_t) ∈ D do
    â_t ← L2A(o_t, l_t);
    l̂_t ← A2L(o_t, â_t);
    c ← L2C(o_t, l_t, l̂_t);
    if c ≤ τ then
        A_cand ← StochasticL2A(o_t, l_t, N);
        for each â_i ∈ A_cand do
            N_p ← 0;
            L_cand ← StochasticA2L(o_t, â_i, N);
            for each l̂_j ∈ L_cand do
                c ← L2C(o_t, l_t, l̂_j);
                if c ≥ τ then
                    N_p ← N_p + 1;
            if N_p/N ≥ ν then
                D_new ← D_new ∪ {o_t, l_t, â_i}
    D_aug ← D ∪ D_new;
```

TABLE I: Reasoning Capability Results in Simulation

Model	L2A (%)	A2L (%)	L2C (%)
GPT-4o w/ Grounding	<u>90</u>	39	8
GPT-4o w/o Grounding	28	<u>40</u>	<u>76</u>
LLaVA-NeXT (base)	6	6	50
LACY (4k data, ours)	**95**	**76**	**95**

Evaluation Metrics: We evaluate our framework using the following metrics:

- **L2A (%):** The task success rate of the L2A module. A task is considered successful if the correct object is grasped and placed at the target location which semantically meets the given spatial description.
- **A2L (%):** The task success rate of the A2L module. A task is considered successful if it correctly mentions the target object to grasp and accurately describes the spatial relationship for placement given the action $\mathbf{a}_t$.
- **L2C (%):** The accuracy of the L2C module in detecting semantically consistent language-action pairs.
- **Pick / Pick & Place (%):** In real-world experiments, the percentage of successful grasping and placement completed by the robot following model-generated actions.

B. Reasoning Capabilities in Simulation

First, we evaluate the core reasoning capabilities of our bidirectional model. To establish a strong performance baseline, we use a version of our LACY model fine-tuned on a larger dataset of 4,000 demonstrations. We compare it

Fig. 6: **Real robot experiment setup.** (Left) The workspace is divided into a 3×3 grid to provide an absolute spatial reference for task descriptions. A top-view image captured by an Intel RealSense D415 camera serves as the visual input to LACY. (Right) Objects used in the real-robot experiment, including both everyday items and selected YCB objects.

against several baselines to justify the need for fine-tuning a specialized model. The results are presented in Table I.

Baselines: We compare our *LACY-Joint (4k data)* model with *LLaVA-NeXT (base)*, which has not been fine-tuned on robotics data, and *GPT-4o*, both with and without ground-truth object location information provided in the prompt.

Analysis: The results highlight that large, general-purpose models like GPT-4o and the base LLaVA-NeXT model struggle with precise spatial grounding from a single image, leading to poor performance. However, when provided with explicit grounding information, GPT-4o achieves near-perfect results, demonstrating its powerful reasoning capability when ambiguity in object grounding is removed. Our fine-tuned LACY model significantly outperforms the non-fine-tuned baselines, showing the effectiveness of our training approach.

C. Ablation Studies

We conduct a series of ablation studies to validate each component of our proposed framework. As our goal is to improve performance in data-limited scenarios, all models in these studies are trained on a reduced dataset of 1,000 demonstrations.

TABLE II: Ablation Study on Joint Training and Filtering

Model	L2A (%)	A2L (%)	L2C (%)
LACY-Ind	78	78	92
LACY-Joint (ours)	83	80	92
LACY-Joint-SI (ours)	**93**	**85**	**95**

Effectiveness of Joint Training and Filtering: We first investigate the benefits of our joint training approach and the self-improvement filtering mechanism. As shown in Table II, the jointly trained model (*LACY-Joint*) outperforms the independently trained one (*LACY-Ind*), supporting our hypothesis that a shared representation through joint training is beneficial. Furthermore, adding our self-improvement cycle (*LACY-Joint-SI*) provides a substantial performance boost across all metrics, confirming the effectiveness of our self-improvement pipeline.

Necessity of CoT: We examine the impact of our Chain-of-Thought (CoT) prompting strategy. Table III compares our CoT-based model with a version that directly generates the output without the intermediate object grounding step (*LACY-non-CoT*). The results indicate that the explicit reasoning step provided by CoT is crucial for achieving higher performance.

TABLE III: Ablation Study on Chain-of-Thought Prompting

Model	L2A (%)	A2L (%)	L2C (%)
LACY-non-CoT	52	43	84
LACY-CoT (ours)	**83**	**80**	**92**

D. Self-Improvement Capability

To evaluate the model's ability to improve through self-generated data, we assess its performance after varying numbers of self-improvement iterations (from one to three), starting from the model trained on only 100 demonstration data points. In each iteration, the model performs data augmentation via the L2C task and fine-tuning with the combined dataset (Algorithm 1), generating 100 new triplets $(\mathbf{o}, \mathbf{l}, \hat{\mathbf{a}})$. Thus, a model with n self-improvement iterations is trained on $100 \times (n+1)$ data points. Fig. 7 reports the task success rates for models trained on the corresponding dataset sizes. *LACY-Joint* is trained solely on the ground-truth dataset, while *LACY-Joint-SI* is trained on a mixture of the ground-truth data and the additional data generated through the L2C-based data augmentation process. While *LACY-Joint-SI* initially underperforms *LACY-Joint* on L2A and A2L, its success rates increase consistently as more self-generated data is incorporated, and it notably surpasses *LACY-Joint* on L2C.

E. Real-World Evaluation

Finally, we evaluate LACY on real data and compare the model performances on a 7-DoF Franka Emika Panda robot. Refer to the real robot experiment setup in Fig. 8. We evaluate the model's ability to generate correct actions, language, and consistency decisions based on real-world images, and separately report the physical grasp success rate. To investigate the sim-to-real gap, we compare the performances between the models by ablating training with real-world data. *LACY-Ind-Real* and *LACY-Joint-Real* are further trained on a real-world dataset from the simulation-trained models *LACY-Ind* and *LACY-Joint*. The results in Table IV show that the performance gains observed in simulation translate effectively to the real world. Notably, both *LACY-Ind* and *LACY-Joint* successfully detect most of the objects in the given image, and even accurately locate the objects that are both in the simulation and real-world (e.g., mustard bottle, sponge). Most of the failure cases are due to misnaming the objects in the image. Through training with real-world data, *LACY-Ind-Real* and *LACY-Joint-Real* improve the performance by successfully deriving correct object names, with *LACY-Joint-Real* accomplishing the best performance across all three tasks. Table V shows that the pick-and-place actions generated by the LACY models are also effective in real-robot tasks.

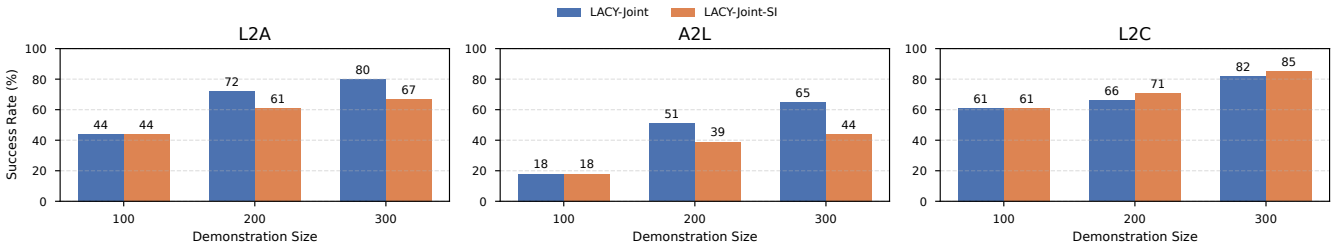

Fig. 7: **Self-improvement of LACY.** LACY-Joint is trained only on ground-truth data, while LACY-Joint-SI is trained on ground-truth plus L2C-sampled data.

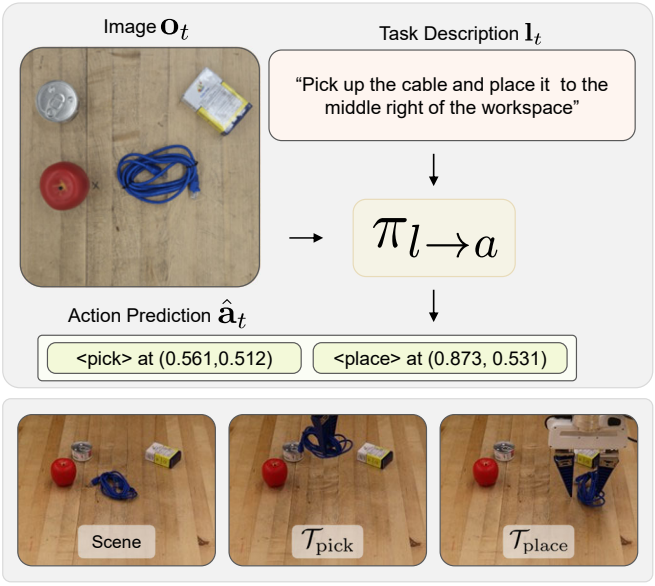

Fig. 8: **Real Robot Reasoning.** (Top) Given an image observation $\mathbf{o}_t$ and a task description $\mathbf{l}_t$, the robot reasons the appropriate pick-and-place action $\hat{\mathbf{a}}_t$ via L2A. (Bottom) The robot grasps the cable and places it in the designated location.

TABLE IV: Real-World Data Evaluation

Model	L2A (%)	A2L (%)	L2C (%)
LACY-Ind	78	36	94
LACY-Ind-Real	82	80	94
LACY-Joint	80	28	**98**
LACY-Joint-Real	**88**	**88**	**98**

V. Conclusion

This paper presented LACY, a VLM-based framework that achieves bidirectional grounding between language and action for robotic manipulation. By unifying action generation (L2A), action explanation (A2L), and semantic verification (L2C) within a single model, LACY enables a novel self-improving cycle that autonomously generates and filters its own training data. Experimental results demonstrate that the synergy between joint bidirectional training and confidence-based active data augmentation significantly improves manipulation success rates in both simulation and real-world settings. These findings highlight that moving beyond the prevailing unidirectional L2A paradigm, and equipping agents with the ability to both act and explain

TABLE V: Real-World Robot Evaluation

Model	Pick (%)	Pick & Place (%)
LACY-Ind-Real	65	55
LACY-Joint-Real	**72.5**	**60**

offers a promising path toward more robust, data-efficient, and generalizable robotic systems.

Despite these encouraging results, several limitations remain and warrant further investigation. Notably, the L2C semantic verification module is not specifically trained to evaluate object grounding quality, making the data generation pipeline susceptible to errors from incorrect object grounding outputs. Because object grounding results play a critical role in the model's behavior through CoT-based fine-tuning, such errors may propagate downstream, leading the model to misidentify picking locations and ultimately reducing task success rates. Future work will address these limitations while expanding the framework's capabilities. In particular, we plan to (1) develop more robust perception and verification modules that can better handle detection uncertainties and (2) extend the LACY framework to more complex, long-horizon tasks and a broader range of manipulation skills. We believe this bidirectional perspective can serve as a foundation for scalable robot intelligence.

References

[1] A. M. Glenberg and M. P. Kaschak, "Grounding language in action," *Psychonomic Bulletin & Review*, vol. 9, no. 3, pp. 558–565, Sept. 2002. [Online]. Available: https://doi.org/10.3758/BF03196313

[2] G. Rizzolatti and M. A. Arbib, "Language within our grasp," *Trends in Neurosciences*, vol. 21, no. 5, pp. 188–194, May 1998.

[3] F. Pulvermüller, "Brain mechanisms linking language and action," *Nature Reviews Neuroscience*, vol. 6, no. 7, pp. 576–582, July 2005, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nrn1706

[4] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[5] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[7] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[8] 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.

[9] 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. 103–120.

[10] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," 2021. [Online]. Available: https://arxiv.org/abs/2109.12098

[11] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023. [Online]. Available: https://arxiv.org/abs/2307.15818

[12] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[13] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," January 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-01-30-llava-next/

[14] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[15] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.

[16] Y. Yang, X. Lou, and C. Choi, "Interactive robotic grasping with attribute-guided disambiguation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4439–4446, 2022.

[17] H. Yu, M. Li, A. Rezazadeh, Y. Yang, and C. Choi, "A parameter-efficient tuning framework for language-guided object grounding and robot grasping," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 14 353–14 360.

[18] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[19] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and more," *arXiv preprint arXiv:2308.12966*, 2023.

[20] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[21] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[22] A. Pari, K. Black, C. Xu, H. Walke, S. Dasari, A. Kumar, A. Rajeswaran, C. Finn, and S. Levine, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2405.08232*, 2024.

[23] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "Fast: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

[24] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox, "Robopoint: A vision-language model for spatial affordance prediction for robotics," *arXiv preprint arXiv:2406.10721*, 2024.

[25] S. Nasiriany, S. Kirmani, T. Ding, L. Smith, Y. Zhu, D. Driess, D. Sadigh, and T. Xiao, "Rt-affordance: Affordances are versatile intermediate representations for robot manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2411.02704

[26] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, R. Yu, C. R. Garrett, F. Ramos, D. Fox, A. Li, A. Gupta, and A. Goyal, "Hamster: Hierarchical action models for open-world robot manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2502.05485

[27] Y. Qian, X. Zhu, O. Biza, S. Jiang, L. Zhao, H. Huang, Y. Qi, and R. Platt, "Thinkgrasp: A vision-language system for strategic part grasping in clutter," *arXiv preprint arXiv:2407.11298*, 2024.

[28] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," in *Conference on Robot Learning*. PMLR, 2023, pp. 1820–1864.

[29] A. Kumar, Z. Fu, D. Pathak, *et al.*, "Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.

[30] D. Kalashnikov, A. Irpan, P. Pastor, *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*, 2018.

[31] Y. Burda, H. Edwards, A. Storkey, *et al.*, "Exploration by random network distillation," in *International Conference on Learning Representations*, 2018.

[32] A. Srinivas, M. Laskin, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*, 2020.

[33] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International Conference on Learning Representations*, 2020.

[34] P. Sermanet, C. Lynch, Y. Chebotar, *et al.*, "Time-contrastive networks: Self-supervised learning from video," in *International Conference on Robotics and Automation*, 2018.

[35] J. Huang, X. Gu, L. Hou, J. Wang, J. Li, G. Chen, C. Chen, Z. Liu, Y. Zhang, T. Gui, *et al.*, "Large language models cannot self-correct reasoning yet," *arXiv preprint arXiv:2310.01798*, 2023.

[36] K. Valmeekam, M. Marquez, S. Kumar, M. Sridharan, and S. Kambhampati, "Can llms really grasp simple causal structures?" *arXiv preprint arXiv:2305.15769*, 2023.

[37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[38] A. Ajay, Y. Saber, B. Roh, and T. Jaakkola, "Cycle-consistent inverse dynamics for visual imitation learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 4359–4366.

[39] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, A. Jones, N. Joseph, N. DasSarma, *et al.*, "Language models (mostly) know what they know," *arXiv preprint arXiv:2207.05221*, 2022.

[40] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[41] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.

[42] P. Manggala, A. Mastakouri, E. Kirschbaum, S. P. Kasiviswanathan, and A. Ramdas, "Qa-calibration of language model confidence scores," *arXiv preprint arXiv:2410.06615*, 2024.

[43] 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.

[44] D. Shah, M. Liang, Y. Liu, A. S. Naren, G. Stone, A. Kumar, S. Scherer, and A. Gupta, "LM-Nav: Robotic navigation with large language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 052–18 062.

[45] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com.