KitchenVLA: Iterative Vision-Language Corrections for Robotic Execution of Human Tasks

Kai Lu[1,2], Chenyang Ma[1,2], Chiori Hori[1*], Diego Romeres[1*]

***Abstract*— In this paper, we present KitchenVLA, a Vision-Language-Action (VLA) framework for generating and optimizing executable robot actions from human instructional videos. While recent advances in video understanding and step generation have shown promising results, translating these steps into robot-executable actions remains challenging, particularly for complex, long-horizon tasks such as those in kitchen environments. These challenges arise from domain discrepancies between human videos and robotic settings, as well as mismatches between human actions and robot capabilities. To address these issues, we propose a zero-shot action planning and correction framework, where a Vision-Language Model (VLM) acts as an evaluator to analyze both the original human video and the robot's observations to detect domain mismatches. The system assesses differences in object states and action feasibility, and generates corrective actions to align the robot's execution with the intended task. By incorporating keyframe selection, language-guided segmentation, and simulation-based verification, KitchenVLA iteratively refines robotic plans to ensure contextual accuracy and executability. Through domain-aware evaluation and correction, our framework enhances the adaptability and robustness of robotic task execution in kitchen environments, advancing the integration of VLMs into robot learning and executable plan correction.**

Keywords: Robot action generation, Interactive error correction, Human-robot collaboration, Multimodal scene understanding, Multimodal LLM

I. Introduction

Learning robot action sequences from human videos is a key problem in embodied intelligence, with broad applications such as household assistants and robotic chefs. Recent advances in embodied AI have shown that Vision-Language Models (VLMs) [1]–[3] and Multimodal Large Language Models (MLLMs) [4], [5] can understand instructional videos and generate textual task steps, making them promising candidates for the core reasoning modules of Vision-Language-Action (VLA) frameworks.

However, directly applying VLMs to robot action generation and execution remains highly challenging. One approach is to extend VLMs with control-level action decoders, but such monolithic solutions require large-scale annotated datasets and expensive computational resources [6]–[8]. Alternatively, hierarchical methods first generate high-level task steps by VLMs or MLLMs, and then translate them into robot-executable actions using a predefined skill library or pretrained policy models [9], [10]. While this approach reduces model complexity, it faces three critical challenges,

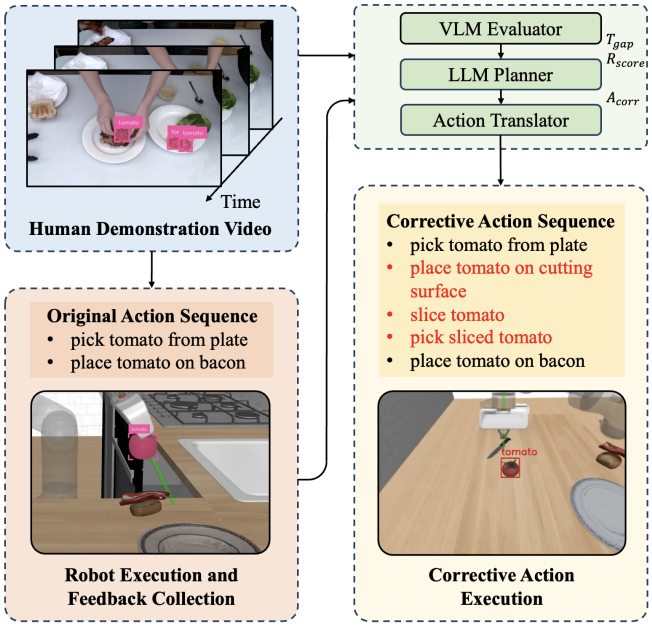

Fig. 1. **Bridging Human-Robot Instruction Gaps via VLM Corrections.** Given a human demonstration video, KitchenVLA first extracts a high-level action sequence, which may fail in the robot environment due to domain discrepancies (e.g., requiring sliced vs. whole tomato). The system collects visual feedback from robot execution and compares it with the human video using a VLM Evaluator. Detected mismatches are passed to an LLM Planner, which generates corrective substeps that are translated into executable actions. This process yields a refined action plan aligned with the robot's capabilities while preserving the original task intent.

which are environmental gap between human videos and the robot workspace, embodiment gap between human actions and robot capabilities, and the semantic loss due to the abstraction of human demonstrations into text, often omitting fine-grained contextual cues.

Existing frameworks typically adopt a one-pass execution strategy, extracting action steps and executing them through modules like LLM-based translators (e.g., code-as-policies [11]), imitation or reinforcement learning policies [12], or predefined action primitives (e.g., SayCan [9]). These methods focus on optimizing execution under the assumption of environmental similarity, and often break down in complex, long-horizon tasks with significant domain shifts, such as cooking. Such tasks demand robots to adapt their actions to the environment and correct execution failures. Previous research on action correction and failure recovery has explored LLM-based plan editing [13], multimodal error reasoning [14], and human-in-the-loop guidance [15]. However, most approaches are limited to single-step recovery or

[1]: Mitsubishi Electric Research Laboratories (MERL), Cambridge, USA.
[2]: Department of Computer Science, University of Oxford, Oxford, UK.
[*]: Corresponding Author.

robot-environment-based corrections and are not designed for domain-level, sequence-wide adaptation of robot behaviors across diverse environments.

To address these challenges, we propose KitchenVLA, a Vision-Language-Action framework for domain-aware evaluation and correction of robot action sequences derived from human instructional videos. At the core of our framework is a feedback loop and replanning module, where VLMs and LLMs act as agents to compare the object states and task progressions in the human video with the robot's observations after each action. When discrepancies are detected, e.g., a robot is instructed to pick a sliced tomato but only has access to a whole tomato, KitchenVLA generates and inserts corrective steps, such as slicing, to align execution with the intended goal. In addition to physical mismatches, our system addresses semantic and logical issues such as incorrect object references (e.g., confusing bacon and bread), ambiguous instructions (e.g., interpreting *add* as *place* vs. *rotate+pour*), and plan-level inconsistencies (e.g., improper *pick/place* ordering for single-arm robots).

KitchenVLA integrates four components: 1) Action Mapping: VLM-generated steps are heuristically grounded to robot-executable actions and objects using a semantic mapping procedure. 2) Robot Execution and Feedback: The robot executes the mapped sequence and logs visual observations and execution outcomes. 3) VLM-Based Evaluation and LLM Replanning: Robot observations are compared with video keyframes to identify domain gaps, and corrective steps are generated. 4) Iterative Plan Refinement: Corrections are compiled and translated into a new action sequence. This process is repeated for multiple iterations to improve plan feasibility.

By leveraging domain-aware multimodal feedback and adaptive plan correction, KitchenVLA enables robots to perform human-demonstrated tasks more reliably in mismatched or unseen environments.

In summary, our contributions are threefold:

- We propose a zero-shot automated error-correction framework that learns from human instructional videos to generate robot-executable plans.
- We enable effective action and skill adaptation for long-horizon complex tasks across different workspaces, addressing environment and embodiment mismatches.
- We introduce an iterative replanning loop based on robot execution feedback to continuously refine action sequences and improve execution robustness.

To this end, we present KitchenVLA to bridge vision-language planning with domain-level correction, enabling robust robotic execution in kitchen scenarios.

II. Related Work

A. Vision-Language Action (VLA) Models and Frameworks

Vision-Language Models (VLMs) [1], [2], [16]–[18] have demonstrated remarkable capabilities in bridging visual perception and natural language understanding, making them well-suited for guiding robotic actions. OKAMI [19] leverages VLMs for human motion retargeting, focusing on mimicking arm motion based on SMPL-H models. RoboMimic [20] presents a structured imitation learning framework, providing valuable datasets and policies for robotic skill acquisition. Hydra [21] integrates hierarchical imitation learning, which is beneficial for long-horizon robotic tasks. Additionally, models such as OpenVLA [8] enhance multimodal learning, improving robots' ability to interpret and act upon visual-language inputs.

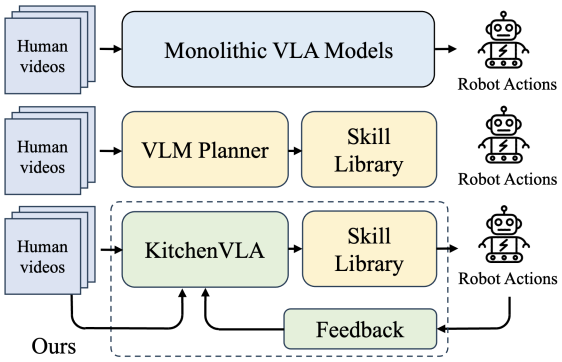

Fig. 2. **Comparison of different methods for vision-language-action planning.** Unlike monolithic VLA models and one-shot VLM planners, KitchenVLA introduces a feedback loop to iteratively refine robot action sequences based on execution outcomes.

Recent advancements such as DP-VLA [22] have demonstrated VLA techniques in structured environments like RoboCasa, enabling robots to execute predefined single-stage tasks by utilizing OpenVLA's latent space. Similarly, SayCan [9] employs LLMs as planners to bridge low-level behavior cloning (BC) with high-level reinforcement learning (RL), showcasing the potential of VLA in long-term mainpulation tasks.

B. Robotic Manipulation for Kitchen Tasks

Cooking robots require the integration of perception, planning, and action execution, often under significant constraints imposed by real-world kitchen environments. YORI [23] introduces a system with dual-arm robotic configurations for kitchen tasks, incorporating state machines to manage action sequences. SeeDo [24] follows a structured pipeline approach to video understanding and action mapping in kitchen environments, emphasizing task organization.

Alternatively, FOON [25] focuses on knowledge graph-based representations of cooking tasks, while Cook2LTL [26] explores logical task formulation using LTL. Research into modular robotic frameworks, such as MOSAIC [27], provides insights into hierarchical control systems for cooking automation. Notably, SliceIt [28] and LAVA [29] address specific kitchen manipulation tasks like cutting and scooping, demonstrating the necessity for skill-specialized robotic interventions in cooking environments.

C. Failure Correction and Replanning in Robotics

Failure recovery in robotics is a crucial component in ensuring robust and reliable performance in real-world scenarios. Traditional approaches often involve re-execution of

failed actions or predefined fallback strategies. However, recent research has emphasized the importance of integrating multimodal reasoning and feedback into the failure correction process.

Recently, methods such as CAPE [13] propose reprompting strategies for replanning upon execution failures, aligning LLM-based approaches with classical task and motion planning (TAMP) paradigms. Similarly, REFLECT [14] incorporates multimodal failure reasoning by integrating visual and audio sensory data to diagnose and correct errors in robotic tasks. These frameworks underscore the need for a multi-modal, context-aware failure recovery mechanism that can adapt to various operational constraints.

D. Video Understanding for Robotics

Video understanding [30]–[33] is a crucial component in enabling robots to learn from human demonstrations by extracting meaningful task sequences and structuring them into executable steps. Approaches such as VideoMME [34] and AVBLIP [5], [35] use video-to-text transformations to generate structured task representations for robotic execution. In robotic learning, the challenge lies in converting raw video demonstrations into a form that aligns with robotic capabilities and keeping the object states the same between the two domains. KitchenVLA builds upon these methodologies by incorporating a VLM Evaluator to assess discrepancies between human demonstrations and robotic execution, ensuring task feasibility and alignment within the robotic environment (see Fig. 2).

III. Problem Definition

A. Generating Robotic Action Sequence from Human Videos

Human instructional videos, such as those in the YouCook dataset [36], provide demonstrations of cooking tasks in kitchen environments. Our goal is to generate feasible robotic action steps to achieve the recipes while handling embodiment and workspace differences. In this work, we utilize the multimodal large language models (MLLM) to extract the action sequence, represented as:

$$A_0 = \text{MLLM}(Vi, Au, Sp), \quad A_0 = [a_0^0, a_0^1, ..., a_0^j], \quad (1)$$

where Vi, Au, and Sp represent the video, audio, and speech modalities, respectively. The output A_0 is a sequence of j action steps, where each a_0^i corresponds to an action primitive such as *pick*, *place*, *add*, *stir*, or *cut*. We denote the ground truth labels used to train the MLLM as A_0^*, which were obtained by human annotators using Amazon MTurk service as described in [5].

While the action sequences A_0 and A_0^* are accurate for human imitators, they are non-trivial to transfer to robots. This is because the annotators focused solely on the human videos without knowing the robotic embodiments and environments. Therefore, when following those action sequences, robots may fail to successfully reproduce the task demonstrated by the humans.

Two common limitations of the human-based action sequences are: 1) ambiguity in actions, where different operations (e.g., adding water vs. adding egg mixture) are both labeled as *add*; 2) mismatches in object conditions, such as distinguishing between whole and sliced bread or tomatoes. These gaps highlight the need for a method to correct and adapt action sequences for robotic execution.

The objective of this work is to generate an executable action sequence:

$$A_{\text{final}} = [a_n^0, a_n^1, \ldots, a_n^k], \quad (2)$$

that enables the robot to successfully replicate the task demonstrated by a human. This sequence is obtained through an iterative process consisting of n loops, each involving robot execution, action correction, and plan optimization. In each iteration, the current plan A_i is refined and updated to produce a new plan of variable length k.

IV. KitchenVLA

In this section, we present our framework that enables robots to bridge the domain gap between human instructional videos and robotic execution through multimodal evaluation and corrective planning. As illustrated in Fig. 3, the framework comprises three main modules: heuristic object and action mapping, robot execution with feedback collection, and iterative correction via a feedback loop involving VLM evaluation and LLM planning.

A. Heuristic Object and Action Mapping

Bridging the domain gap between human instructional videos and robotic execution lies in the discrepancies of objects and action capabilities. Objects that appear in human demonstrations often differ in form, availability, or granularity compared to the robotic environment. Likewise, human actions may not directly correspond to the robot's available skill library, therefore, we design a heuristic mapping procedure comprising object mapping and action mapping.

1) Object Mapping: Given the object set O_{human} extracted from the human video and the robot's known object set O_{robot}, we employ a LLM to perform semantic matching. For each object $o_h \in O_{\text{human}}$, we identify the most similar object $o_r \in O_{\text{robot}}$ based on textual similarity, producing a mapping with an associated similarity score:

$$\text{map}_O(o_h) = \arg \max_{o_r \in O_{\text{robot}}} \phi(o_h, o_r), \quad (3)$$

where $\phi(\cdot, \cdot)$ denotes the similarity function computed by the LLM.

2) Action Mapping: Human demonstrations often involve complex, unstructured actions that must be adapted into primitive robot skills. To achieve this, we first parse the textual descriptions of human action steps and perform frequency analysis across the dataset to identify the most common action verbs, as shown in Fig. 4.

Based on this analysis, we manually construct a mapping between high-frequency human actions and available robot

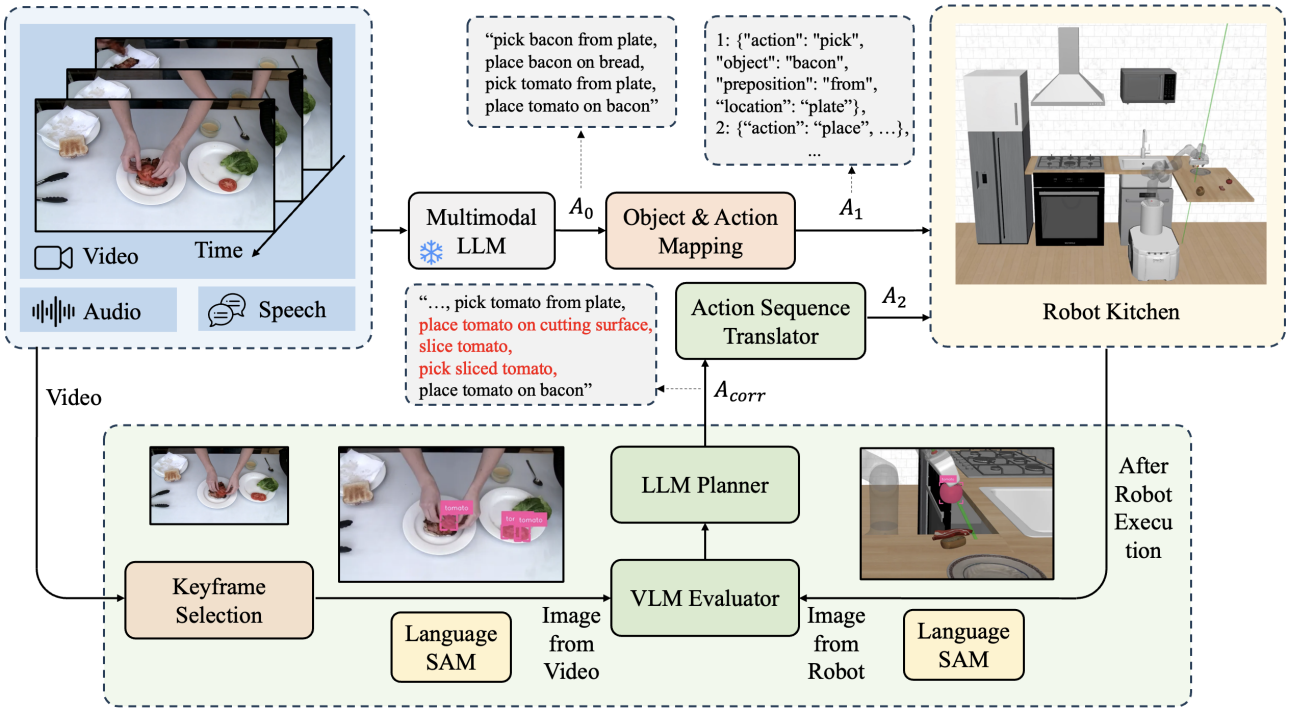

Fig. 3. **Overall architecture of KitchenVLA.** The top panel illustrates the initial plan generation process, where a multimodal LLM processes human video to produce an action sequence. This sequence is grounded into robot-executable actions via object and action mapping and executed in the robot environment. The bottom panel depicts the feedback-driven correction loop: visual keyframes from the human video and observations from the robot are compared by a VLM evaluator to detect domain mismatches. An LLM planner then generates corrective actions, which are translated into an updated sequence. This iterative process enables KitchenVLA to adapt human demonstrations to the robot environment with improved executability.

skills. Formally, given a human action step $a_0^i \in A_0$, we define the action mapping as:

$$\text{map}_A(a_0^i) = a_1^i, \tag{4}$$

where $a_1^i \in A_1$ denotes one or more corresponding robot skills selected from the robot's predefined action library.

3) Initial Action Sequence Generation: By applying the object and action mappings, we generate an initial robot-executable action sequence A_1 from the extracted human plan A_0:

$$A_1 = \text{map}(A_0) = \left[\text{map}_A(a_0^1), \text{map}_A(a_0^2), \dots\right], \tag{5}$$

where each mapped action is grounded to the most similar object according to map_O.

This heuristically generated sequence A_1 serves as the starting point for KitchenVLA's feedback-driven iterative refinement process described in the following sections.

B. Robot Execution with Feedback Collection

Although the initial action sequence A_1 is heuristically constructed through object and action mappings, successful execution in a robotic environment is not guaranteed. Due to differences in embodiment, environment, and the MLLM annotators' lack of awareness of robotic environment, robots may encounter execution failures even when following accurate human instructions (such as the human labels of the cooking videos).

During the execution of each action in A_1, the robot collects feedback that supports downstream correction and replanning. Failures can typically be attributed to three types of mismatches: 1) *Object Condition Mismatch*: the object in the robot's environment may differ in type, condition, or appearance from the corresponding object in the human video. 2) *Robotic Action Limitation*: the robot may be unable to perform the intended action due to its ambiguous description or being unavailable in the robot skill library. 3) *Logical Inconsistency*: The action sequence may violate capability logic, such as attempting to place an object before picking it when using a single-arm robot. These issues often result from a combination of object mismatches and action misinterpretations.

According to these possible mismatches, we collect multi-modal feedback after each action execution. Specifically, for each executed action a_i, we record a tuple:

$$f_i = (s_i, e_i, I_i), \tag{6}$$

where s_i is the similarity score from object/action mappings, e_i is a binary success flag from the skill execution interface, and I_i is the post-execution RGB observation captured by the robot camera. These signals are used by the VLM Evaluator to detect discrepancies and guide subsequent plan correction.

C. Iterative Action Corrections with VLM Evaluation

To correct the action sequences for achieving the task goal of human videos, we incorporate an iterative feedback-driven planning loop to refine actions based on observed execution

results and the original videos. This loop, visualized in Fig. 3, and detailed in Algorithm 1, forms the core of our framework's ability to adapt the actions to robotic environments.

After executing the action sequence A_n, the robot collects the feedback triplet f_i for each executed step a_n^i. These feedbacks are then passed to the VLM Evaluator, which compares the robot observation I_i and the corresponding keyframe from the human video. The evaluator outputs domain gap analysis and the action step similarity scores:

$$(T_{\text{gap}}, R_{\text{score}}) = F_{\text{eval}}(a_n^i, a_n^{i+1}, I_v, I_r). \tag{7}$$

In this process, the VLM will output as the following format:

```
Gaps:
1. Object coherence: The objects involved are similar
(bacon and bread), but the plate and additional
objects differ.
2. Motion feasibility: The human uses hands while
the robot uses a gripper, leading to potential
differences in manipulation.
3. Sequence logic correctness: The sequence of
actions is consistent between the human and the
robot.
Score: 0.5
```

The LLM Replanner then takes the analysis, scores, and an action window to replan the current step accordingly:

$$A_i^{\text{corr}} = F_{\text{replan}}(a_n^i, a_n^{i+1}, T_{\text{gap}}, R_{\text{score}}). \tag{8}$$

The corrected sub-sequences are collected into A_{corr}, and translated into a new full sequence for the next iteration:

$$A_{n+1} = F_{\text{translate}}(A_{\text{corr}}). \tag{9}$$

```
act_trans: [
{ "action": "pick", "object": "bacon", "preposition":
"from", "location": "plate" },
{ "action": "place", "object": "bacon",
"preposition": "on", "location": "bread" },
...]
```

This process is repeated for a maximum of k iterations and then the final action sequence A_{final} is taken as A_{k+1}.

Algorithm 1 Iterative Corrections with VLM Evaluation

Require: Video $\mathbf{v}$, Initial Sequence $\mathcal{A}_0 = [a_0^0, a_0^1, a_0^2, \dots]$, Initial mapping `map`, Max Iterations k
Ensure: Final Action Sequence $\mathcal{A}_{\text{final}}$
  $\mathcal{A}_1 \leftarrow \texttt{map}(\mathcal{A}_0)$
  for $n = 1$ to k **do**
    $\mathcal{A}_{\text{corr}} \leftarrow [\,]$
    for a_n^i in $\mathcal{A}_n$ **do**
      $\texttt{rob.exec}(a_n^i)$
      $\texttt{I}_v \leftarrow f_{\text{keyframe}}(\mathbf{v})$
      $\texttt{I}_r \leftarrow \texttt{rob.get\_obs}(t_i)$
      $(T_{\text{gap}}, R_{\text{score}}) \leftarrow F_{\text{eval}}(a_n^i, a_n^{i+1}, \texttt{I}_v, \texttt{I}_r)$
      $\mathcal{A}_i^{\text{corr}} \leftarrow F_{\text{replan}}(a_n^i, a_n^{i+1}, T_{\text{gap}}, R_{\text{score}})$
      $\mathcal{A}_{\text{corr}} \leftarrow \mathcal{A}_{\text{corr}} \bigcup \mathcal{A}_i^{\text{corr}}$
    $\mathcal{A}_{n+1} \leftarrow F_{\text{translate}}(\mathcal{A}_{\text{corr}})$
  return $\mathcal{A}_{\text{final}} \leftarrow \mathcal{A}_{k+1}$

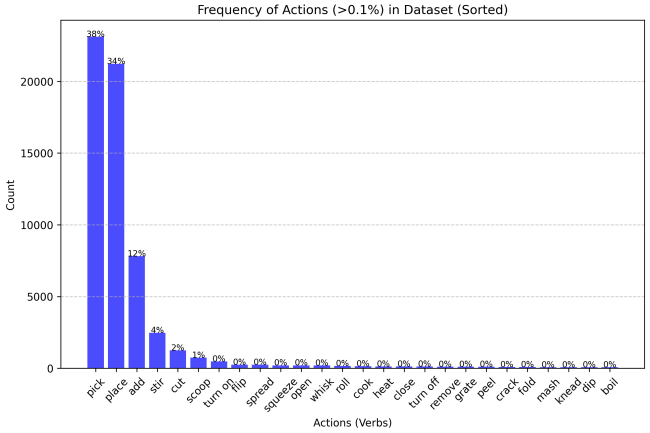

Fig. 4. **Distribution of action verbs in the dataset.** We filtered to show actions with frequency $> 0.1\%$. The task space is dominated by primitive actions like *pick*, *place*, and *add*. This long-tail distribution reflects the complexity of kitchen tasks and motivates our correction approach.

V. Experimental Results

We evaluate the effectiveness of our proposed framework in bridging the gap between human demonstrations and robotic execution. This section presents the experimental setup, analysis, and results across a wide range of kitchen videos.

A. Experimental Setups

Environment and Dataset. Our robotic experiments is conducted in the RoboCasa simulation environment [37] using a mobile manipulator equipped with a Franka Emika Panda arm and a pseudo suction gripper. The robot performs task executions based on human instructions extracted from the YouCook dataset [36], which offers diverse cooking demonstrations. In this work, we use 867 videos (filtered by the available robot skill library) for our evaluation. We further utilize fine-grained robot action annotations of the YouCook videos from the AVBLIP-2 work [5] to support instruction-to-execution comparisons.

Evaluation Metrics. Since ground-truth corrected action sequences for robot execution are not available due to the need for human annotation (which we consider as ongoing work) and the YouCook dataset does not provide action labels transferable to a specific robotic environment, we adopt three evaluation metrics:

- **LLM-based Judgment:** We provide a large language model (LLM) with the goal of the cooking task, such as *"prepare a pair of buns"*, and prompt it to judge which action sequence better fulfills the task. The result is denoted as:

$$\text{Score}_{\text{LLM}} = \text{Judge}_{\text{LLM}}(\text{Goal}, A) \tag{10}$$

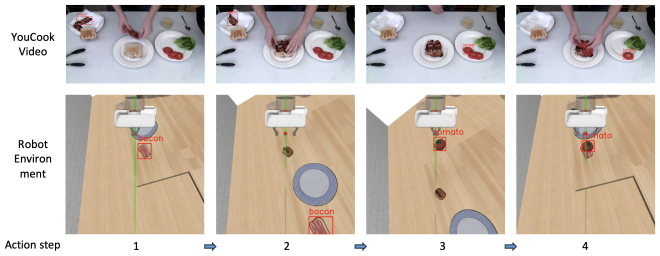

Fig. 5. **Action sequence before correction.** This figure shows the action sequence: pick bacon from plate, place bacon on bread, pick tomato from plate, place tomato on bacon.

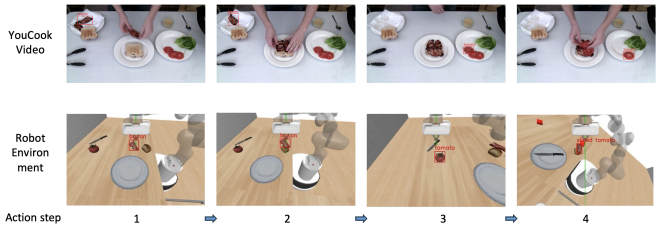

Fig. 6. **Action sequence after correction (ours).** This figure shows the action sequence: pick bacon from plate, place bacon on bread, pick knife, slice tomato, place knife, pick sliced tomato, place sliced tomato on bacon

- **VLM-based Judgment:** In addition to the goal description, we provide a vision-language model (VLM) with an initial image of the robot environment (where all necessary objects are placed on the counter). The model then evaluates which action sequence is more appropriate for that context:

$$\text{Score}_{\text{VLM}} = \text{Judge}_{\text{VLM}}(\text{Goal}, I_{\text{init}}, A) \tag{11}$$

- **Human Evaluation (AMT):** We collect corrected action sequences from human annotators on Amazon Mechanical Turk (AMT) and compare them with the model-generated sequences. The similarity is measured using standard NLP metrics, BLEU-4, METEOR, and a custom Task Completion Ratio (TCR), which represents the percentage of video clips where the generated plan matches the human-corrected sequence completely.

In addition, we qualitatively compare the textual action sequences and visualize the robot execution results through image snapshots to showcase the effectiveness of our framework.

Implementation Details. We use GPT-4o [1] as both the VLM and LLM agents in our framework. LanguageSAM [38] is employed for language-guided object segmentation, and the MLLM component leverages the AVBLIP-2 work [5]. Robot control relies on a predefined skill library and an operational space controller (OSC). All experiments are conducted on an A10 GPU server.

B. Analysis of Domain Gap

To better understand the challenges of transferring human demonstrations to robot-executable plans, we analyze the distribution of action verbs in the YouCook dataset. As shown in Fig. 4, the videos exhibit a wide variety of actions, reflecting the inherent complexity of kitchen tasks. However, due to the limitations of the robot's skill library, we adopt a predefined mapping strategy augmented with feedback-driven correction and refinement to improve the robustness of action transfer.

TABLE I
EVALUATION SCORES OF ACTION SEQUENCES BY LLM AND VLM JUDGES

Action Sequence	LLM Judge	VLM Judge
A_1 (before replan)	0.45	0.35
A_2 (after replan)	0.67	0.65

We observe that the action distribution follows a long-tail pattern, where a small number of primitive actions, such as *pick*, *place*, and *add*, account for the majority of steps. This distribution motivates our design: focusing on reliable execution of high-frequency actions can already cover a large portion of real-world tasks.

Nonetheless, we also find that execution failures still occur even for seemingly simple actions like *pick* and *place*. These failures often stem not from the action type itself, but from discrepancies in object state between the video and the robot environment (e.g., sliced vs. whole ingredients). This highlights the necessity of addressing object state mismatches in order to enable accurate and robust skill transfer.

C. Effectiveness Evaluation of Feedback Loop

To further assess the effectiveness of our feedback-based corrections, we compare the quality of action sequences before and after iterative refinement. As shown in Table I, we report evaluation scores from both an LLM and a VLM acting as external judges. The original action sequence A_1 is derived directly from the video demonstration without correction, while A_2 is the refined version produced by KitchenVLA after iterative feedback and replanning.

Both the LLM and VLM judges consistently rate A_2 higher than A_1, indicating that our framework improves the semantic and contextual alignment of the action plan with the intended task. In particular, the improvement in LLM-based scores (from 0.45 to 0.67) suggests enhanced task-level reasoning, while the VLM-based improvement (from 0.35 to 0.65) reflects better consistency between the corrected plan and the robot environment. These results highlight the effectiveness of incorporating multimodal feedback to bridge domain gaps and optimize robot actions.

D. Human Evaluation and Importance of Keyframe Quality

To evaluate the quality of action sequence generation, Amazon Mechanical Turk (AMT) workers corrected the action sequence generated using KitcheVLA. We evaluated the sequence similarity between before and after the human correction. As shown in Table III, BLEU-4, METEOR, and Task Completion Ratio (TCR), were applied as quantitative metrics. TCR stands for the number of the video clips achieves 100% correct action sequence generation. VLM_IS

TABLE II
COMPARISON OF ORIGINAL AND CORRECTED ACTION SEQUENCES

Video Id	Sequence Type	Action Sequence
s99K_WyajB8	Original Sequence	pick onion, place onion to pan
	Corrected Sequence (Ours)	pick onion, place onion to cutting board, pick knife, cut onion, place knife, pick onion, place onion to pan, press start
TFJ6oR89Vb8	Original Sequence	pick egg from plate, add egg on pan
	Corrected Sequence (Ours)	pick egg from plate, crack egg into bowl, whisk egg mixture, pour egg mixture into pan
vZ7Pz9jM7zk	Original Sequence	pick chicken breast, place chicken breast in egg mixture
	Corrected Sequence (Ours)	pick chicken breast, place chicken breast in egg mixture, press chicken breast in egg mixture
Xp2HNiLabRI	Original Sequence	pick butter stick from plate, place butter stick on bread, pick bread from plate, place bread on pan
	Corrected Sequence (Ours)	pick butter stick from plate, place butter stick on bread, pick knife from drawer, spread butter on bread, pick bread from plate, place bread on pan, press start button on stove
8lqdPpg3w08	Original Sequence	pick egg from plate, place egg on grill, pick bottled_water, add bottled_water on grill
	Corrected Sequence (Ours)	pick spatula, pick egg with spatula, place egg on grill, place spatula back, pick bottled_water, move_to grill, pour bottled_water on grill, place bottled_water on table

TABLE III
QUALITY OF GENERATED ACTION SEQUENCE.
TCR: TASK COMPLETION RATIO

	All			Aligned		
	BLEU-4	**METEOR**	**TCR**	**BLEU-4**	**METEOR**	**TCR**
VLM_IS	0.496	0.388	0.27	0.591	0.431	0.28

refers to our full method, where both the action step texts and keyframe image are used for VLM-based evaluation and correction. Although we tested 867 video clips in total, there remained key frame selection errors. We separately evaluated the 357 video clips where the key frame selection was correct and the given steps were aligned with the key frame images. The quality using the aligned key frames outperformed in every metric. This indicates the accurate key frame selection is essential.

E. Textual Comparison of Action Correction

To qualitatively evaluate the improvements introduced by our correction framework, we present side-by-side comparisons of original and corrected action sequences in Table II. Each example shows the output before and after feedback-driven refinement, based on video demonstrations from the YouCook dataset.

We observe that the corrected sequences generated by KitchenVLA include additional actions that are necessary for successful task execution but are missing from the original video-derived sequences. For example, in the first row, actions such as *"pick knife from drawer"*, *"spread butter on bread"*, and *"press start button on stove"* are inserted, reflecting both object-state awareness and task completeness. In another example, the system automatically inserts intermediate steps like *"place onion to cutting board"*, *"pick knife"*, and *"cut onion"*, demonstrating its capacity to reason about physical prerequisites and proper tool usage. These results highlight KitchenVLA's ability to adapt high-level plans to the robot's operational context and resolve implicit gaps in human instructions.

F. Visualization of Action Sequence Refinements

To visually illustrate the effectiveness of our feedback loop, we provide a demonstration comparing the robot's performance in two consecutive execution loops, as shown in Fig. 5 and Fig. 6. The first execution follows the initial action sequence directly inferred from video demonstrations, often leading to execution failures due to domain discrepancies. The second execution incorporates our framework's corrections, demonstrating improved task alignment and success rates. This comparison highlights the system's ability to refine robotic behavior iteratively and adapt to environment variations.

VI. CONCLUSIONS

In this work, we proposed KitchenVLA, a vision-language-action framework that bridges the gap between human instructional videos and robotic execution through object/action mapping, visual feedback, and iterative plan refinement. By integrating VLM-based evaluation and LLM-guided replanning, our system dynamically adapts robotic action sequences to real-world constraints while preserving the semantics of the original human demonstrations. Preliminary results suggest the potential of KitchenVLA for robust execution in kitchen environments.

In future work, we will extend KitchenVLA to real-world robotic platforms and explore its generalization to other long-horizon manipulation domains beyond kitchen environments. We also plan to investigate more robust keyframe selection and grounding mechanisms to further improve the reliability of feedback-based plan correction.

REFERENCES

[1] O. AI, "Hello gpt-4o," https://openai.com/index/hello-gpt-4o/, 2024.

[2] A. D. et al., "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[3] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[4] K. Hori, K. Suzuki, and T. Ogata, "Interactively robot action planning with uncertainty analysis and active questioning by large language model," in *Proc. SII*, 2024.

[5] C. Hori, M. Kambara, K. Sugiura, K. Ota, S. Khurana, S. Jain, R. Corcodel, D. Jha, D. Romeres, and J. Le Roux, "Interactive robot action replanning using multimodal llm trained from human demonstration videos," in *International Conference on Acoustics, Speech and Signal Processing*, 2025, pp. 1–5.

[6] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[8] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," in *Conference on Robot Learning*, 2024, pp. 2679–2713.

[9] B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do as I can, not as I say: Grounding language in robotic affordances," in *Conference on Robot Learning*, 2022, pp. 287–318.

[10] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: An embodied multimodal language model," 2023. [Online]. Available: https://arxiv.org/abs/2303.03378

[11] 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," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[12] T. Carta, C. Romac, T. Wolf, S. Lamprier, O. Sigaud, and P.-Y. Oudeyer, "Grounding large language models in interactive environments with online reinforcement learning," 2024. [Online]. Available: https://arxiv.org/abs/2302.02662

[13] S. S. Raman, V. Cohen, I. Idrees, E. Rosen, R. Mooney, S. Tellex, and D. Paulius, "CAPE: corrective actions from precondition errors using large language models," in *International Conference on Robotics and Automation*. IEEE, 2024, pp. 14 070–14 077.

[14] Z. Liu, A. Bahety, and S. Song, "REFLECT: summarizing robot experiences for failure explanation and correction," in *Conference on Robot Learning*, 2023, pp. 3468–3484.

[15] M. Shirasaka, T. Matsushima, S. Tsunashima, Y. Ikeda, A. Horo, S. Ikoma, C. Tsuji, H. Wada, T. Omija, D. Komukai, and Y. M. Y. Iwasawa, "Self-recovery prompting: Promptable general purpose service robot system with foundation models and self-recovery," 2023. [Online]. Available: https://arxiv.org/abs/2309.14425

[16] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2024, pp. 26 286–26 296.

[17] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. J. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2024, pp. 14 455–14 465.

[18] C. Ma, K. Lu, T.-Y. Cheng, N. Trigoni, and A. Markham, "Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors," in *Proceedings of the Conference on Neural Information Processing Systems*, 2024.

[19] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "OKAMI: teaching humanoid robots manipulation skills through single video imitation," in *Conference on Robot Learning*, 2024, pp. 299–317.

[20] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning*, 2021, pp. 1678–1690.

[21] S. Belkhale, Y. Cui, and D. Sadigh, "HYDRA: hybrid robot actions for imitation learning," in *Conference on Robot Learning*, 2023, pp. 2113–2133.

[22] B. Han, J. Kim, and J. Jang, "A dual process VLA: efficient robotic manipulation leveraging VLM," *arXiv preprint arXiv:2410.15549*, 2024.

[23] D. Noh, H. Nam, K. Gillespie, Y. Liu, and D. W. Hong, "YORI: autonomous cooking system utilizing a modular robotic kitchen and a dual-arm proprioceptive manipulator," *arXiv preprint arXiv:2405.11094*, 2024.

[24] B. Wang, J. Zhang, S. Dong, I. Fang, and C. Feng, "VLM see, robot do: Human demo video to robot action plan via vision language model," *arXiv preprint arXiv:2410.08792*, 2024.

[25] C. R. Nallu, "Task tree retrieval for robotic cooking," *arXiv preprint arXiv:2312.09434*, 2023.

[26] A. Mavrogiannis, C. I. Mavrogiannis, and Y. Aloimonos, "Cook2ltl: Translating cooking recipes to LTL formulae using large language models," in *International Conference on Robotics and Automation*. IEEE, 2024, pp. 17 679–17 686.

[27] H. Wang, K. Kedia, J. Ren, R. Abdullah, A. Bhardwaj, A. Chao, K. Y. Chen, N. Chin, P. Dan, X. Fan, G. Gonzalez-Pumariega, A. Kompella, M. A. Pace, Y. Sharma, X. Sun, N. Sunkara, and S. Choudhury, "MOSAIC: A modular system for assistive and interactive cooking," *arXiv preprint arXiv:2402.18796*, 2024.

[28] C. C. Beltran-Hernandez, N. Erbetti, and M. Hamaya, "Sliceit! - A dual simulator framework for learning robot food slicing," in *International Conference on Robotics and Automation*. IEEE, 2024, pp. 4296–4302.

[29] A. Bhaskar, R. Liu, V. D. Sharma, G. Shi, and P. Tokekar, "LAVA: long-horizon visual action based food acquisition," in *International Conference on Intelligent Robots and Systems*. IEEE, 2024, pp. 8929–8935.

[30] B. Lin, Y. Ye, B. Zhu, J. Cui, M. Ning, P. Jin, and L. Yuan, "Video-llava: Learning united visual representation by alignment before projection," *arXiv preprint arXiv:2311.10122*, 2023.

[31] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, "Video-chatgpt: Towards detailed video understanding via large vision and language models," *arXiv preprint arXiv:2306.05424*, 2023.

[32] F. Yang, C. Ma, J. Zhang, J. Zhu, W. Yuan, and A. Owens, "Touch and go: Learning from human-collected vision and touch," *Advances in Neural Information Processing Systems*, pp. 8081–8103, 2022.

[33] F. Yang and C. Ma, "Sparse and complete latent organization for geospatial semantic segmentation," in *Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1809–1818.

[34] C. Fu, Y. Dai, Y. Luo, L. Li, S. Ren, R. Zhang, Z. Wang, C. Zhou, Y. Shen, M. Zhang, P. Chen, Y. Li, S. Lin, S. Zhao, K. Li, T. Xu, X. Zheng, E. Chen, R. Ji, and X. Sun, "Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis," *arXiv preprint arXiv:2405.21075*, 2024.

[35] M. Kambara, C. Hori, K. Sugiura, K. Ota, D. K. Jha, S. Khurana, S. Jain, R. Corcodel, D. Romeres, and J. Le Roux, "Human action understanding-based robot planning using multimodal LLM," in *Proc. ICRA Workshop on "Cooking Robotics: Perception and Motion Planning"*, 2024.

[36] L. Zhou, C. Xu, and J. J. Corso, "Towards automatic learning of procedures from web instructional videos," in *Proc. AAAI*, 2018.

[37] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," in *RSS 2024 Workshop: Data Generation for Robotics*.

[38] L. Medeiros, "lang-segment-anything," https://github.com/luca-medeiros/lang-segment-anything, 2023, accessed: 2025-05-01.