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

Compositional Context Fine-Tuning Vision-Language Model for Complex Assembly Action Understanding from Videos

Hao Zheng*[,1,4], Jinyi Huang[4], Tiantian Zheng[1,2], Xun Xu[4], Tuka Alhanai[1,3]

***Abstract*— Assembly action understanding is a key enabler for effective human-robot collaborative assembly, yet it remains challenging due to subtle motions and fine-grained hand–object interactions. We adapt vision-language models (VLMs) to this challenging domain with Compositional Context Fine-Tuning (CCFT), a method that decomposes assembly actions into semantic elements (*Verb*, *Object*, *Tool*) and fine-tunes VLMs to recognize each action element using templated question-answering pairs. This approach ensures near-deterministic outputs. To enable efficient and effective multi-task learning under limited data, a Layer-Partitioned Alternating Training (LP-AT) method is presented, which assigns distinct model layers to recognize specific action elements through element-specific low-rank adapters. LP-AT alternates weight updates across element-specific adapters, reducing cross-task interference while enabling per-adapter hyperparameter optimization. Furthermore, we create HA-ViD-VQA and IKEA-ASM-VQA datasets from existing assembly video datasets. Extensive experiments on these datasets demonstrate that our method consistently outperforms strong action recognition baselines while providing interpretable element-level predictions that can support diverse downstream applications. Code and dataset are released at `https://github.com/x-labs-xyz/CCFT`.**

I. Introduction

Building on the strong textual reasoning of large language models (LLMs), vision–language models (VLMs) have advanced multimodal understanding by integrating visual and linguistic cues [1], [2]. Multimodal perception enables VLMs to tackle a broader range of real-world challenges, positioning them as promising foundations for human-robot collaboration (HRC), which demands highly precise environmental understanding and physical interaction.

Assembly action understanding represents a critical component of collaborative robotics, as robots must comprehend complex assembly actions to effectively assist humans and acquire manipulation skills through demonstration [3], [4]. However, assembly action understanding poses unique challenges for robotic perception: subtle and intricate actions, distinct actions with similar motions, identical actions with distinct motions, and nuanced hand-object interactions. While VLMs offer considerable potential for advancing complex assembly action understanding, this application domain remains largely unexplored.

Most existing VLM research targets general scene understanding, supporting only basic video summarization or simple question-answering (QA) tasks [5], and lacking specialized adaptions for recognizing complex assembly actions. Moreover, HRC applications demand deterministic outputs, which conflicts with the generative nature of VLM outputs.

This paper proposes two complementary technical ingredients to address these challenges. First, a *Compositional Context Fine-Tuning* (CCFT) method is proposed to decompose assembly actions into semantic elements—*Verb*, *Object*, and *Tool*, and fine-tune VLMs to recognize each element using templated visual question-answering (VQA) pairs. Compared to traditional paradigms, CCFT ensures near-deterministic outputs through templated VQA queries under limited answer spaces and yields interpretable element-level outputs that can support downstream HRC applications. Second, to effectively and efficiently fine-tune VLMs to recognize action elements under limited data, we propose a *Layer-Partitioned Alternating Training* (LP-AT) method. LP-AT assigns disjoint layer groups to individual elements and trains element-specific low-rank adapters (LoRA) [6] applied to specific groups. LP-AT alternately optimizes each element's adapter, thereby confining gradient updates to element-specific parameter subspaces and reducing interference while preserving parameter efficiency. Additionally, LP-AT offers granular hyperparameter tuning (e.g., layer selection, learning rate, rank) flexibility for each adapter.

To validate our approach, we reformulate two existing assembly video datasets, HA-ViD [7] and IKEA-ASM [8], into compositional VQA datasets (HA-ViD-VQA and IKEA-ASM-VQA) with templated QA pairs and action element-level annotations. A state-of-the-art VLM (Qwen2.5-VL [9]) is fine-tuned and evaluated on these datasets using our method. The experiments show that our method consistently outperforms strong action recognition baselines while providing interpretable element-level predictions that can support diverse downstream applications.

Our main contributions are:

- CCFT is proposed to adapt VLMs for assembly action understanding, which decomposes assembly actions into semantic elements and fine-tunes VLMs to recognize each element, yielding near-deterministic and interpretable element-level predictions.
- LP-AT is introduced to facilitate multi-task learning, which assigns disjoint layer groups to subtasks and alternately optimizes task-specific LoRA adapters to reduce cross-task interference and enable per-task hy-

*Hao Zheng is corresponding author: h.zheng@nyu.edu; [1]Department of Computer Engineering, New York University Abu Dhabi, UAE; [2]Center for Quantum and Topological Systems, New York University Abu Dhabi, UAE; [3] Center for AI and Robotics, NYUAD, UAE; [4] Department of Mechanical and Mechatronics Engineering, The University of Auckland, New Zealand. H.Z.: hzhe951@aucklanduni.ac.nz; J.H: jhua658@aucklanduni.ac.nz; X.X: x.xu@auckland.ac.nz. T.A acknowledges support by CAIR and CQTS funded by Tamkeen NYUAD Research Institute Award CG010 and CG008, respectively.

perparameter tuning.
- Two compositional VQA assembly video datasets, HA-ViD-VQA and IKEA-ASM-VQA, are released.

II. Related Work

a) Assembly Action Understanding for Human-Robot Collaborative Assembly: Assembly action understanding is fundamental to enabling effective human-robot collaborative assembly, where robots must comprehend human assembly actions to provide timely assistance, learn manipulation skills, and ensure seamless task coordination [3], [4]. General video action understanding has advanced considerably by leveraging convolutional neural networks [10], [11], skeleton-based methods [12], and transformer architectures [13], [14]. However, assembly actions present distinct and formidable challenges which include the inherent subtlety and intricacy of manual operations, distinct actions with similar motion patterns, identical actions with distinct motion patterns, and fine-grained hand-object interactions. To address these challenges, researchers have explored enhancing the general action understanding methods by explicitly modeling human-object interactions [15] and leveraging detailed human and object pose information [16]. However, we note the semantic ambiguity inherent in assembly actions, where visual similarity does not imply semantic equivalence (e.g. tightening a screw or loosening a screw both show a screwdriver turning but semantically they represent opposite actions). This gap motivates our exploration of VLMs in assembly action understanding.

b) Vision–Language Models for Assembly Action Understanding: VLMs have extended LLM capabilities to multimodal tasks, enabling joint reasoning across images, videos, and text [1], [2]. Early foundational models, such as CLIP [17] and ALIGN [18], pioneered contrastive image-text alignment, demonstrating strong performance on image-text tasks. More recently, unified multimodal architectures such as LLaVA-Video [19], and Qwen2.5-VL [9] have further extended these capabilities to video understanding [20]. However, these models remain primarily designed for general tasks, such as video summarization and simple QA [5], and have been largely underexplored for specialized domains such as assembly video understanding. Given the strong visual perception and reasoning abilities of VLMs, they hold significant potential to address the unique challenges of assembly action understanding. Adapting these powerful generalist models to the manufacturing domain typically requires fine-tuning. Common strategies include full-model updates or parameter-efficient techniques such as LoRA [6] and its variants [21], [22], which are particularly crucial for managing computational costs. Crucially, HRC applications demands exceptionally low tolerance for output ambiguity, challenging VLMs' tendency toward hallucination. Furthermore, while existing assembly video datasets, such IKEA-ASM [8], Assembly101 [23], and HA-ViD [7] lack structured VQA pairs necessary for VLM adaption.

c) Compositional Visual Question Answering: Compositionality, the principle that complex meanings emerge from the combination of simpler elements, is fundamental to human intelligence and essential for computer vision where the scene can be described by their components, including objects, their attributes, and their relationships [24]. Building on this principle, researchers have developed compositional VQA approaches for VLMs that explicitly decompose complex visual queries into interpretable subcomponents, employing compositional prompting strategies and causal or scene graph modeling [25], [26]. Compositional VQA enhances reasoning transparency and supporting generalization to unseen scenes. However, these works remain predominantly focused on image understanding, with limited exploration in video domains. For assembly video understanding, Zheng et al. [27] demonstrated the effectiveness of compositional understanding by decomposing actions into four elements (*verb*, *manipulated object*, *target object* and *tool*) and developing separate action segmentation models for each element, subsequently recombining the segmented elements into holistic actions. However, this work relied on traditional approaches and leveraged heuristic reasoning mechanisms to recombine the action elements to form the holistic actions. Given the multimodal reasoning strengths of VLMs, it is promising to apply compositional VQA to assembly action understanding using VLMs.

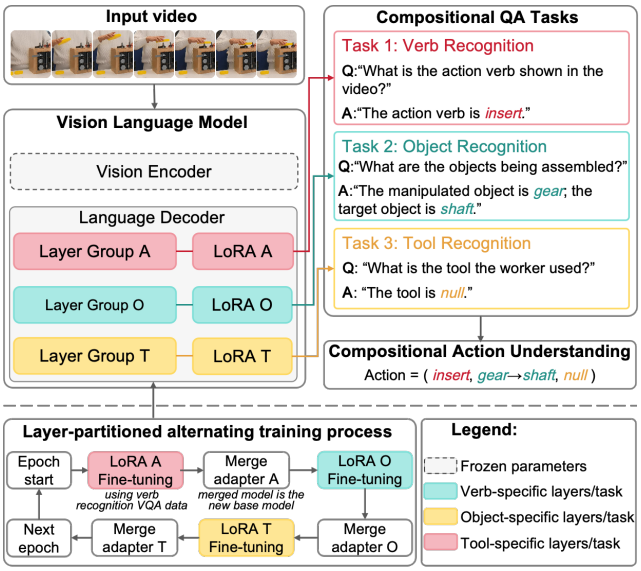

Fig. 1: The Overall Framework of Compositional Context Fine-Tuning with Layer-Partitioned Alternating Training.

III. Methodology

Fig. 1 presents our overall framework of CCFT and LP-AT. In this section, we formalize the problem (Section III-A), describe the construction of compositional VQA datasets (Section III-B), introduce the method of CCFT (Section III-C) and LP-AT (Section III-D), and define the evaluation protocol (Section III-E).

A. Problem Formulation

We formulate assembly action understanding as a compositional multimodal learning problem. Given an assembly video clip $x = \{f_1, f_2, ..., f_n\}$ where f_n represents the n-th

frame, our objective is to identify the assembly action $a \in \mathcal{A}$ shown in the video x, where $\mathcal{A}$ is the set of all possible actions. Traditional action recognition methods directly map $x \rightarrow a$, which is a classification problem given the predefined action class set $\mathcal{A}$. In contrast, our method decomposes each assembly action a into semantic action elements: Verb (v), Object (o), and Tool (t). Based on the target application (e.g. on different datasets), the specific action elements can vary. Formally, we aim to train a VLM to generate responses to the queries for each action element, producing (v, o, t), where (v, o, t) can be reconstructed to the corresponding action a. Since our approach trains the VLM on VQA tasks, it is not necessary to explicitly provide predefined class sets $\mathcal{V}$, $\mathcal{O}$, and $\mathcal{T}$ for each action element. Instead, the model learns to generate near-deterministic responses in natural language from the limited answer space of each element, providing greater flexibility and scalability.

B. Dataset Construction

We reformulate two existing assembly video datasets, HA-ViD [7] and IKEA ASM [8], into compositional VQA format datasets, HA-ViD-VQA and IKEA-ASM-VQA, facilitating compositional reasoning over fine-grained action elements. Given a video instance $x \in \mathcal{X}$, where $\mathcal{X}$ denotes the entire set of video samples, we define its ground truth action annotation as a structured tuple $a \in \mathcal{A}$, with the composition depending on the dataset's original annotation structure. For HA-ViD, which provides structured action labels with four distinct elements—verb (v), manipulated object (o_m), target object (o_t), and tool (t), we define $a = (v, o_m, o_t, t) \in \mathcal{A}_{HA-ViD}$. In contrast, IKEA ASM contains only verb and object annotations, leading to: $a = (v, o) \in \mathcal{A}_{IKEA}$. Each element $a_i \in a$ is queried independently using a templated natural language question q_i, yielding an answer $\hat{a}_i$ from the VLM. This decomposition transforms the holistic action understanding task into a set of compositional QA subtasks, $\mathcal{T} = \{\mathcal{T}_v, \mathcal{T}_o, \mathcal{T}_t\}$, where each subtask $\mathcal{T}_i$ targets the recognition of one action element. Taking an HA-ViD sample for instance:

- q_v: "What is the action verb shown in the video?"
  a_v: "The action verb is *screw*."
- q_o: "What are the objects being assembled?"
  a_o: "The manipulated object is *nut*. The target object is *bolt*."
- q_t: "What is the tool the worker used in the video?"
  a_t: "The tool is *wrench*."

C. Compositional Context Fine-Tuning

The VLM model is fine-tuned in a multi-task manner using compositional supervision. Specifically, given a video x and a templated question q_i, the model is trained to maximize the conditional likelihood $p(a_i|x, q_i)$, where a_i is the answer. Unlike open-ended VQA, CCFT leverages low-entropy and template-based supervision to minimize ambiguity and improve semantic precision.

Let θ_b denote the base parameters of the pretrained VLM. To enable multi-tasking and efficient training, a subtask-specific low-rank adapter (LoRA) [6] $\Delta\theta_i$ is introduced for each subtask $\mathcal{T}_i$, and the adapted model parameters are defined as:

$$\theta = \theta_b + \sum_i \Delta\theta_i \tag{1}$$

The training objective for each subtask is to minimize the negative log-likelihood:

$$\mathcal{L}_i = -\mathbb{E}_{(x,q_i,a_i)\sim\mathcal{D}_i} \left[\log p_{\theta_b+\Delta\theta_i}(a_i \mid x, q_i)\right] \tag{2}$$

where $\mathcal{D}_i$ denotes the dataset of annotated QA triplets for subtask $\mathcal{T}_i$. However, naively summing adapters across tasks leads to parameter interference and degrades performance on individual subtasks. To address this, we propose a *Layer-Partitioned Alternating Training* (LP-AT) method, detailed in Section III-D, which isolates task-specific updates by partitioning parameter subsets across model layers during fine-tuning.

Additionally, in this work, Qwen2.5-VL [28] is chosen as the base model. We only fine-tune the language encoder while keeping the vision encoder frozen, because (1) the pretrained vision encoder in Qwen2.5-VL is highly capable; and (2) fine-tuning the vision encoder on our relatively small datasets does not yield significant performance gains.

D. Layer-Partitioned Alternating Training

To enable multi-task adaptation while mitigating cross-task interference, we propose LP-AT. Let $L = \{l_1, l_2, \ldots, l_N\}$ represent the N layers of the VLM, partitioned into disjoint task-specific groups $\mathcal{G} = \{\mathcal{G}_v, \mathcal{G}_o, \mathcal{G}_t\}$, where each $\mathcal{G}_i \subseteq L$ contains the layers allocated to subtask $\mathcal{T}_i$, and $\bigcup_i \mathcal{G}_i = L$.

For each subtask $\mathcal{T}_i$, LP-AT applies a LoRA adapter into the designated layer group $\mathcal{G}_i$, with rank r_i, learning rate η_i, and scaling factor α_i. Let $\theta_b = \{W_{\mathcal{G}_i}\}_{\mathcal{G}_i\in\mathcal{G}}$ denote base model parameters, where $W_{\mathcal{G}_i} \in \mathbb{R}^{d\times d}$ is a weight matrix at layer group $\mathcal{G}_i$. The adapted weight at each $\mathcal{G}_i$ is defined as:

$$\hat{W}_{\mathcal{G}_i} = W_{\mathcal{G}_i} + \alpha_i A_{\mathcal{G}_i} B_{\mathcal{G}_i}, A_{\mathcal{G}_i} \in \mathbb{R}^{d\times r_i}, B_{\mathcal{G}_i} \in \mathbb{R}^{r_i\times d} \tag{3}$$

During training, the adapter parameters $\Delta\theta_i = \{A_{\mathcal{G}_i}, B_{\mathcal{G}_i}\}$ are optimized by minimizing the subtask-specific loss:

$$\min_{\Delta\theta_i} \mathcal{L}_i = -\mathbb{E}_{(x,q_i,a_i)\sim\mathcal{D}_i} \left[\log p_{\theta_b+\Delta\theta_i}(a_i \mid x, q_i)\right] \tag{4}$$

After each update, the adapter is merged into the base model:

$$\theta_b \leftarrow \theta_b + \Delta\theta_i \tag{5}$$

and the process proceeds to the next subtask $\mathcal{T}_{i+1}$. The updated base θ_b serves as the initialization for the following step. Once all subtasks are traversed, the next epoch begins.

This alternating method allows each subtask to selectively adapt a distinct subset of the network via ($\mathcal{G}_i$), with specific LoRA rank (r_i), scaling factor (α_i) and learning rate (η_i), supporting modular, compositional, and efficient fine-tuning.

E. Evaluation Protocol

In manufacturing scenarios, it is essential to ensure unambiguous and deterministic model outputs. Our model outputs the action elements encoded as strings in a structured format (see Section III-B). We adopt a compositional evaluation protocol to assess the discriminative accuracy at both the element and action levels. This protocol facilitates fine-grained analysis of model performance, providing actionable insights for diagnosis and targeted improvement.

Let the ground-truth labels be y_i and the model predictions $\hat{y}_i$ for each action element $i \in \{v, o, t\}$, where y_i and $\hat{y}_i$ are string labels, and they are extracted from the dataset annotation and model output, respectively.

We define the per-element accuracy as:

$$\mathrm{Acc}(i) = \mathbb{I}[\mathrm{sim}(\hat{y}_i, y_i) \geq \tau] \quad \text{for } i \in \{v, o, t\} \tag{6}$$

where $\mathrm{sim}(\cdot, \cdot) \in [0, 1]$denotes a normalized string similarity, and τ is a similarity threshold.

The predicted holistic action $\hat{y}_a$ is constructed by combining the predicted action elements $\hat{y}_i$ ($i \in \{v, o, t\}$) following the predefined format. The holistic action accuracy is then defined as:

$$\mathrm{Acc}(a) = \mathbb{I}\left[\mathrm{sim}(\hat{y}_a, y_a) \geq \tau\right] \tag{7}$$

where y_a denotes the ground-truth holistic action string. In our implementation, similarity is computed as the cosine similarity between label embeddings encoded by MiniLM [29]. τ is set to 0.95 to ensure robustness to minor formatting or encoding errors while preserving the near-exact semantic matches. Although this specific threshold value is an empirical choice that did not impact overall performance metrics in our experiments, the use of a configurable threshold enhances the robustness of the evaluation method.

IV. Experiment

A. Datasets

We select representative subsets from the HA-ViD [7] and IKEA-ASM [8] and convert their annotations into our structured VQA format (Section III-B), creating HA-ViD-VQA and IKEA-ASM-VQA[1].

HA-ViD-VQA captures two-handed assembly actions from three camera viewpoints. The dataset contains 511 left-hand videos (444 train/67 test) and 536 right-hand videos (452 train/84 test) per viewpoint. Each action is annotated with 4 elements: *verb*, *manipulated object*, *target object*, and *tool*. The vocabulary comprises 6 verbs, 38 objects (including manipulated and target objects), and 5 tools, yielding 56 distinct holistic actions.

IKEA-ASM-VQA focuses on furniture assembly actions captured from a top-view perspective, containing 559 videos with 473/86 train/test splits. Unlike HA-ViD, this dataset provides 2-element annotations (*verb*, *object*) without hand differentiation. The vocabulary includes 12 verbs and 11 objects, forming 24 holistic actions.

[1]More details in supplementary document: `https://github.com/x-labs-xyz/CCFT/tree/main/Supplementary`

B. Implementation Details

We use Qwen2.5-VL-7B as the base VLM and fine-tune its language decoder (28 layers). Each video keeps the original frame rate but is temporally downsampled to a maximum of 76 frames, and the processing rate is set to 2 fps, due to GPU memory constraints. For HA-ViD-VQA, we partition the layers 0–9 for verb recognition, 10–23 for manipulated and target object recognition, and 24–27 for tool recognition. This design is motivated by two considerations: (1) the verb $\rightarrow$ object $\rightarrow$ tool progression intuitively follows the compositional order in our action decomposition framework, which also parallels the developmental trajectory in human motor learning: basic motor skills $\rightarrow$ object manipulation $\rightarrow$ tool use [30], and (2) the layer allocation (10:14:4) corresponds to the estimated relative complexity of each recognition task and has been empirically validated through preliminary experiments (see supplementary document). AdamW optimizers using learning rates of 5×10^{-5}, 5×10^{-5}, 2×10^{-5} with cosine scheduling are applied to the three adapters respectively. For IKEA-ASM-VQA, layers 0–13 are allocated for verb recognition, and 14–27 for object recognition, with AdamW optimizers using a learning rate of 5×10^{-5} and cosine scheduling for both adapters. All adapters are applied with rank 256 and a scaling factor of 256. Training is conducted for 40 epochs with a per-device batch size of 1, distributed across four NVIDIA RTX A6000 GPUs (48GB each). We use a warmup ratio of 0.1 and gradient clipping at 1.0 to ensure stable training.

C. Main Results

1) Comparison with strong baselines: We benchmark our method against three representative action recognition models: a CNN-based model with temporal shift (TSM [11]), a unified transformer-based model (UniFormerV2 [13]), and a pre-trained masked autoencoder (VideoMAE V2 [14]). All models are evaluated on the HA-ViD-VQA and IKEA-ASM-VQA datasets for the holistic action recognition accuracy. A key methodological distinction is that our approach utilizes VQA annotation files, whereas the baselines employ traditional categorical action labels. Quantitative results comparing the accuracy of our method against the baselines are presented in Table I. While action recognition studies commonly report both Top-1 and Top-5 accuracies [11], [13], [14], our method produces direct semantic outputs rather than ranked scores, making Top-5 evaluation inapplicable. We therefore report only Top-1 accuracy using Equation 7.

Table I shows that our method consistently outperforms all baselines across both datasets. These results highlight the effectiveness of our compositional fine-tuning strategy for VLMs, which leverages fine-grained VQA annotations to improve assembly action understanding. Notably, right-hand action recognition tends to be less accurate than left-hand due to more complex motion patterns and frequent occlusions caused by right-handed workers [7]. This suggests that future research should explore approaches that better handle visual occlusions to further improve recognition performance.

TABLE I: Action Recognition Accuracy on HA-ViD-VQA and IKEA-ASM-VQA. The accuracy (%) for left-hand and right-hand action recognition is reported separately under side, front, and top views for HA-ViD-VQA. Our method outperforms baselines.

HA-ViD-VQA	Method	Left-hand Acc. (%)	Right-hand Acc. (%)
Side-view	TSM	44.78	20.24
	UniFormerV2	43.28	38.10
	VideoMAEv2	41.79	36.90
	Ours	**46.27**	**47.62**
Front-view	TSM	38.81	21.43
	UniFormerV2	43.28	36.90
	VideoMAEv2	40.29	29.76
	Ours	**49.25**	**42.86**
Top-view	TSM	38.81	22.62
	UniFormerV2	44.78	**38.10**
	VideoMAEv2	40.29	36.90
	Ours	**53.73**	**38.10**
IKEA-ASM-VQA	Method	Accuracy (%)	
	TSM	25.58	
	UniFormerV2	46.51	
	VideoMAEv2	52.32	
	Ours	**66.28**	

2) Action Element Recognition Performance: To further evaluate the compositional action understanding capability of our method, we report in Table II the recognition accuracy of individual action elements, as well as the holistic action on HA-ViD-VQA and IKEA-ASM-VQA datasets using Equation 6.

TABLE II: Element-wise Action Recognition Accuracy on HA-ViD-VQA and IKEA-ASM-VQA. For HA-ViD-VQA, we report the recognition accuracy (%) for each action element: Verb, Manipulated Object (MO), Target Object (TO), Tool, and the holistic Action. LH and RH denote left-hand and right-hand actions, respectively. SV, FV, and TV refer to side, front, and top views. Tool recognition achieves the highest accuracy, while verb recognition consistently outperforms object recognition. For IKEA-ASM-VQA, we report the recognition accuracy (%) for Verb, Object, and the holistic Action.

HA-ViD	Verb v	MO o_m	TO o_t	Tool t	Action a
LH-SV	61.19	55.22	53.73	88.06	46.27
LH-FV	62.69	58.21	53.73	95.52	49.25
LH-TV	71.64	68.66	61.19	92.54	53.73
RH-SV	64.29	59.52	58.33	96.43	47.62
RH-FV	58.33	52.38	51.19	96.43	42.86
RH-TV	52.38	48.81	48.81	97.62	38.10
IKEA	Verb v		Object o		Action a
	68.60		79.10		66.28

Considering the number of categories for each action element (see Section IV-A), the results in Table II reveal that recognition accuracy is closely related to task complexity. Tool recognition achieves the highest accuracy, partly because of the small number of tool categories and the fact that only 15 out of 56 holistic actions involve tools. In addition, tools tend to have more visually distinctive appearances compared to the objects in HA-ViD-VQA. On HA-ViD-VQA, verbs consistently outperform object recognition, suggesting that motion patterns provide stronger discriminative signals than (object) appearance for fine-grained assembly actions. Conversely, IKEA-ASM-VQA shows the opposite trend. This reversal likely reflects the different action granularities: HA-ViD captures micro-manipulations requiring precise motion understanding, while IKEA-ASM's furniture assembly involves macro-manipulations on larger, more visually distinct objects that are easier to identify. The gap between element-wise and holistic action accuracy further highlights the challenge of composing multiple element predictions into a coherent action prediction. While decomposition enables interpretable predictions and targeted improvements, the composition mechanism itself needs sophisticated aggregation beyond simple concatenation to handle element uncertainties effectively.

3) Qualitative Comparison: To better understand how CCFT improves assembly action understanding, we compare outputs from three configurations: our fine-tuned model, base Qwen2.5-VL, and base model with contextual prompting (providing lists of possible verbs, manipulated objects, target objects, and tools). Fig. 2 shows results for a video in HA-ViD-VQA with ground truth: *screw the hex screw into the screw hole C3 using the hex screwdriver*. Our fine-tuned model correctly identifies all elements with precise, deterministic outputs suitable for manufacturing applications. In contrast, the base model, despite correctly identifying the verb "*screw*," generates vague object descriptions ("*mechanical device*," "*small-scale machine*") and verbose explanations lacking assembly-specific details. Providing contextual prompts yields only marginal improvements—while identifying the tool correctly, the model still produces imprecise object descriptions and verbose outputs. This demonstrates that compositional fine-tuning is essential for learning assembly-specific visual-semantic mappings; simply constraining the output space through prompting is insufficient.

D. Ablation Studies

We conduct ablation studies to analyze the contribution of each key component in our method. All experiments use HA-ViD-VQA due to its comprehensive four-element annotations, using the same configurations in Section IV-B.

Compositional vs. Non-compositional Fine-tuning: We compare CCFT against a baseline that fine-tunes the same VLM on holistic action recognition with no decomposition. Table III compares holistic action recognition accuracy between compositional (CCFT) and non-compositional fine-tuning on HA-ViD-VQA. CCFT delivers consistent gains, with improvements of up to 4.77 percentage points and an average increase of 2.33 percentage points. We attribute these improvements to task decomposition, which reduces label ambiguity and focuses model attention on element-specific visual and semantic cues. Beyond higher accuracy, CCFT generates interpretable intermediate outputs for downstream applications, and its compositional structure permits targeted improvements through element-specific optimization.

Layer Partitioning vs. Shared Adapter: We compare the layer partition mechanism of LP-AT against a variant that uses a single LoRA adapter shared across all layers

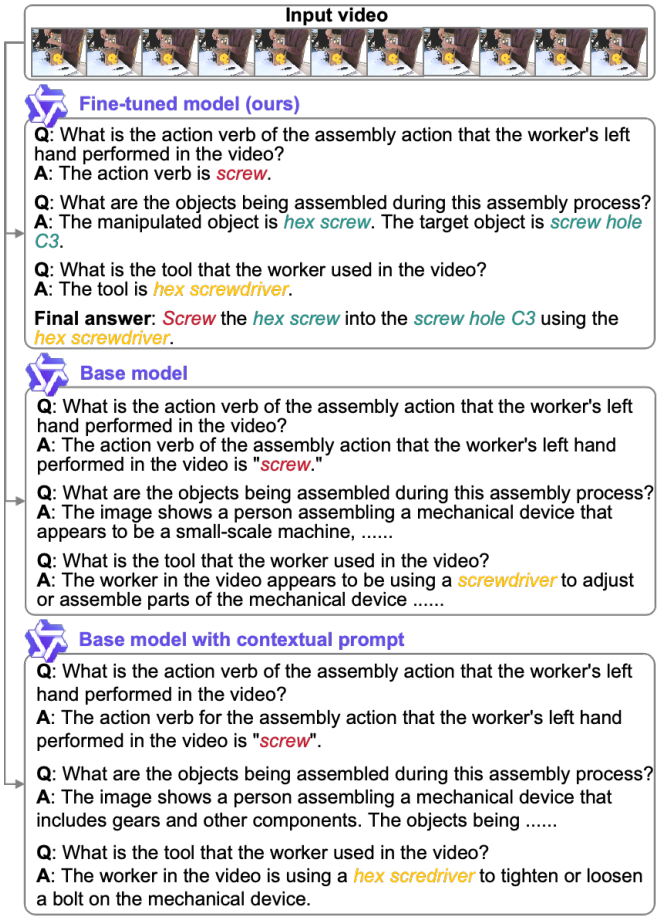

Fig. 2: Qualitative Comparison of Assembly Action Understanding Outputs from Our Fine-Tuned Model, the Base Model, and the Base Model with Contextual Prompt.

and subtasks (see Table IV). Compared to LP-AT (Table II), the shared adapter consistently yields lower performance, especially in object and holistic action recognition. This degradation can be attributed to the shared adapter's inability to model the distinct learning dynamics and feature requirements of each subtask, as all subtasks must adapt within a common parameter space. In contrast, our layer-partitioned strategy allows different network layers to specialize for individual subtasks via isolated adapters, improving learning efficiency and reducing representational conflicts. Moreover, since the shared adapter represents a conventional multi-task fine-tuning approach, the observed performance gap suggests that, under limited data, layer-wise specialization may be more effective in learning stable and deterministic representations for fine-grained action understanding.

Alternating vs. Independent vs. Sequential Training: We compare three training schemes: (i) *Alternating* (LP-AT): adapters of each subtask trained and merged iteratively within each epoch, enabling cross-task knowledge transfer; (ii) *Independent*: adapters of each subtask trained separately then merged, lacking inter-task learning; (iii) *Sequential*: adapters of each subtask trained one after another with cumulative merging. This comparison isolates the effect of interleaved versus decoupled subtask optimization.

TABLE III: Comparison of Compositional vs. Non-compositional Fine-tuning for Action Recognition on HA-ViD-VQA. Top-1 accuracy (%) for left-hand and right-hand action recognition is reported separately under side, front, and top views. Compositional approach improves performance up to 4.77% and average of 2.33%.

View	Method	Left-hand Acc. (%)	Right-hand Acc. (%)
Side	Compositional	46.27	47.62
	Non-compositional	44.78	47.62
Front	Compositional	49.25	42.86
	Non-compositional	46.27	38.10
Top	Compositional	53.73	38.10
	Non-compositional	53.73	33.33

TABLE IV: Element-wise Action Recognition Accuracy on HA-ViD-VQA using Shared Adapter. We report the recognition accuracy (%) for each action element: Verb, Manipulated Object (MO), Target Object (TO), Tool, and the holistic Action. LH and RH denote left-hand and right-hand actions, respectively. SV, FV, and TV refer to side, front, and top views. ΔLA-PT reflects the average gain by applying our method relative to a shared adapter approach, which is superior in object and overall action recognition by 2.68%-4.77% and 5.31% respectively.

	Verb v	MO o_m	TO o_t	Tool t	Action a
LH-SV	61.19	49.25	47.76	92.54	38.81
LH-FV	65.57	52.24	53.73	95.52	43.28
LH-TV	73.13	68.66	65.67	92.54	53.73
RH-SV	67.86	50.00	47.62	92.86	40.48
RH-FV	61.90	50.00	50.00	97.62	36.90
RH-TV	67.86	47.62	45.24	95.24	36.90
ΔLA-PT	-2.30	**4.77**	**2.68**	-0.42	**5.31**

Table V reveals catastrophic performance collapse when merging independently trained adapters. While individual adapters achieve competitive task-specific accuracy before merging, especially in the action elements, the merged model exhibits severe degradation, with an average 29.12% drop across all elements. This decline arises from parameter interference: each adapter modifies the shared parameter space without awareness of other tasks' requirements, creating conflicting gradients that, when combined, result in a model that fails at all tasks.

Table VI illustrates the catastrophic forgetting inherent in sequential training. In this scheme, adapters are trained consecutively (Verb → Manipulated and Target Objects → Tool), with each newly trained adapter merged into the model before proceeding to the next adapter training. We report two critical measurements: performance immediately after training each subtask-specific adapter (showing the adapter's initial performance on the specific subtask) versus performance of the final model after all subsequent training steps. The results reveal severe retroactive interference. While each adapter achieves competitive accuracy when first trained, performance degrades by an average of 46.04% as subsequent adapters are added. This progressive degradation demonstrates that each new adapter overwrites the representations learned by previous ones, as the model lacks mechanisms to preserve earlier knowledge.

This ablation study highlights the advantage of our alternating strategy: by iteratively training and merging adapters within each epoch, it learns mutually compatible representations that maintain performance for each subtask.

TABLE V: Element-wise accuracy (%) using independent training scheme on HA-ViD-VQA. Each cell shows performance before/after merging all the adapters.

	Verb v	MO o_m	TO o_t	Tool t
LH-SV	59.70/53.73	59.70/38.81	56.72/38.81	91.04/50.75
LH-FV	64.18/58.21	59.70/38.81	53.73/38.81	94.03/61.19
LH-TV	76.12/74.63	65.67/38.81	62.69/38.81	88.06/86.57
RH-SV	59.52/55.95	58.33/21.43	58.33/28.57	89.29/89.29
RH-FV	55.95/36.90	55.95/21.43	51.19/28.57	97.62/89.29
RH-TV	63.10/63.10	58.33/21.43	58.33/28.57	90.48/85.71
Average drop	10.23	49.85	40.61	15.78

TABLE VI: Element-wise accuracy (%) using sequential training scheme on HA-ViD-VQA. Each cell shows the performance after subtask-specific training/final model performance. The average drop indicates the relative performance decrease of the sequential training scheme.

	Verb v	MO o_m	TO o_t	Tool t
LH-SV	59.70/38.81	46.27/38.81	52.24/38.81	95.52
LH-FV	64.18/38.81	58.21/38.81	56.72/38.81	95.52
LH-TV	76.12/38.81	73.13/38.81	62.69/38.81	94.03
RH-SV	59.52/21.43	55.95/21.43	57.14/28.57	95.24
RH-FV	55.95/21.43	55.95/21.43	50.00/28.57	97.62
RH-TV	63.10/21.43	53.57/21.43	52.38/ 28.57	97.62
Average drop	52.55	46.63	38.95	

E. Discussion

While our experiments demonstrate the effectiveness of CCFT and LP-AT for assembly action understanding, we identify several limitations and future research directions. **1. Integration into Real-World HRC**: The current work is a foundational step in leveraging VLMs for assembly action understanding. A critical next step is integration into real-world HRC. Future work must focus on optimizing computational efficiency for real-time action recognition, enabling a robot to analyze human actions and provide timely assistance during cooperative assembly tasks.

2. Adaptive Hyperparameter Optimization: Although LP-AT supports subtask-specific hyperparameter configuration (e.g., LoRA rank r_i , scaling factor α_i and learning rate η_i), our implementation relied on empirically tuned settings. These hyperparameters may be suboptimal as different action elements exhibit varying learning dynamics throughout training. Future work should explore adaptive hyperparameter optimization strategies that could dynamically adjust hyperparameters for each subtask during training, ensuring stable and efficient learning.

3. Principled Layer Partitioning: On HA-ViD-VQA, tool recognition achieves strong performance with only four layers, whereas verb recognition uses ten layers yet yields lower accuracy. This discrepancy suggests that different subtasks may require distinct network depths and capacities. However, our layer partitioning strategy remains heuristic, while informed by task complexity estimates, lacking theoretical grounding. A promising direction is to conduct layer-wise relevance analysis or develop data-driven layer partitioning methods, informing more principled partitioning decisions.

4. Granularity of Action Decomposition: Our action decomposition stays at the semantic level. Assembly actions inherently involve kinematic constraints and exhibit complex motion patterns, therefore, finer decomposition (e.g. motion-level), deserves attention. Future research should explore multi-level hierarchical decomposition that extends beyond semantic level, enabling hierarchical reasoning and facilitating transfer learning to robotic manipulation tasks [31].

5. Dedicated Visual Representation Learning: Due to the small dataset scale, we froze the visual encoder during training, which is a pragmatic choice that avoids instability. However, as Table II indicates, generic visual features inadequately capture assembly-specific visual patterns or nuanced hand-object interactions. Future efforts should develop assembly-specific vision encoders that emphasize assembly attribute recognition and hand-object interactions.

6. Compositional Reasoning: Our pipeline currently forms the final action predictions by concatenating the predicted action elements. While deterministic, this makes predictions sensitive to single-element errors. Future work should explore robust composition mechanisms, such as lightweight rule-based logic or LLM-driven inference, to reconcile or correct inconsistent element predictions and infer the most plausible holistic action. LLM-based reasoning would also facilitate seamless adaptation of our compositional outputs to new application scenarios [32].

V. Conclusions

This paper addressed a critical gap in human-robot collaborative assembly: adapting VLMs for complex assembly action understanding. We presented CCFT with LP-AT, demonstrating that structured task decomposition, subtask-specific layer allocation, and alternating subtask-specific weight updates are key to successful VLM adaptation in this domain. CCFT stems from two key insights: (1) complex tasks become tractable when decomposed into simpler subtasks; (2) different subtasks demand different network capacities. LP-AT operationalizes these insights by alternately optimizing subtask-specific LoRA adapters, mitigating cross-subtask interference. Beyond quantitative improvements, our method provides interpretable, near-deterministic element-level outputs essential for manufacturing deployment.

To facilitate research in this area, we created the HA-ViD-VQA and IKEA-ASM-VQA datasets as benchmarks for VLM-based compositional assembly action understanding. Extensive experiments and ablations confirm the superiority of our method over state-of-the-art action recognition baselines, validating the benefits of compositional over non-compositional approach, layer partitioning over non-partitioned adapters, and alternating over non-alternating training strategies.

REFERENCES

[1] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5625–5644, 2024.

[2] Z. Li, X. Wu, H. Du, F. Liu, H. Nghiem, and G. Shi, "A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges," 2025. [Online]. Available: https://arxiv.org/abs/2501.02189

[3] D. Aganian, B. Stephan, M. Eisenbach, C. Stretz, and H.-M. Gross, "Attach dataset: Annotated two-handed assembly actions for human action understanding," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 367–11 373.

[4] H. Zheng, W. Xia, and X. Xu, "A human-robot collaborative assembly framework with quality checking based on real-time dual-hand action segmentation," *Robotics and Computer-Integrated Manufacturing*, vol. 94, p. 102976, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584525000304

[5] T. Nguyen, Y. Bin, J. Xiao, L. Qu, Y. Li, J. Z. Wu, C.-D. Nguyen, S.-K. Ng, and L. A. Tuan, "Video-language understanding: A survey from model architecture, model training, and data perspectives," 2025. [Online]. Available: https://arxiv.org/abs/2406.05615

[6] 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," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[7] H. Zheng, R. Lee, and Y. Lu, "Ha-vid: A human assembly video dataset for comprehensive assembly knowledge understanding," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 67 069–67 081. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/d40e6e4b3ee6c24f2bf2cb72c2412f4b-Paper-Datasets_and_Benchmarks.pdf

[8] Y. Ben-Shabat, X. Yu, F. Saleh, D. Campbell, C. Rodriguez-Opazo, H. Li, and S. Gould, "The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 846–858.

[9] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," 2025. [Online]. Available: https://arxiv.org/abs/2502.13923

[10] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2740–2755, 2019.

[11] J. Lin, C. Gan, K. Wang, and S. Han, "Tsm: Temporal shift module for efficient and scalable video understanding on edge devices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2760–2774, 2022.

[12] J. Lee, M. Lee, D. Lee, and S. Lee, " Hierarchically Decomposed Graph Convolutional Networks for Skeleton-Based Action Recognition ," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2023, pp. 10 410–10 419. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00958

[13] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, L. Wang, and Y. Qiao, "Uniformerv2: Unlocking the potential of image vits for video understanding," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1632–1643.

[14] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang, and Y. Qiao, "Videomae v2: Scaling video masked autoencoders with dual masking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 549–14 560.

[15] H. Zheng, R. Lee, Y. Lu, and X. Xu, "Duha: a dual-hand action segmentation method for human-robot collaborative assembly," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 522–527.

[16] Y. Zhang, K. Ding, J. Hui, S. Liu, W. Guo, and L. Wang, "Skeleton-rgb integrated highly similar human action prediction in human–robot collaborative assembly," *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102659, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584523001345

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[18] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 4904–4916. [Online]. Available: https://proceedings.mlr.press/v139/jia21b.html

[19] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li, "Llava-video: Video instruction tuning with synthetic data," 2025. [Online]. Available: https://arxiv.org/abs/2410.02713

[20] C. Wei and Z. Deng, "Incorporating scene graphs into pre-trained vision-language models for multimodal open-vocabulary action recognition," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 440–447.

[21] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023. [Online]. Available: https://arxiv.org/abs/2305.14314

[22] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, and M.-H. Chen, "Dora: Weight-decomposed low-rank adaptation," 2024. [Online]. Available: https://arxiv.org/abs/2402.09353

[23] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhania, R. Wang, and A. Yao, "Assembly101: A large-scale multi-view video dataset for understanding procedural activities," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 21 064–21 074.

[24] Z. Ma, J. Hong, M. O. Gul, M. Gandhi, I. Gao, and R. Krishna, " @ CREPE: Can Vision-Language Foundation Models Reason Compositionally? ," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2023, pp. 10 910–10 921. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01050

[25] C. Mitra, B. Huang, T. Darrell, and R. Herzig, " Compositional Chain-of-Thought Prompting for Large Multimodal Models ," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2024, pp. 14 420–14 431. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01367

[26] F. Parascandolo, N. Moratelli, E. Sangineto, L. Baraldi, and R. Cucchiara, "Causal graphical models for vision-language compositional understanding," 2025. [Online]. Available: https://arxiv.org/abs/2412.09353

[27] H. Zheng, R. Lee, H. Liang, Y. Lu, and X. Xu, "Ducas: a knowledge-enhanced dual-hand compositional action segmentation method for human-robot collaborative assembly," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7175–7180.

[28] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," 2025. [Online]. Available: https://arxiv.org/abs/2502.13923

[29] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," 2020. [Online]. Available: https://arxiv.org/abs/2002.10957

[30] A. W. Needham and E. L. Nelson, "How babies use their hands to learn about objects: Exploration, reach-to-grasp, manipulation, and tool use," vol. 14, no. 6, p. e1661. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/10.1002/wcs.1661

[31] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2021, p. 6541–6548. [Online]. Available: https://doi.org/10.1109/ICRA48506.2021.9561548

[32] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, "Self-refine: Iterative refinement with self-feedback," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 534–46 594, 2023.