Computers and Electronics in Agriculture 255 (2026) 112347

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Scene-graph-grounded vision–language reasoning to action for autonomous multipurpose agricultural robots☆

Yonghyun Park [a,d], Changjo Kim [b,c], Gangmin Kim [b,c], Hyoung Il Son [b,c,d,*]

[a] Physical-AI Operated Locomotion and Autonomous Robotics in Extreme Environments (POLAR-AI) Research Center, Gwangju Institute of Science and Technology (GIST), Cheomdangwagi-ro 123, Gwangju 61005, Republic of Korea
[b] Department of Convergence Biosystems Engineering, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea
[c] Interdisciplinary Program in IT-Bio Convergence System, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea
[d] Research Center for Biological Cybernetics, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea

ARTICLE INFO

Dataset link: https://github.com/parkoh25/SGG_VLM

Keywords:
Autonomous farming
Dual-arm agricultural robot
Grounded reasoning
Scene-graph generation
Language-based planning

ABSTRACT

Robotic manipulation in horticultural production remains challenging due to canopy occlusion, mechanically diverse foliage, and the need to prioritize agronomic tasks under uncertain visual evidence. This paper proposes a scene-graph-grounded vision–language reasoning-to-action framework that links structured perception to dual-arm access-oriented manipulation for autonomous multipurpose agricultural robots. Given an observation, a scene-graph generation (SGG) module extracts object instances, attributes, and relationships, which are transformed into a grounded FACT set used as the only visual evidence for planning. A constrained vision–language reasoning module generates a task-level plan and a dual-arm strategy while enforcing strict schema compliance, FACT consistency, and feasibility screening to prevent unsupported visual claims from reaching the robot controller. A minimal reasoning-to-action interface converts the verified plan into executable *approach* and *push* primitives for target access and cooperative occlusion management. Field validation is conducted on a dual-arm platform in a commercial greenhouse with $N = 50$ trials (11 harvesting, 14 leaf-removal/pruning, and 25 thinning; 14 cooperative trials). Under standard SGG evaluation, the perception stack achieves relationship recall@50/mean recall (mR)@50 of 63.5/36.2 (predicate classification), 44.1/27.5 (scene-graph classification), and 35.0/22.1 (scene-graph detection; SGDet), with attribute performance of recall@50/mR@50 = 64.0/47.0 (SGDet). End-to-end evaluation yields a plan-validity rate of 0.82 for PSR ∧ FnE ∧ CAS and a strict accepted-validity rate of 0.78 after excluding hallucination-flagged trials. The component-level scores are 0.90 for precondition satisfaction, 0.82 for feasibility/executability, 0.90 for correct action sequencing, and 0.90, 0.82, and 0.82 for task, cooperation, and target agreement, respectively, with a hallucination rate of 0.16. Physical execution achieves an action success rate of 0.60 and a successful-and-safe execution rate of 0.56; all invalid logical plans were blocked before execution (UEPR = 1.00). These results indicate that scene-graph-grounded vision–language reasoning provides an auditable bridge from structured perception to access-oriented multipurpose manipulation in occlusion-heavy environments, while execution robustness remains the primary bottleneck beyond plan-level validity.

1. Introduction

Agricultural production is increasingly adopting sensing, artificial intelligence, data analytics, and autonomous control to address persistent labor shortages and improve operational consistency (Ryan, 2023; Abbasi et al., 2022). Recent reviews identify intelligent perception, decision-making, dexterous manipulation, and reconfigurable control as core enabling technologies for agricultural robots (Liu et al., 2024; Hernández et al., 2025). However, greenhouse deployment remains difficult because biological variability, dense canopies, and task-dependent interactions limit the transfer of task-specific solutions across agricultural operations (Sánchez-Molina et al., 2024).

To operate robustly in such environments, agricultural robots must move beyond preprogrammed task execution toward scene-aware and

☆ This article is part of a Special issue entitled: 'Foundation Models in Agriculture' published in Computers and Electronics in Agriculture.

* Corresponding author at: Department of Convergence Biosystems Engineering, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea.
E-mail addresses: parkoh25@gist.ac.kr (Y. Park), ckddnckdwh12@jnu.ac.kr (C. Kim), gangmin0321@jnu.ac.kr (G. Kim), hison@jnu.ac.kr (H.I. Son).
URL: http://hralab.jnu.ac.kr (H.I. Son).

https://doi.org/10.1016/j.compag.2026.112347
Received 12 February 2026; Received in revised form 18 August 2026; Accepted 20 August 2026
Available online 31 August 2026

context-aware autonomy (Sánchez-Molina et al., 2024). When biological objects overlap or occlude one another, such as leaves covering a target fruit, conventional systems often struggle to adjust their manipulation strategies online (Park et al., 2025). A robot that detects objects as isolated instances may identify *what* is present and *where* it is located but still fail to determine *how* those objects should be manipulated in context. This limitation makes structured scene understanding and grounded reasoning central requirements for adaptive multipurpose agricultural manipulation.

The motivation of this study is therefore not to apply a language model to agriculture as a general trend, but to address a specific deployment gap in greenhouse manipulation. In dense tomato canopies, the same detected object can require different actions depending on its relation to the target: a leaf may be ignored, avoided, or actively pushed aside; a stem may be treated as a rigid obstacle; and an immature fruit may become the target only under a thinning task. Object-level perception alone cannot resolve these action-dependent meanings. At the same time, a fixed rule table can be too brittle when several plausible manipulation strategies exist, whereas unconstrained language-based planning can introduce unsupported visual claims. This combination of relational ambiguity, physical feasibility, and safety risk motivates a verified reasoning-to-action framework grounded in explicit scene evidence.

Recent advances in learning-based perception and three-dimensional sensing have improved target localization, pose estimation, and visual servoing for agricultural manipulation (Park et al., 2024, 2025). Nevertheless, object-level perception primarily explains *what* is present and *where* it is located, but it does not explicitly represent the relational and functional context needed to determine *how* the robot should interact with surrounding plant organs. Scene-level representations are therefore required to encode occlusion, attachment, proximity, and other task-dependent relations for action-oriented reasoning (Krishna et al., 2017; H. Li et al., 2024).

Language-based reasoning is useful in this setting only when it is strictly grounded in the current visual evidence. We do not use the language model as a general-purpose task planner that reasons freely over raw images or prior knowledge. Instead, the planner is restricted to a scene-specific FACT set derived from SGG outputs. This restriction is necessary because unconstrained language generation can produce unsupported object, attribute, or relation claims, which may lead to unsafe or physically infeasible actions in embodied systems. The role of the language model in this study is therefore limited and operational: it resolves task and action choices over explicit scene facts, while schema validation, FACT-consistency checking, and feasibility gating determine whether the generated plan can reach the robot.

Scene graphs provide a natural representation for this purpose because they encode objects, attributes, and relationships as structured visual evidence (Krishna et al., 2017; Zellers et al., 2018). More broadly, robotics studies have begun to ground language reasoning in structured scene representations to improve controllability and support multi-step planning under spatial constraints (H. Li et al., 2024). For greenhouse manipulation, this means that the planning module should reason not over raw pixels, but over machine-readable scene evidence that preserves occlusion, attachment, proximity, and other task-relevant relations.

Building on this line of research, prior work introduced a visual scene understanding and task-planning framework that applies scene-graph generation (SGG) to encode inter-object relationships and physical attributes for agricultural manipulation (Park and Son, 2025). That system used deterministic rules to differentiate among harvesting, leaf-removal/pruning, and thinning tasks and to coordinate dual-arm execution according to the inferred rigidity of the occluding objects. Although this rule-based framework demonstrated the value of relational awareness for task selection and cooperative manipulation, its reliance on fixed logic limited adaptability in unforeseen or ambiguous canopy configurations. The present study addresses that limitation by replacing deterministic rule mapping with grounded language-based reasoning while preserving explicit scene evidence and execution verification.

This distinction positions the present work: the goal is to preserve the inspectability of scene-graph-based planning while allowing more flexible reasoning over ambiguous canopy states. The core technical advance of this study is the verification-aware interface between structured perception and executable dual-arm action. A simple combination of existing components would pass SGG outputs to a VLM planner and then execute the generated action sequence. Such a pipeline remains fragile because errors can occur at all interfaces: SGG outputs may be incomplete or incorrect, a VLM planner may refer to objects or relations not present in the scene, and a robot controller may receive action arguments that are syntactically valid but physically infeasible. The proposed framework addresses these interface failures by treating the scene-graph output as a closed evidence set, deterministically serializing it into grounded FACTs with persistent object identifiers, constraining the planner to a predefined decision schema, checking every generated object, attribute, and relation against the FACT set, repairing invalid candidates using explicit violation messages, and passing only feasibility-screened primitives to the dual-arm controller. Therefore, the contribution is not a loose aggregation of SGG, VLM prompting, and robot control, but an auditable perception–reasoning–action contract that can accept, reject, or repair a plan before physical execution.

To prevent misunderstanding, the scope of the claims is stated explicitly. This study contributes four elements: (i) a reasoning-to-action formulation in which SGG output is used as a *closed* FACT evidence set for planning; (ii) an inference-time verification-and-repair layer that enforces schema validity, FACT consistency, and action feasibility before execution; (iii) a minimal, auditable dual-arm reasoning-to-action interface based on *approach* and *push* primitives; and (iv) closed-loop field validation with a validity-aware evaluation protocol. It does not claim a new SGG architecture or state-of-the-art relationship prediction, unconditional superiority over rule-based planners, an end-to-end learned VLA policy, or full crop-handling automation beyond access-oriented primitives. The empirical contribution is conditional: given imperfect relational perception, the verification layer keeps language-based planning auditable and blocks unsupported or infeasible plans before execution.

Despite the promise of language-based reasoning, a substantial gap remains in ensuring that high-level plans can be translated into safe and reliable action sequences on physical platforms, particularly under severe occlusion in dense greenhouse canopies (Park et al., 2023). Flexible reasoning alone is insufficient if the planner ignores kinematic limits, reachability, collision risk, or the immediate scene context. Likewise, without explicit grounding, VLM planners may generate unsupported visual assumptions that compromise both decision validity and execution safety. Moving from proof-of-concept reasoning to practical greenhouse deployment therefore requires field validation of an end-to-end process that links structured semantic inference to physically executable interaction (Seol et al., 2024).

To address these deployment-driven requirements, this study proposes a FACT-grounded reasoning-to-action framework for autonomous multipurpose agricultural robots. The framework is designed around a conservative principle: a robot action should be generated only from explicit scene evidence and should be rejected when the evidence or feasibility constraints are insufficient. The proposed approach therefore transitions from raw visual input to embodied physical interaction through three stages. First, SGG decomposes the canopy into structured object–relation–attribute evidence and serializes it into grounded FACTs. Second, the VLM reasoning module selects the task, target, cooperation mode, and action sequence under schema and FACT-consistency constraints. Third, the reasoning-to-action interface maps only verified symbolic plans to executable *approach* and *push* primitives for dual-arm manipulation. Accordingly, the novelty lies not

in adopting a language model for agricultural robotics, but in formalizing and field-validating an auditable interface between imperfect relational perception, constrained symbolic reasoning, and executable dual-arm action.

Importantly, the study is evaluated as a closed-loop robotic system rather than as an offline data-processing pipeline. Each field trial begins with a greenhouse observation, passes through SGG-based FACT extraction, language-based reasoning, validation, and reasoning-to-action conversion, and terminates with physical dual-arm execution or an explicit safety/feasibility abort. This design allows the evaluation to separate three levels of performance: relational perception quality, plan-level validity, and physical execution success.

The principal contributions of this study are as follows:

- This study proposes a FACT-grounded reasoning-to-action framework for resolving action choices under relational ambiguity in dense greenhouse canopies, treating SGG output as a closed evidence set for planning (Sections 3.1 and 3.2).
- This study introduces a verification-and-repair layer that enforces schema validity, FACT consistency, and action feasibility before any generated plan reaches the robot controller (Section 3.3).
- This study defines a minimal dual-arm action interface that converts verified symbolic decisions into executable *approach* and *push* primitives for cooperative occlusion management (Section 3.4).
- This study provides field validation in a commercial greenhouse against a rule-based reference planner, focusing on logical validity, decision agreement, hallucination control, and execution outcomes, together with a relation-error attribution analysis that links SGG relation errors to cooperation-decision errors (Sections 4 and 4.7).

2. Related work

This section positions the proposed framework relative to prior studies in agricultural robot perception, scene-graph generation, and language-based robot planning.

2.1. Artificial intelligence and robotics in agriculture

Agricultural robotics increasingly combines electronic sensing, artificial-intelligence-based perception, data-driven decision support, communication architectures, and automated control (Abbasi et al., 2022; Liu et al., 2024). Recent research also emphasizes reconfigurable robotic architectures and generative artificial intelligence as potential routes toward more adaptable agricultural automation (Hernández et al., 2025; Pallottino et al., 2025).

Greenhouse robotics remains a particularly demanding application because robots must operate around deformable crops, dense occlusions, narrow workspaces, and task-dependent plant interactions (Sánchez-Molina et al., 2024; Park et al., 2023). Although task-specific systems have advanced rapidly, field-validated frameworks that jointly connect relational perception, adaptive task reasoning, and executable manipulation remain limited.

The scope of the claims is stated explicitly in the Introduction: the framework targets the interface-level problem of converting imperfect relational perception into auditable symbolic evidence and executable dual-arm action, rather than competing with recent SGG architectures or end-to-end VLA policies on their own benchmarks. The detailed positioning relative to recent approaches is given in Section 2.5.

2.2. Perception and scene understanding for agricultural manipulation

Robust perception is essential for manipulating living plants because the robot must localize targets and relevant plant structures despite illumination variation, motion, and severe occlusion (Park et al., 2024, 2025). RGB-D sensing and geometry-aware multimodal representations further support metric localization and structured scene interpretation. REL-SF4PASS, for example, demonstrates how an explicit depth representation and multimodal fusion can improve geometry-aware scene understanding (Li et al., 2026).

Nevertheless, object labels and poses alone do not explain whether a plant organ should be ignored, avoided, pushed, or selected as a task target. Agricultural manipulation therefore requires relational representations that preserve occlusion, attachment, and proximity information for downstream reasoning and action selection.

2.3. Relationship modeling and scene-graph generation

Scene graph generation represents an image as a structured graph of localized objects, attributes, and pairwise predicates (Krishna et al., 2017). Global-context methods such as Neural Motifs model contextual dependencies among object and relation labels, while recent surveys summarize advances in relation prediction and structured visual reasoning (Zellers et al., 2018; H. Li et al., 2024). These representations provide a machine-readable basis for describing occlusion, attachment, proximity, and other relations relevant to robot decision-making.

Recent SGG research has addressed predicate bias and long-tailed relation distributions through unbiased and debiased learning (Tang et al., 2020; Zhou et al., 2023). Causal relation modeling further seeks to separate task-relevant relational cues from spurious contextual correlations (Zhou et al., 2025). Relationship-incremental and multilabel SGG support newly introduced predicates and multiple simultaneous relations between object pairs (Li et al., 2025; X. Li et al., 2024). Uncertainty-aware and modality-aligned methods address ambiguous relation evidence and cross-modal representation quality (Li et al., 2023; Yang et al., 2026).

These studies primarily improve the SGG model itself. The present work is complementary: Neural Motifs is used as the relation predictor, while the technical contribution lies in the downstream interface. Predicted objects, attributes, and relations are serialized into a closed FACT set, and every VLM-generated plan is checked against this evidence before dual-arm execution. Thus, the contribution is a verified SGG-to-action framework rather than a new generic SGG backbone.

2.4. Vision–language models and grounded cognitive reasoning

Grounded language-based robot planning and vision–language-action models demonstrate that language and visual representations can support high-level task decomposition and action selection (Brohan et al., 2023b,a). In agriculture, generative AI is also being investigated for knowledge services, decision support, and data interpretation, although reliable grounding remains an open requirement (Pallottino et al., 2025).

Ungrounded neural generation can produce objects, relations, or plans that are inconsistent with the observed scene (Ji et al., 2023). For agricultural manipulation, language-based reasoning must therefore remain linked to vision-derived structured evidence so that unsupported or infeasible actions can be rejected before physical execution.

2.5. Positioning relative to recent SOTA approaches

The proposed framework is related to, but different from, three recent research directions. First, recent SGG methods improve relation prediction, uncertainty modeling, predicate debiasing, and multimodal fusion (Li et al., 2023; Zhou et al., 2025). These methods improve the graph itself but do not specify how predicted relations should be

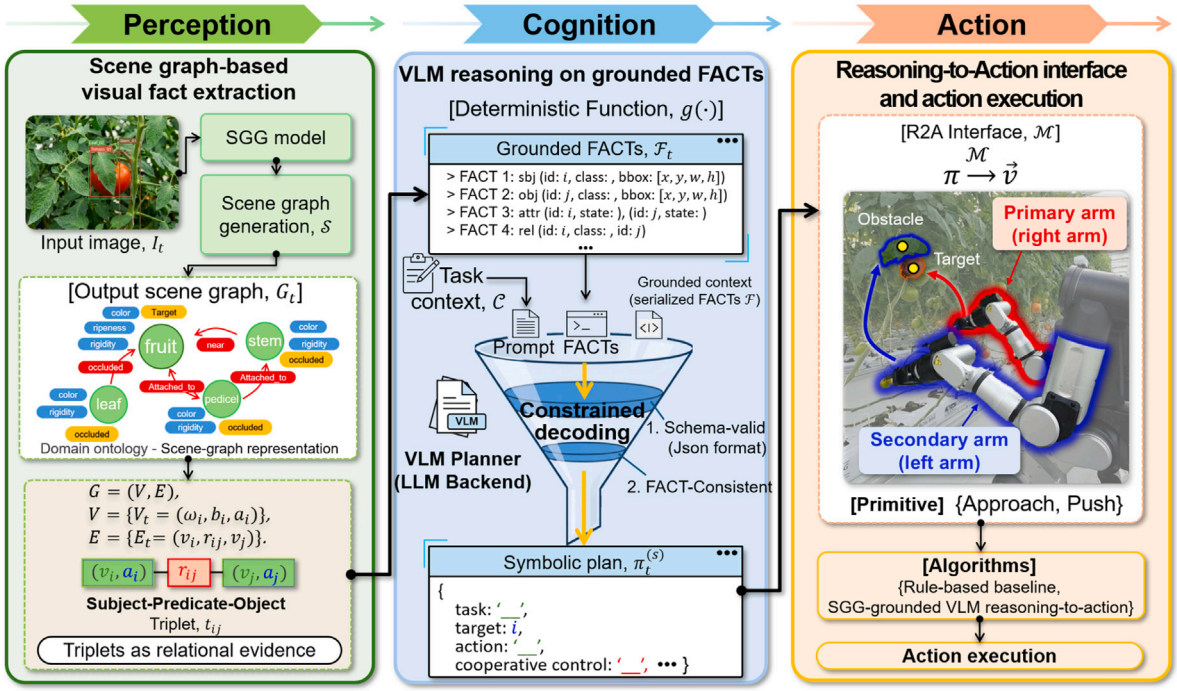

Fig. 1. Overall framework of scene-graph-grounded vision–language reasoning to action for autonomous multi-purpose agricultural robots. "Context" denotes the grounded context of the reasoning module: the serialized FACT set $\mathcal{F}$ (object identifiers, attributes, and relations) derived deterministically from the scene graph and injected into the VLM prompt as the only visual evidence.

verified and converted into robot actions. Second, grounded language-based planning constrains high-level language proposals using available robot skills or affordances (Brohan et al., 2023b). Third, end-to-end VLA policies learn direct mappings from visual observations and instructions to robot actions using large-scale robot data (Brohan et al., 2023a). In contrast, the present study uses a modular and inspectable FACT-based pipeline designed for limited-data greenhouse deployment.

2.6. Task planning and decision-making in agricultural robotics

Agricultural task planning must jointly consider agronomic intent, relational scene evidence, reachability, and safe interaction with surrounding plant structures (Park and Son, 2025). For occlusion-heavy manipulation, dual-arm systems can assign one arm to environmental management while the other performs the primary operation, but their practical use requires coordinated sensing, task allocation, and execution control (Jo et al., 2025). Existing agricultural systems still lack a field-validated mechanism that consistently links relational scene evidence, grounded high-level reasoning, and executable dual-arm action sequences in dense greenhouse environments.

3. Materials and methods

3.1. Overall system architecture and notation

The proposed framework separates vision-based evidence from language reasoning and maps verified plans to executable access and occlusion-management primitives (Fig. 1). The framework follows a perception–cognition–action pipeline, where scene graphs are extracted from visual observations and deterministically converted into grounded FACTs. A scene-graph-grounded VLM reasoning module generates symbolic plans under schema and FACT consistency constraints, which are then translated into executable dual-arm primitives via a reasoning-to-action interface. The system follows a three-stage architecture that transforms a raw observation into an executable dual-arm action sequence: (i) *structured visual evidence extraction* via SGG, (ii) *grounded FACT construction* as a deterministic transformation of scene-graph output, and (iii) a *VLM reasoning-to-action* framework that generates a task-level plan conditioned on grounded FACTs and converts it into action primitives for execution. This design separates "vision" (structure-grounded evidence) from "language" (reasoning and planning over evidence) to avoid mixing pixel-level uncertainty with symbolic decision logic.

The robot observation at decision step t is defined as I_t (RGB-D). The SGG module maps the observation to a directed scene graph as follows:

$$\mathcal{S} : I_t \mapsto G_t = (V_t, E_t), \tag{1}$$

where each node $v_i \in V_t$ represents an entity and each directed edge $e_{ij} \in E_t$ indicates a predicate between two entities, which is consistent with standard scene-graph formulations (Krishna et al., 2017; Zellers et al., 2018). Each entity node is defined as

$$v_i \triangleq (\omega_i, b_i, \mathbf{a}_i), \tag{2}$$

where $\omega_i \in \Omega$ indicates an object class label (e.g., *fruit*, *leaf*, *stem*, and *pedicel*), $b_i \in \mathbb{R}^4$ denotes a 2D bounding box, and $\mathbf{a}_i$ represents an attribute indicator vector over a predefined attribute vocabulary $\mathcal{A}$ (e.g., *ripe/unripe* and *rigid/flexible*). Each relationship edge is defined as follows:

$$e_{ij} \triangleq (v_i, r_{ij}, v_j), \quad r_{ij} \in \mathcal{R}, \tag{3}$$

with a task-relevant predicate set $\mathcal{R}$ (e.g., *occluded*, *attached-to*, and *near*). For compactness, a relational triplet is denoted as

$$t_{ij} = \langle (v_i, \mathbf{a}_i), r_{ij}, (v_j, \mathbf{a}_j) \rangle. \tag{4}$$

The reasoning module must not infer new visual statements beyond SGG outputs to enforce strict grounding. Instead, the grounded FACT set is constructed using a deterministic transformation of the triplet set:

$$\mathcal{F}_t = g(\Gamma_t), \quad \Gamma_t = \{t_{ij} \mid e_{ij} \in E_t\}, \tag{5}$$

Table 1
Dataset statistics and fixed SGG-module configuration for the tomato-greenhouse domain.

Item	Value
SGG framework	SGG-Benchmark
Relation predictor	Neural Motifs
Detector backbone	ResNet-50-FPN (R-50-FPN)
Image acquisition/network resize	1280 × 720/shorter side 600 pixels, maximum side 1000 pixels
Detection threshold τ_{det}	0.05
Class-wise NMS IoU	0.50
Relation filtering	Constrained ranking; top 100 relation triplets per image
Ripeness mapping	red → ripe; green or yellow → unripe
Rigidity mapping	leaf → flexible; stem, pedicel, or fruit → rigid/non-pushable
Annotated scenes (train/validation/test)	200 (120/40/40)
Object instances	1201 (fruit: 342; leaf: 576; stem: 148; pedicel: 135)
Relation triplets	1520 (occluded: 574; attached_to: 124; near: 679; in_front_of: 143)
Attribute vocabulary	7 labels (ripe, unripe, rigid, flexible, red, green, yellow)
Optimizer/learning rate	SGD/0.015 (momentum: 0.9; weight decay: 1×10^{-4})
Batch size/training schedule	16/40,000 iterations (learning-rate steps at 20,000 and 30,000 iterations)

where $g(\cdot)$ converts node existence, node attributes, and edges into symbolic facts (e.g., $\text{obj}(v_i, \omega_i)$, $\text{attr}(v_i, a)$, and $\text{rel}(v_i, r_{ij}, v_j)$). Therefore, this paper defines **vision** as $\mathcal{F}_t$ (objects, relationships, or attributes grounded by SGG), not raw pixels.

Given grounded FACTs, the VLM reasoning module (implemented using an LLM backend) makes high-level decisions using the FACT set and the task context C:

$$\mathcal{L} : (\mathcal{F}_t, C) \mapsto \pi_t^{(s)}, \tag{6}$$

where $\pi_t^{(s)}$ denotes a symbolic plan specifying the task type τ_t (e.g., *harvest*, *prune*, and *thin*), target entity $v_t^\star$, cooperative-control flag $\kappa_t \in \{0, 1\}$, and an ordered list of symbolic actions, consistent with recent studies that use LLMs for structured planning over grounded representations (Brohan et al., 2023b).

Finally, the reasoning-to-action interface converts $\pi_t^{(s)}$ into an executable primitive sequence for the dual-arm platform:

$$\mathcal{M} : (\pi_t^{(s)}, \mathcal{F}_t) \mapsto \pi_t^{(a)} = \left[(\xi_m, \theta_m)\right]_{m=1}^{M}, \tag{7}$$

where ξ_m denotes an action primitive (e.g., *approach* and *push*), and θ_m represents its parameter set (e.g., acting arm, target point, and displacement). Then, the resulting $\pi_t^{(a)}$ is passed to the low-level controller for *action execution*. **Language** is denoted by the inference and planning process $\mathcal{L}$ operating strictly on $\mathcal{F}_t$, and the **reasoning-to-action** interface is denoted by the conversion $\mathcal{M}$ that produces robot-executable actions.

3.2. Scene graph-based visual fact extraction

In this study, *vision* is defined as the grounded FACT set $\mathcal{F}_t$ (i.e., objects, attributes, and relationships derived from SGG), rather than raw pixel observations. This design follows the standard definition of a visually grounded scene graph, localizing object instances and modeling relationships between pairs of localized entities (Krishna et al., 2017).

The SGG module was implemented by adapting the public SGG-Benchmark framework with Neural Motifs as the relation prediction model (Zellers et al., 2018). Neural Motifs was used because it predicts pairwise visual relations from contextualized object features and is compatible with the standard SGG evaluation settings used in this study. The generic SGG vocabulary was replaced with a tomato-greenhouse ontology defined for robotic manipulation.

Table 1 reports the fixed dataset, training, and inference configuration used for the SGG module. The RGB-D acquisition resolution is distinguished from the detector-side image resizing, and all listed parameters were held fixed across the training, validation, test, and field-evaluation stages.

The associated scene-graph data were self-defined and annotated for the greenhouse domain. Each annotated scene contains object instances, object categories, task-relevant attributes, and pairwise relations. The object vocabulary includes *fruit*, *leaf*, *stem*, and *pedicel*. The relation vocabulary includes *occluded*, *attached_to*, *near*, and *in_front_of*. The annotated data were split at the scene level into training, validation, and test subsets to avoid leakage between visually similar greenhouse frames.

3.2.1. Scene-graph representation and greenhouse ontology

Given an RGB image I, the SGG module generates a directed graph $G = (V, E)$, in which each node $v_i \in V$ corresponds to an object instance grounded to an image region, and each directed edge $e_{ij} \in E$ represents a semantic relationship from a subject v_i to an object v_j. Following the standard SGG formulation, a bounding box $b_i \in \mathbb{R}^4$ and object category label $\omega_i \in \Omega$ represent each object instance, and a predicate label $r_{ij} \in \mathcal{R}$ associated with an ordered pair (v_i, v_j) represents each relationship (H. Li et al., 2024). In addition, each node may have a (possibly null) set of attributes $a_i \subseteq \mathcal{A}$ (e.g., state or physical/functional properties), consistent with datasets and benchmarks that annotate objects, attributes, and relationships (Krishna et al., 2017).

A task-relevant ontology is defined to ensure that downstream reasoning is aligned with the greenhouse robotic manipulation context, and it is used consistently throughout this paper (Table 2). The ontology comprises (i) object categories Ω (e.g., `fruit`, `leaf`, and `stem`), (ii) attribute vocabulary $\mathcal{A}$ (e.g., `ripe`, `unripe`, `rigid`, and `flexible`), and (iii) relationship predicates $\mathcal{R}$ (e.g., `occluded`, `attached_to`, and `near`). This ontology limits which visual concepts are represented as machine-readable evidence for planning without increasing inference beyond the SGG output.

3.2.2. Object detection over the greenhouse vocabulary

The first stage of the SGG cascade localizes and classifies plant organs over the four-class greenhouse vocabulary $\Omega = \{\texttt{fruit}, \texttt{leaf}, \texttt{stem}, \texttt{pedicel}\}$. A two-stage detector of the Faster R-CNN family is used, following the detector configuration of the SGG-Benchmark implementation of Neural Motifs (Zellers et al., 2018). A convolutional backbone with a feature pyramid maps the RGB observation I to feature maps $\Phi(I)$, and a region proposal network (RPN) generates class-agnostic candidate boxes $\{\hat{b}_p\}$. For each proposal, RoIAlign pooling extracts a fixed-size feature vector

$$\mathbf{f}_p = \text{RoIAlign}\big(\Phi(I), \hat{b}_p\big) \in \mathbb{R}^d, \tag{8}$$

which is passed to a classification head and a class-wise box-refinement head:

$$P(\omega \mid \mathbf{f}_p) = \text{softmax}\big(\mathbf{W}_{\text{cls}}\mathbf{f}_p + \mathbf{b}_{\text{cls}}\big), \qquad b_p = \hat{b}_p + \Delta_{\text{reg}}\big(\mathbf{f}_p, \omega_p\big), \tag{9}$$

where $\omega_p = \arg\max_{\omega \in \Omega \cup \{\text{bg}\}} P(\omega \mid \mathbf{f}_p)$ and Δ_{reg} denotes the predicted box offset for the selected class. The detector is trained with the standard multi-task objective

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{cls}} + \lambda_{\text{reg}}\,\mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{rpn}}, \tag{10}$$

Table 2
Task-relevant ontology for robotic manipulation in greenhouses.

Category	Vocabulary
Objects Ω	`fruit`, `leaf`, `stem`, `pedicel`
Attributes $\mathcal{A}$	ripeness: {`ripe`, `unripe`}, rigidity: {`rigid`, `flexible`}, color: {`red`, `green`, `yellow`}
Relations $\mathcal{R}$	`occluded`, `attached_to`, `near`, `in_front_of`
Tasks $\mathcal{T}$	Harvest, Prune, Thin
Primitives Ξ	{Approach, Push} (primitive instance: $\xi \in \Xi$)

where $\mathcal{L}_{\text{cls}}$ is the cross-entropy classification loss, $\mathcal{L}_{\text{reg}}$ is the smooth-L_1 box-regression loss, and $\mathcal{L}_{\text{rpn}}$ combines the proposal objectness and regression terms. At inference, detections with class confidence below a threshold τ_{det} are discarded, and class-wise non-maximum suppression (NMS) is applied to the remainder. The retained set $\{(\omega_i, b_i, s_i^{\text{obj}})\}_{i=1}^{n}$, with $s_i^{\text{obj}} = \max_{\omega \in \Omega} P(\omega \mid \mathbf{f}_i)$, defines the node candidates of the scene graph. Two domain properties motivated the configuration choices summarized in Table 1. First, `stem` and `pedicel` instances are thin and strongly elongated, so the anchor aspect ratios and the RoI pooling resolution must preserve narrow structures. Second, greenhouse canopies exhibit heavy instance overlap, so the class-wise NMS threshold balances duplicate suppression against the retention of genuinely overlapping instances. The fixed detection and NMS settings listed in Table 1 were used throughout all data splits and field trials; no scene-specific or trial-specific threshold adjustment was performed.

3.2.3. Relationship prediction with contextualized object features

Pairwise relations over the detected nodes are predicted with the Neural Motifs architecture (Zellers et al., 2018), which conditions each predicate decision on the global object context of the scene rather than on isolated object pairs. Object context is computed by a bidirectional LSTM (biLSTM) over the detected instances:

$$\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_n] = \text{biLSTM}\big([\mathbf{f}_i;\ \mathbf{W}_1 \hat{\mathbf{p}}_i]_{i=1,\dots,n}\big), \tag{11}$$

where $\mathbf{f}_i$ is the RoI feature of node v_i, $\hat{\mathbf{p}}_i$ is the detector class distribution over $\Omega \cup \{\text{bg}\}$, and $[\cdot\,;\,\cdot]$ denotes concatenation. A decoding LSTM predicts the final object labels sequentially from the contextualized states, $\omega_i = \arg\max \text{softmax}(\mathbf{W}_o\, \mathbf{h}_i)$, so that each label decision is informed by the labels of the surrounding canopy structure. A second biLSTM then produces edge-level context:

$$\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_n] = \text{biLSTM}\big([\mathbf{c}_i;\ \mathbf{W}_2\, \text{emb}(\omega_i)]_{i=1,\dots,n}\big), \tag{12}$$

where $\text{emb}(\cdot)$ is a learned class embedding. For an ordered pair (v_i, v_j), the pair representation combines head and tail projections of the edge context with the visual feature of the union box $b_{i \cup j}$, and the predicate posterior adds a class-conditioned frequency bias:

$$\mathbf{g}_{ij} = \big(\mathbf{W}_h \mathbf{d}_i\big) \circ \big(\mathbf{W}_t \mathbf{d}_j\big) \circ \mathbf{u}_{ij}, \qquad \mathbf{u}_{ij} = \text{RoIAlign}\big(\Phi(I),\ b_{i \cup j}\big), \tag{13}$$

$$P\big(r \mid v_i, v_j\big) = \text{softmax}\big(\mathbf{W}_r\, \mathbf{g}_{ij} + \beta_{\omega_i, \omega_j}\big)_r, \qquad r \in \mathcal{R} \cup \{\emptyset\}, \tag{14}$$

where $\circ$ denotes element-wise multiplication, $\emptyset$ denotes the no-relation class, and $\beta_{\omega_i, \omega_j} \in \mathbb{R}^{|\mathcal{R}|+1}$ denotes the class-conditioned frequency-bias term used by Neural Motifs. This term is retained as part of the standard Neural Motifs relation head and is not treated as a separate methodological contribution of this study. The relation head is trained with cross-entropy over the annotated predicates, and the triplet score used for recall-based evaluation combines the object-detection and predicate confidences:

$$s_{ij} = s_i^{\text{obj}} \cdot s_j^{\text{obj}} \cdot \max_{r \in \mathcal{R}} P\big(r \mid v_i, v_j\big). \tag{15}$$

The three standard evaluation settings in Section 4.2 correspond directly to stages of this cascade: predicate classification (PredCls) provides ground-truth boxes and labels and evaluates Eqs. (13)–(14) in isolation; scene-graph classification (SGCls) provides ground-truth boxes but requires label decoding through Eq. (11); and scene-graph detection (SGDet) runs the full cascade from Eq. (8) onward. Because downstream planning consumes directed relations, the directionality convention of every predicate is fixed identically across annotation, prediction, and FACT serialization. All predicates are subject-centric: $\langle v_i, \texttt{occluded}, v_j \rangle$ states that the subject v_i is occluded by the object v_j (i.e., v_j is the occluder), which matches the occlusion test in Algorithm 1; $\langle v_i, \texttt{attached\_to}, v_j \rangle$ states that v_i is physically attached to v_j (e.g., `fruit` attached to `pedicel` and `pedicel` attached to `stem`); $\langle v_i, \texttt{near}, v_j \rangle$ states that v_i lies within a local neighborhood of v_j; and $\langle v_i, \texttt{in\_front\_of}, v_j \rangle$ states that v_i is closer to the camera than v_j along the viewing axis. Fixing these conventions removes a potential ambiguity between "occludes" and "occluded" when relations are serialized into FACTs.

3.2.4. Attribute classification for planning-relevant properties

We distinguish relation prediction from attribute handling. Neural Motifs was used for pairwise relation prediction. Visual attributes used for task selection, such as ripeness and color, were estimated from the perception output. In contrast, planning-oriented attributes such as rigidity were not learned as continuous physical properties. They were assigned deterministically from object category as conservative action-planning tags; for example, leaves were treated as flexible, whereas stems and pedicels were treated as rigid or non-pushable. This distinction prevents the reasoning module from interpreting rigidity as an independently measured physical stiffness value.

The appearance-derived attribute used for task selection is represented as a categorical fruit-color token, $\text{col}(v_i) \in \{\texttt{red}, \texttt{green}, \texttt{yellow}\}$. The implementation does not estimate a continuous ripeness value. Instead, ripeness is assigned through the categorical rule

$$\text{ripe}(v_i) = \begin{cases} 1, & \text{col}(v_i) = \texttt{red}, \\ 0, & \text{col}(v_i) \in \{\texttt{green}, \texttt{yellow}\}. \end{cases} \tag{16}$$

Accordingly, red fruits are treated as `ripe`, whereas green and yellow fruits are treated as `unripe`. This attribute is used as a categorical task cue rather than as a continuous maturity estimate.

In contrast, the rigidity attribute is a category-derived, task-oriented planning tag rather than a measured physical stiffness. It is assigned by the deterministic mapping

$$\mu(\omega_i) = \begin{cases} \texttt{flexible}, & \omega_i = \texttt{leaf}, \\ \texttt{rigid}, & \omega_i \in \{\texttt{stem}, \texttt{pedicel}, \texttt{fruit}\}, \end{cases} \tag{17}$$

so that only leaves are considered pushable occluders, whereas stems, pedicels, and fruits are conservatively treated as rigid or non-pushable to avoid crop damage. This deterministic assignment ensures conservative and reproducible cooperation decisions, and it prevents the reasoning module from interpreting rigidity as an independently measured material property.

Attribute prediction is evaluated with attribute recall (Attr R@50 and mR@50) computed on the union of ripeness and rigidity labels against ground-truth annotations, where a prediction is counted as correct only if the object category is correctly detected and the associated attribute token matches the reference label. Because rigidity is derived deterministically from the predicted category through Eq. (17), the attribute scores primarily reflect the correctness of appearance-derived ripeness labels, while rigidity functions as a safety-oriented planning tag. This caveat is stated explicitly so that attribute performance is interpreted in terms of discrete decision cues for task planning rather than precise physical measurements.

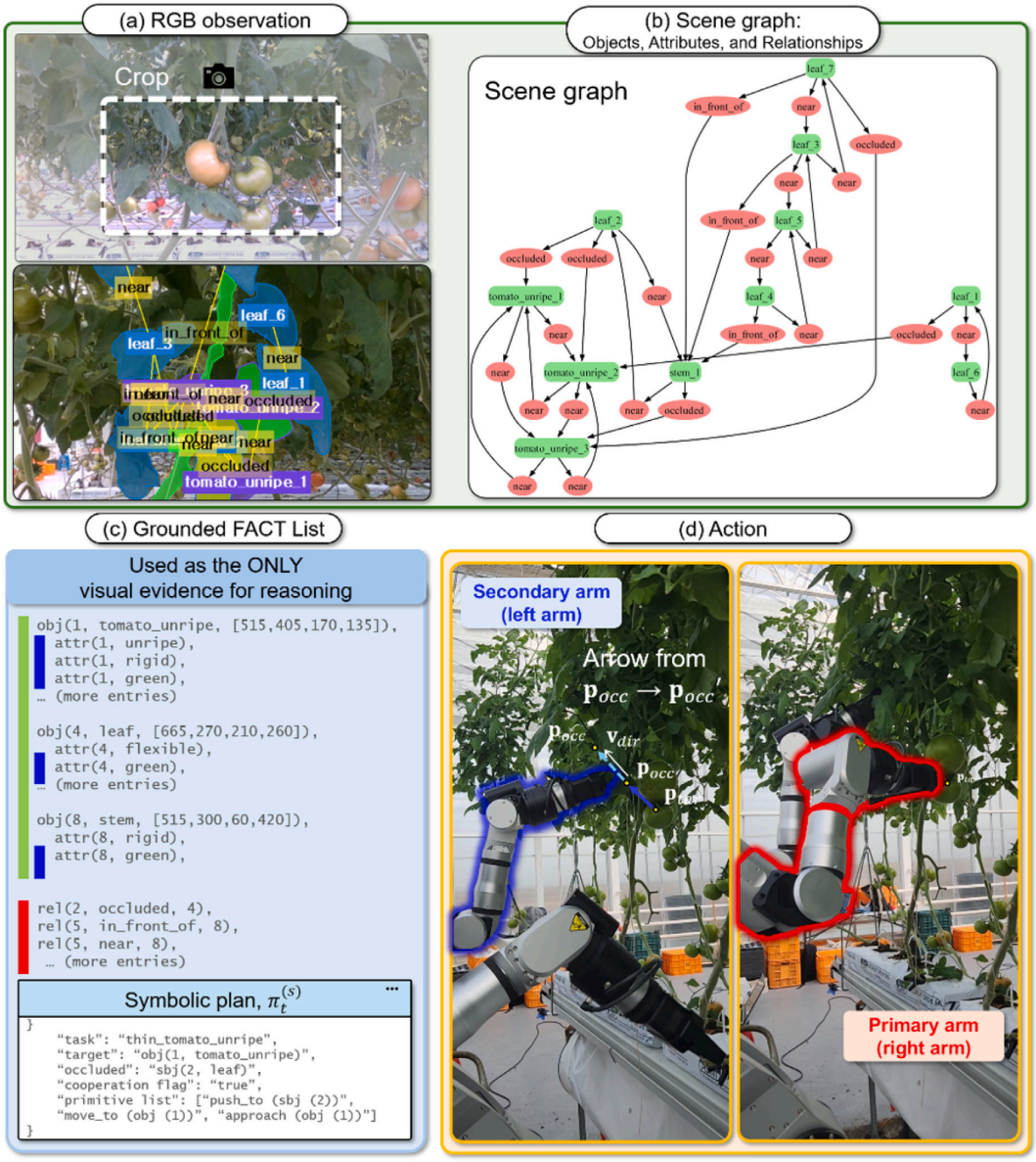

Fig. 2. FACT-based reasoning-to-action pipeline: (a) RGB observation; (b) scene graph with objects, relationships, and attributes; (c) the deterministically constructed grounded FACT list used as the only visual evidence for reasoning; and (d) action execution with dual-arm control.

3.2.5. *From triplets to a serialized grounded FACT set*

A set of grounded relational triplets is constructed from G:

$$t_{ij} = \langle (v_i, a_i),\ r_{ij},\ (v_j, a_j) \rangle, \quad \forall e_{ij} \in E, \tag{18}$$

where each triplet preserves (i) the identity and localization of the participating objects using their node indices and boxes, and (ii) the semantic predicate as an explicit directed relationship. The resulting triplet set is denoted as $\Gamma = \{t_{ij} \mid e_{ij} \in E\}$. Fig. 2 illustrates an example of scene-graph triplets and the corresponding grounded FACT serialization.

To prevent ungrounded assumptions at the reasoning stage, Γ is converted into a set of symbolic *grounded FACTs* via a deterministic transformation $g(\cdot)$:

$$\mathcal{F} = g(\Gamma), \qquad g : \Gamma \rightarrow 2^{\mathfrak{F}}, \tag{19}$$

where $\mathfrak{F}$ denotes the universe of fact templates. This approach creates object facts $f_i^{\text{obj}} = \texttt{obj}(i, \omega_i, b_i)$, attribute facts $f_{ik}^{\text{attr}} = \texttt{attr}(i, a_{ik})$ for $a_{ik} \in a_i$, and relationship facts $f_{ij}^{\text{rel}} = \texttt{rel}(i, r_{ij}, j)$. Notably, $g(\cdot)$ represents a *deterministic transformation*; it does not hypothesize missing entities or relationships but only re-encodes the SGG output into an established, machine-readable form tied to visual grounding (e.g., bounding boxes) (Krishna et al., 2017).

Before serialization, both nodes and edges pass explicit admission gates so that only sufficiently supported predictions become FACTs. A detected node is admitted if $s_i^{\text{obj}} \geq \tau_{\text{det}}$. Relations are admitted under the constrained ranking used at inference: each ordered pair contributes at

most one predicate,

$$r_{ij} = \arg\max_{r \in \mathcal{R}} P\left(r \mid v_i, v_j\right), \tag{20}$$

candidate triplets are ranked by the score s_{ij} of Eq. (15), and only the top-K_{rel} triplets per image enter the FACT set (K_{rel}=100; Table 1). The detection threshold τ_{det} and the relation budget K_{rel} follow the standard operating point of the SGG-Benchmark implementation and were held fixed throughout the field evaluation. The budget bounds false admissions (e.g., a spurious `occluded` edge that would trigger unnecessary cooperation) while retaining task-critical relations (e.g., the `occluded` edges required for a cooperative *push*). The gate determines which predictions enter the FACT set; it does not modify the deterministic re-encoding $g(\cdot)$ itself.

For the VLM reasoning module, $\mathcal{F}$ is serialized into a structured list (or equivalent structured text), preserving unique object indices and predicate and attribute tokens. Because $\mathcal{F}$ is derived deterministically from Γ, the reasoning module is restricted from introducing visual statements that are not present in the SGG output, ensuring strict grounding.

Applying SGG to greenhouse manipulation is more challenging than applying generic SGG to ordinary object-centric scenes. Plant organs are thin, deformable, visually similar, and heavily overlapped, which makes both object boundaries and pairwise predicates ambiguous. Moreover, several predicates are task-critical rather than merely descriptive. For example, a missed *occluded* relation can remove the evidence needed to select a cooperative *push* primitive, whereas a false *occluded* relation can trigger unnecessary environmental interaction. The proposed framework does not claim to eliminate these SGG errors. Instead, it constrains their downstream use by converting predicted objects, attributes, and relations into a closed FACT set and accepting a symbolic plan only when its claims are supported by this FACT set and by action-feasibility constraints.

3.3. VLM reasoning on grounded FACTs

This study represents vision as a set of grounded, discrete FACTs extracted using SGG (i.e., object, attribute, or relationship evidence), rather than raw pixels, and language indicates high-level reasoning and planning performed *only* on these vision-derived FACTs. Given a FACT set $\mathcal{F}$ (Section 3.2) and a task context $\mathcal{C}$ (ontology, action schema, and robot capability assumptions), the reasoning module outputs a symbolic decision tuple for a dual-arm platform:

$$\begin{aligned} \pi^{(s)} &= \left(\tau,\ v^{\star},\ \kappa,\ \mathbf{A}\right), \\ \mathbf{A} &= \{(\xi_k, \boldsymbol{\theta}_k)\}_{k=1}^{K}, \quad \xi_k \in \Xi, \end{aligned} \tag{21}$$

where τ represents the selected task type (e.g., harvesting, pruning, or thinning), $v^{\star} \in V$ denotes the target object node in the scene graph, $\kappa \in \{0,1\}$ indicates whether cooperative dual-arm assistance is required, and $\mathbf{A}$ represents a finite sequence of motion primitives with parameters $\boldsymbol{\theta}_k$ (e.g., target 3D point, push direction, or push magnitude), which are executed by the downstream controller.

To connect structured FACTs with a text-based LLM backend, $\mathcal{F}$ is serialized into a standard textual form $\mathbf{x} = \text{ser}(\mathcal{F}, \mathcal{C})$ that includes (i) object identifiers and classes ω_i, (ii) attributes a_i (e.g., ripeness and rigidity), and (iii) relationships r_{ij} as grounded predicates between object pairs. This representation aligns with recent robotics studies that ground language-model planning in explicit scene structures (e.g., 3D scene graphs or skill/affordance libraries) to enhance compositional reasoning under spatial constraints. Accordingly, VLM in this study denotes vision–language reasoning grounded in scene-graph-derived visual FACTs; the model does not receive raw images directly, but reasons over structured visual evidence produced by the SGG module.

3.3.1. Grounded context and constrained plan generation

Let $\Pi^{(s)}$ denote the set of candidate symbolic plans that satisfy the action schema in Eq. (21). The reasoning module is implemented as constrained text generation, with an instruction-tuned LLM from the Llama 3.1 family as the backend, a fixed prompt template, and deterministic post-generation verification, to produce a symbolic plan $\hat{\pi}^{(s)}$ conditioned on $\mathbf{x}$:

$$\begin{aligned} &\hat{\pi}^{(s)} = \arg\max_{\pi^{(s)} \in \Pi^{(s)}} \log P_\theta\left(\pi^{(s)} \mid \mathbf{x}\right) \\ &\text{s.t.} \quad C_{\text{schema}}(\pi^{(s)}) = 1,\ C_{\text{fact}}(\pi^{(s)}; \mathcal{F}) = 1, \end{aligned} \tag{22}$$

where $C_{\text{schema}}(\cdot)$ enforces strict compliance with the predefined output fields (task, target, cooperation flag, and primitive list), and $C_{\text{fact}}(\cdot;\mathcal{F})$ enforces *FACT-consistency*. Every referenced object ID must exist in the current scene graph, and every asserted relationship/attribute must be grounded in $\mathcal{F}$. Eq. (22) defines an *inference-time* constrained generation process.

The constrained decoding in Eq. (22) is implemented as a deterministic inference-time verification-and-repair process rather than prompt-only guidance, thereby improving reproducibility and preventing unconstrained language generation from directly propagating to execution. The VLM module must generate action plans in a strict JSON-based schema, where each candidate plan specifies the task types, target object identifiers, referenced relationships, and ordered sequences of action primitives.

After action-plan generation, each candidate's output is parsed deterministically and validated against two constraint sets. First, a *schema constraint* C_{schema} enforces the structural validity of the output (e.g., the required fields, admissible task labels, and allowed action primitives). Second, a *FACT constraint* C_{fact} verifies that all referenced objects, attributes, and relationships exactly match the grounded FACTs derived from the scene graph. Any additional visual claims not supported by the FACT set (e.g., referencing nonexistent objects, relationships, or attributes) are rejected.

Specifically, the planner output follows a fixed schema:

- `task`: one of {`harvest`, `prune`, `thin`, `idle`},
- `target_id`: an integer object ID,
- `cooperation`: a Boolean flag,
- `actions`: an ordered list of primitives, each with `primitive` ∈ {`approach`, `push`} and its parameters (e.g., acting arm, reference object ID, and push displacement).

The verifier applies rule-based checks: (i) *schema validity* (required keys, admissible tokens, and type checks); (ii) *FACT consistency* (all referenced `target_id` and action arguments must exist in $\mathcal{F}_t$ as object facts, and any used attribute/relation tokens must match the FACT vocabulary); and (iii) *execution feasibility gates* (e.g., IK success and collision-free planning for the resulting geometric goals). If any check fails, the system returns a structured violation message indicating the failure type (e.g., missing field, unknown token or object ID, unsupported relation or attribute token, or infeasible primitive) and requests a repair, with up to three attempts per trial.

If a generated plan violates any schema, grounding, or feasibility constraint, it is discarded, and the model is prompted to regenerate a new candidate under the same FACT set and prompt configuration. This regeneration process, based on repairs, is repeated up to three times; if unsuccessful, the planning step is considered a failure. Notably, no free-form natural-language output is accepted at any stage; only schema-valid and FACT-consistent plans are allowed to propagate to the action execution module. Therefore, constrained decoding serves as an explicit verification layer to enforce grounding consistency at inference time, rather than relying on implicit prompt engineering. Fig. 3 summarizes the inference-time verification and repair process.

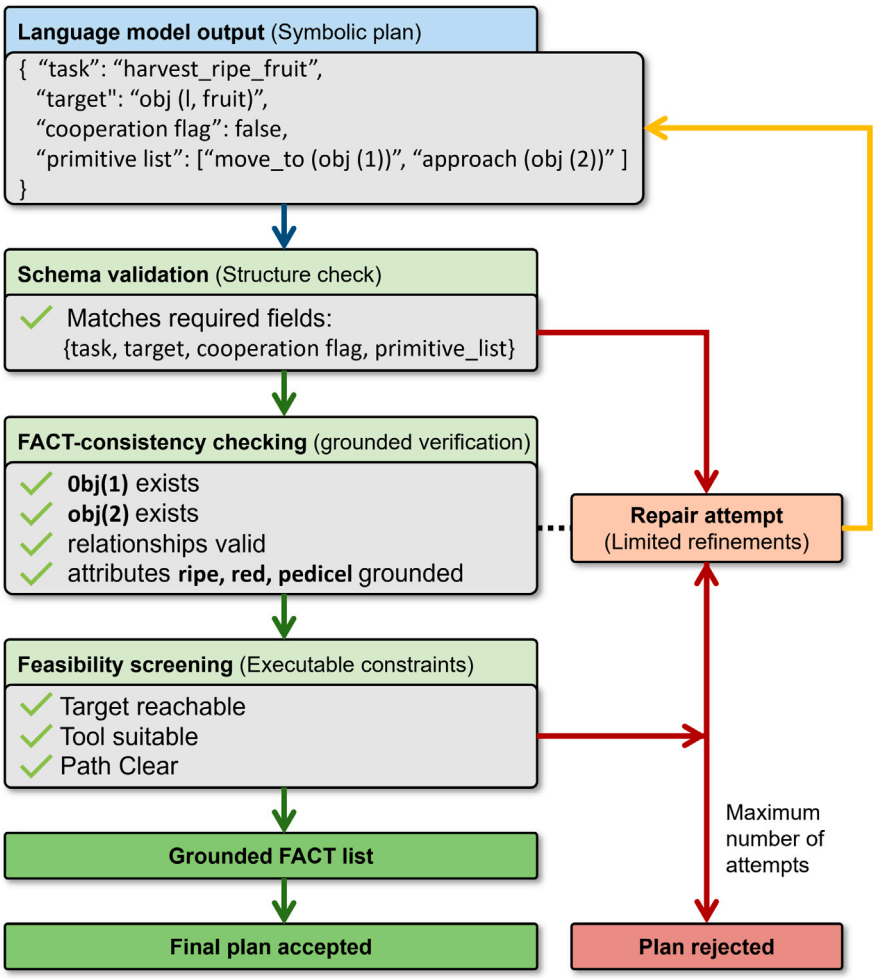

Fig. 3. Constrained decoding and verification: Schema validation, FACT-consistency checking, and feasibility screening with limited repair attempts.

3.3.2. Implementation, grounding safeguards, and schema compatibility

For reproducibility, the reasoning backend was instantiated with a fixed model configuration throughout all trials: Llama 3.1-8B-Instruct, temperature 0.2, top-p 0.9, and maximum output length 512 tokens. The same system prompt, ontology definition, JSON output schema, and repair protocol were used for all episodes. No scene-specific prompt tuning or manual intervention was performed during inference. When a candidate output violated the schema, referenced nonexistent FACTs, or failed the feasibility gate, the verifier returned a structured error message, and the model was allowed up to three repair attempts under the same grounded context.

During field operation, the elapsed time from submitting the serialized FACT prompt to receiving the final accepted symbolic plan was approximately 3–14 s per decision. This interval includes the initial VLM generation and any verification-triggered repair attempts. First-pass accepted plans were generally near the lower end of the range, whereas longer outputs or repair attempts increased the latency. This high-level reasoning delay is incurred once before motion execution and lies outside the low-level robot-control loop.

The computational burden of the reasoning pipeline can be separated into learned inference modules and deterministic checking modules. SGG inference and LLM token generation are the dominant learned inference components. In contrast, FACT construction, schema validation, FACT-consistency checking, repair-condition evaluation, and reasoning-to-action conversion are deterministic operations over the generated scene graph and plan. Let N denote the number of detected objects, R the number of predicted relations, P the number of generated plan arguments or steps, and K the number of repair attempts. FACT construction scales linearly with $N + R$, while FACT-consistency checking can be implemented by hash lookup over the FACT set and scales approximately with P. The repair loop adds at most K additional LLM generations, where K is bounded by the maximum retry number.

Unconstrained language generation suffers from hallucination (i.e., statements inconsistent with available evidence). In embodied settings, this may directly produce unsafe or nonexecutable plans. A grounding-first strategy is adopted to mitigate this risk: (i) the LLM receives only $\mathbf{x}$ derived from $\mathcal{F}$ as the visual context and (ii) the generated plan is validated by C_{schema} and C_{fact} before being passed to the action-interface module. This design aligns with the robotics literature, demonstrating that grounding LLMs with explicit feasibility or structure signals (skills, affordances, or scene graphs) enhances reliability in long-horizon planning. Moreover, this design aligns with surveys or benchmarks that promote evidence-consistency as a core requirement by explicitly constraining language-based reasoning to operate only on verified scene-graph FACTs, blocking unsupported visual claims during task planning.

For a controlled comparison with the prior rule-based visual scene understanding planner, the reasoning module retains the same agricultural ontology (task set, object and attribute vocabulary, and relationship predicates) and generates plans expressed in the same primitive library $\{\textsc{Approach}, \textsc{Push}\}$. Thus, differences in later experiments originate from the *reasoning layer*, which maps the grounded FACTs to the symbolic plan $\pi^{(s)}$, whereas the downstream geometric conversion and motion execution remain consistent across methods (see Fig. 4).

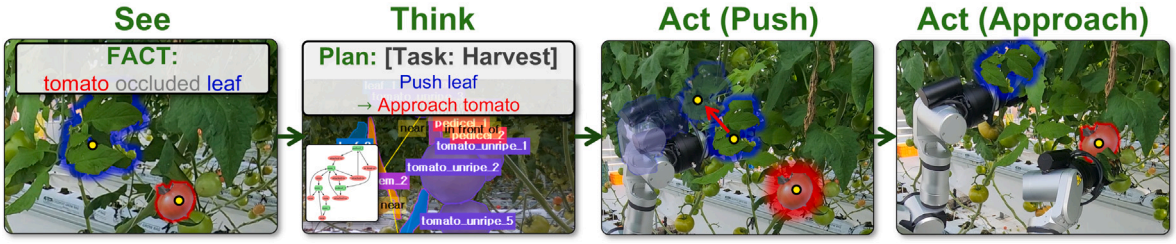

Fig. 4. Example of the perception–cognition–action pipeline driven by scene-graph-grounded reasoning. From left to right, the system (See) extracts grounded FACTs from the scene graph (e.g., tomato occluded by leaf), (Think) generates a symbolic task plan via VLM reasoning, and (Act) executes sequential manipulation primitives, including leaf pushing and target approach, to resolve occlusion and enable target access for harvesting. Additional trial-level examples, including both a successful case and a limitation case, are provided in Fig. 10.

3.4. Reasoning-to-action interface and action execution

Given the grounded FACT set $\mathcal{F}$ (from SGG) and the reasoning output (from the VLM reasoning module), this study represents the final executable plan as the following ordered action sequence:

$$\pi^{(a)} = \left[a_1, a_2, \ldots, a_M\right], \quad a_m \in \{\text{approach, push}\}, \tag{23}$$

where each a_m is instantiated with a designated arm (primary or secondary), a target entity index (a node in the scene graph), and primitive-specific parameters. This explicit *reasoning-to-action* interface is minimal; it converts LLM-level symbolic decisions made on $\mathcal{F}$ into executable geometric objectives while maintaining low-level feasibility (inverse kinematics, collision checking, and controller limits) in the robotic motion layer.

The primitive layer should be interpreted as an access-oriented manipulation interface rather than a complete crop-specific end-effector stack. The high-level task labels (Harvest, Prune/leaf removal, and Thin) specify the agronomic decision and target selection, whereas the primitive layer executes the approach or occlusion-clearing motion required to reach the target safely. Tool-specific operations such as cutting, detaching, grasping, or removing plant material are outside the primitive vocabulary used in this study and are treated as downstream end-effector actions or future extensions. This scope prevents the reported ASR from being interpreted as a universal success rate for full harvesting, pruning, or thinning automation.

3.4.1. 3D target-point computation from vision-derived detection

Let the intrinsic matrix of the RGB-D camera be $K \in \mathbb{R}^{3\times3}$ and a detected entity (e.g., a fruit or occlusion) have a representative image point $\mathbf{u} = [u, v, 1]^\top$ (e.g., the bounding-box center) with an associated depth d at that pixel. Under the standard pinhole depth-camera model, the corresponding 3D point in the camera frame is acquired using back-projection:

$$\mathbf{p}^C = \Pi^{-1}(\mathbf{u}, d; K) = d\, K^{-1}\mathbf{u}, \tag{24}$$

and the point in the robotic base frame is

$$\bar{\mathbf{p}}^B = {}^B T_C\, \bar{\mathbf{p}}^C, \quad \bar{\mathbf{p}} = \begin{bmatrix}\mathbf{p}\\ 1\end{bmatrix}, \tag{25}$$

where ${}^B T_C \in SE(3)$ denotes the calibrated camera-to-base extrinsic transform.

3.4.2. Manipulation primitives: approach and push

For a designated target node v_{tar}, the 3D center $\mathbf{p}^B_{\text{tar}}$ is computed via Eqs. (24) and (25), and an approach waypoint is defined as an offset along a fixed approach direction. Let $\mathbf{z}^C = [0, 0, 1]^\top$ denote the optical axis of the camera and $\mathbf{z}^B = {}^B R_C \mathbf{z}^C$ be its direction in the base frame, where ${}^B R_C$ represents the rotational part of ${}^B T_C$. Then, the approach point is

$$\mathbf{p}^B_{\text{app}} = \mathbf{p}^B_{\text{tar}} - \Delta_{\text{app}}\, \mathbf{z}^B, \quad \Delta_{\text{app}} > 0, \tag{26}$$

enforcing a frontal, camera-aligned approach that is consistent with the depth-ray geometry. The end-effector orientation is set using a fixed reference (e.g., camera-facing) orientation R^B_{EE} to reduce ambiguity and avoid unnecessary wrist reorientation. Hence, the resulting goal pose is $X^B_{\text{app}} = (R^B_{\text{EE}}, \mathbf{p}^B_{\text{app}})$.

When cooperative control is initiated by reasoning on $\mathcal{F}$ (e.g., a flexible occlusion is identified), the secondary arm executes a push action on the occluding entity v_{occ} to clear a path for the primary arm, which is consistent with the dual-arm cooperative paradigm (Jo et al., 2025; Park and Son, 2025). Let $\mathbf{p}^B_{\text{occ}}$ be the occlusion center and $\mathbf{p}^B_{\text{tar}}$ be the target center. This study defines the push direction as follows:

$$\mathbf{v}_{\text{dir}} = \frac{\mathbf{p}^B_{\text{occ}} - \mathbf{p}^B_{\text{tar}}}{\left\|\mathbf{p}^B_{\text{occ}} - \mathbf{p}^B_{\text{tar}}\right\|_2}, \tag{27}$$

and the displaced point is calculated as follows:

$$\mathbf{p}^B_{\text{occ}}{}' = \mathbf{p}^B_{\text{occ}} + \alpha\, \mathbf{v}_{\text{dir}}, \quad \alpha > 0, \tag{28}$$

which moves the occlusion away from the target along the relative line of sight. This study adopts pushing as a simple nonprehensile manipulation primitive, which has been widely studied as a planning-relevant interaction modality. In practice, the push primitive is executed as a short, approximately straight end-effector segment from a precontact pose toward $\mathbf{p}^B_{\text{occ}}{}'$, maintaining a conservative contact profile suitable for foliage-level interaction (without relying on learned vision–language-action (VLA) policies). Fig. 5 illustrates the geometric construction of approach and push primitives.

3.4.3. Feasible motion generation and execution

For each primitive goal pose X^B (approach or push waypoint), the motion layer calculates a joint configuration $\mathbf{q}^\star$ by solving the inverse kinematics subject to the joint limits:

$$\begin{aligned}\mathbf{q}^\star &= \arg\min_{\mathbf{q}} \left\|\mathrm{FK}(\mathbf{q}) \ominus X^B\right\| \\ \text{s.t.} \quad & \mathbf{q}_{\min} \le \mathbf{q} \le \mathbf{q}_{\max},\end{aligned} \tag{29}$$

where $\mathrm{FK}(\cdot)$ denotes forward kinematics, and $\ominus$ represents a pose-difference operator. Standard sampling-based motion planning with collision checking is employed as a feasibility filter to respect reachability constraints in cluttered greenhouse workspaces, a standard approach for high degrees-of-freedom (DoF) manipulation. If no feasible trajectory is determined for a candidate primitive, the system rejects that primitive and requests replanning at the reasoning level on refreshed grounded FACTs rather than forcing execution.

The proposed method differs from representative approaches primarily in how relational structure is exposed and controlled during planning (Table 3). Unlike end-to-end models that lack explicit relational structure, the proposed framework uses FACT-based representations to enable relational reasoning and occlusion-aware task planning, while maintaining explicit control over execution.

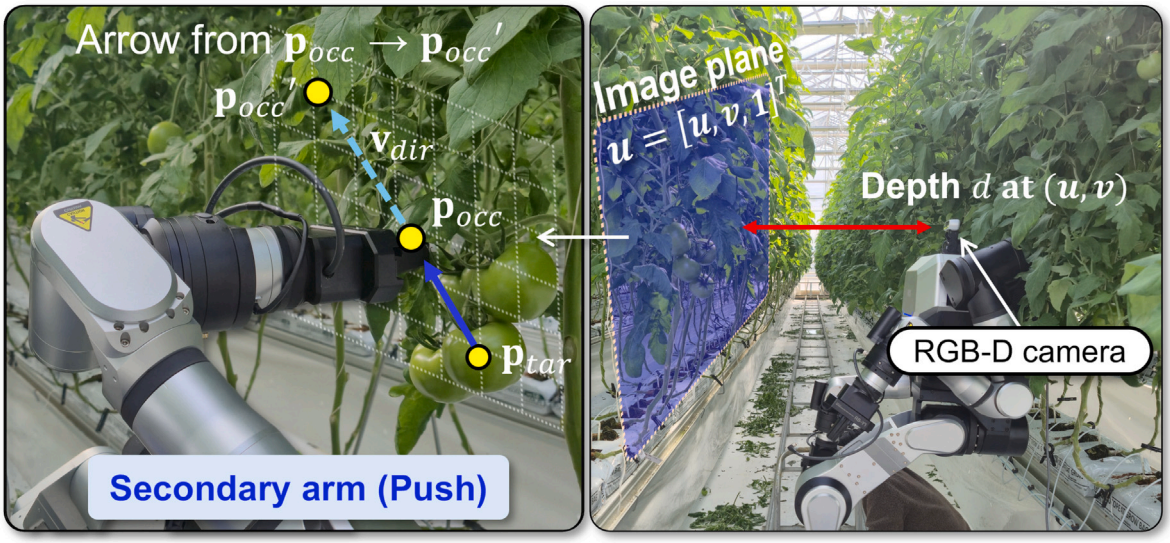

Fig. 5. Reasoning-to-action geometry: The frame transform for target points, approach offsets, and push directions defined relative to the target and occlusion.

Table 3
Comparison of system characteristics across approaches.

Method	Structured rep.	Relational reasoning	Occlusion-aware	Planning type	Execution
End-to-end VLA models	×	✓	Limited	Implicit	Direct
Classical pipeline	Partial	×	Limited	Rule-based	Explicit
Ours (FACT-based)	✓	✓	✓	Language-based	Explicit

Algorithm 1 Rule-based baseline (Park and Son, 2025).

Require: Scene graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, FACTs $\mathcal{F}_t$, rule set $\mathcal{R}_{\text{rule}}$, threshold δ_{leaf}
Ensure: Task τ_t, target set $\mathcal{V}_t^\star$, cooperation flag c_t, occlusion sets $\mathcal{V}_t^{\text{rig}}, \mathcal{V}_t^{\text{flex}}$

$\mathcal{V}_t^{\text{ripe}} \leftarrow \{v \in \mathcal{V}_t \mid \omega(v) = \texttt{fruit}, \texttt{ripe} \in \mathbf{a}(v)\}$
$\mathcal{V}_t^{\text{leaf}} \leftarrow \{v \in \mathcal{V}_t \mid \omega(v) = \texttt{leaf}\}$
$\mathcal{V}_t^{\text{unripe}} \leftarrow \{v \in \mathcal{V}_t \mid \omega(v) = \texttt{fruit}, \texttt{unripe} \in \mathbf{a}(v)\}$
if $|\mathcal{V}_t^{\text{ripe}}| > 0$ **then** ▷ Case 1: Harvest
  $\tau_t \leftarrow$ HARVEST; $\mathcal{V}_t^\star \leftarrow \mathcal{V}_t^{\text{ripe}}$
else if $|\mathcal{V}_t^{\text{leaf}}| \geq \delta_{\text{leaf}}$ **then** ▷ Case 2: Prune
  $\tau_t \leftarrow$ PRUNE; $\mathcal{V}_t^\star \leftarrow \mathcal{V}_t^{\text{leaf}}$
else if EXISTSDENSEUNRIPECLUSTER$(\mathcal{G}_t, \mathcal{V}_t^{\text{unripe}})$ **then** ▷ Case 3: Fruit-thinning
  $\tau_t \leftarrow$ THIN; $\mathcal{V}_t^\star \leftarrow$ SELECTTHINTARGETS$(\mathcal{G}_t, \mathcal{V}_t^{\text{unripe}})$
else
  $\tau_t \leftarrow$ IDLE; $\mathcal{V}_t^\star \leftarrow \emptyset$
$\mathcal{V}_t^{\text{occ}} \leftarrow \{v_j \mid \exists (v_i, v_j) \in \mathcal{E}_t,\ v_i \in \mathcal{V}_t^\star,\ r_{ij} = \texttt{occluded}\}$
$\mathcal{V}_t^{\text{rig}} \leftarrow \{v \in \mathcal{V}_t^{\text{occ}} \mid \texttt{rigid} \in \mathbf{a}(v)\}$
$\mathcal{V}_t^{\text{flex}} \leftarrow \{v \in \mathcal{V}_t^{\text{occ}} \mid \texttt{flexible} \in \mathbf{a}(v)\}$
$c_t \leftarrow \mathbf{1}\left(|\mathcal{V}_t^{\text{flex}}| \geq 1\right)$
return $(\tau_t, \mathcal{V}_t^\star, c_t, \mathcal{V}_t^{\text{rig}}, \mathcal{V}_t^{\text{flex}})$

3.5. Algorithms

This subsection summarizes the two algorithms in this study: (i) a rule-based baseline planner adapted from the prior visual scene understanding task-planning framework (Park and Son, 2025) and (ii) the proposed SGG-grounded VLM reasoning-to-action framework that converts grounded FACTs into an executable dual-arm action sequence.

3.5.1. Algorithm 1: Rule-based baseline task-and-cooperation decision

Let $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ denote the scene graph at time t and $\mathcal{F}_t$ be the corresponding grounded FACT set (defined in Section 3.2). The baseline planner $h_{\mathcal{R}_{\text{rule}}}$ maps FACTs to a task label $\tau_t \in \mathcal{T}$ and a target node set $\mathcal{V}_t^\star \subseteq \mathcal{V}_t$, that is,

$$(\tau_t, \mathcal{V}_t^\star) = h_{\mathcal{R}_{\text{rule}}}(\mathcal{F}_t), \tag{30}$$

where $\mathcal{R}_{\text{rule}}$ denotes an adapted deterministic rule set. This paper considers $\mathcal{T} = \{\text{HARVEST}, \text{PRUNE}, \text{THIN}\}$, where THIN indicates fruit thinning.

After selecting $(\tau_t, \mathcal{V}_t^\star)$, the baseline assesses whether dual-arm cooperative control is necessary by detecting occlusions of the selected target and checking their rigidity attributes:

$$\begin{aligned}
\mathcal{V}_t^{\text{occ}} &= \left\{ v_j \in \mathcal{V}_t \,\middle|\, \exists (v_i, v_j) \in \mathcal{E}_t,\ v_i \in \mathcal{V}_t^\star,\ r_{ij} = \texttt{occluded} \right\}, \\
\mathcal{V}_t^{\text{rig}} &= \left\{ v \in \mathcal{V}_t^{\text{occ}} \,\middle|\, \texttt{rigid} \in \mathbf{a}(v) \right\}, \\
\mathcal{V}_t^{\text{flex}} &= \left\{ v \in \mathcal{V}_t^{\text{occ}} \,\middle|\, \texttt{flexible} \in \mathbf{a}(v) \right\}, \\
c_t &= \mathbf{1}\left[\left| \mathcal{V}_t^{\text{flex}} \right| \geq 1 \right],
\end{aligned} \tag{31}$$

where $\mathbf{a}(v)$ denotes the attribute set of node v, and $c_t \in \{0, 1\}$ represents a cooperation flag.

3.5.2. Algorithm 2: SGG-grounded VLM reasoning-to-action framework

Given $\mathcal{F}_t$, the proposed cognitive planner generates an executable action sequence $\pi_t^{(a)} = (a_1, \ldots, a_M)$ with parameters $\Theta_t = (\theta_1, \ldots, \theta_M)$ under explicit constraints, as follows:

$$\begin{aligned}
(\pi_t^{(a)}, \Theta_t) &= \arg\max_{\pi^{(a)}, \Theta} P_{\text{VLM}}(\pi^{(a)}, \Theta \mid \mathcal{F}_t, C_t) \\
\text{s.t.} \quad & a_m \in \Xi \ \ \forall m, \text{Ver}(\pi^{(a)}, \Theta; \mathcal{F}_t, C_t) = 1,
\end{aligned} \tag{32}$$

where $\Xi = \{\text{APPROACH}, \text{PUSH}\}$ represents the motion-primitive set, C_t denotes the execution context (robot state, camera calibration, and safety margins), and $\text{Ver}(\cdot) = 1$ indicates a Boolean feasibility-and-consistency checker.

4. Experiments and results

This section presents the experimental setup, evaluation protocol, and results in a unified flow, and outlines the field-experiment protocol for validating the proposed SGG-grounded VLM reasoning process with an emphasis on reproducibility. The protocol includes (i) defining and logging a trial (episode), (ii) evaluating perception as structured triplets/FACTs, (iii) assessing the logical validity and decision quality

Algorithm 2 SGG-grounded VLM reasoning-to-action framework (constrained generation on grounded FACTs).

Require: FACTs $\mathcal{F}_t$, context C_t, primitive set $\Xi = \{\textsc{Approach}, \textsc{Push}\}$, VLM planner P_{VLM}, checker $\mathsf{Ver}(\cdot)$
Ensure: Executable action sequence $(\pi_t^{(a)}, \Theta_t)$
Plan generation (grounded reasoning):
  Query P_{VLM} with $\mathcal{F}_t$ and an action schema to obtain a candidate plan $\widehat{\pi}_t^{(a)}$ and parameters $\widehat{\Theta}_t$
Parsing and grounding:
  Parse $(\widehat{\pi}_t^{(a)}, \widehat{\Theta}_t)$ into $(\pi_t^{(a)}, \Theta_t)$ such that each argument refers to an entity, attribute, or relationship in $\mathcal{F}_t$
Consistency and feasibility filtering:
if $\exists a \in \pi_t^{(a)}$ s.t. $a \notin \Xi$ **or** $\mathsf{Ver}(\pi_t^{(a)}, \Theta_t; \mathcal{F}_t, C_t) = 0$ **then**
  Request a repair from P_{VLM} using the detected violations; update $(\pi_t^{(a)}, \Theta_t)$
Action execution:
for $k = 1$ to $|\pi_t^{(a)}|$ **do**
  Compute 3D goals required by (a_k, θ_k) from the vision-derived geometry (Section 3.4)
  Execute a_k on the dual-arm platform with safety checks; update C_t
return $(\pi_t^{(a)}, \Theta_t)$

of planned actions, and (iv) measuring physical action outcomes on the dual-arm platform. The results are reported along the same evaluation axes so that each outcome can be directly linked to the corresponding experimental condition and metric definition.

4.1. Field experiment setup

A trial is defined as a single end-to-end episode starting from an RGB-D observation of a canopy and terminating after executing the corresponding action sequence (or aborting because of safety or feasibility problems). For each trial, an expert operator assigned reference labels for the task class, cooperative-control necessity, and target instance according to a predefined annotation rule set based on visible crop state, occlusion context, and immediate action feasibility. A single experimenter performed all metric computations for the precondition satisfaction rate (PSR), feasibility and executability (FnE), and correct action sequence (CAS) using a predefined, deterministic evaluation protocol (Section 4.2).

The validation was conducted as a closed-loop field experiment rather than as an offline perception or reasoning benchmark. In each trial, the robot received a new greenhouse scene observation, generated SGG-derived FACTs, reasoned over the FACT set, validated the symbolic plan, converted the plan into action primitives, and attempted physical execution with the dual-arm platform. A trial ended when the commanded action sequence succeeded, failed during execution, or was rejected by feasibility or safety checks.

Environment. All experiments were conducted on a dual-arm robotic platform in a greenhouse environment at Smart Farm Innovalley in Goheung, Republic of Korea. The greenhouse is a real-world horticultural production environment characterized by dense canopies; frequent occlusion of leaves, stems, and fruits; and heterogeneous plant-growth conditions. The robotic system was mounted on a standard manual work cart commonly used by workers in daily greenhouse operations to enable direct access to the crop rows without modifying the greenhouse infrastructure. This setup enabled the robot to be positioned at various points along the cultivation lanes, closely emulating practical deployment conditions in which robots must share a workspace with human operators and tools. Throughout all experiments, the robot was considered a fixed-base system after being placed on the cart,

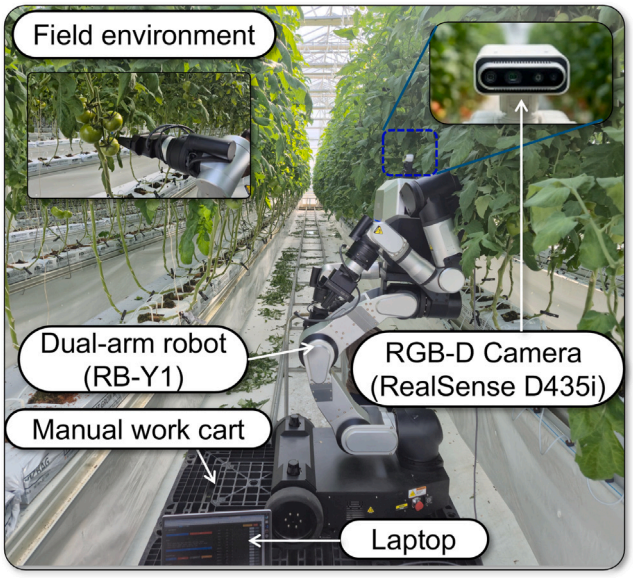

Fig. 6. Field experiment setup in a commercial tomato greenhouse: Robot on a manual work cart, RB-Y1 dual-arm platform, and forward-facing RGB-D sensor configuration.

without any imposed workspace constraints or artificial scene simplifications. The experimental workspace was entered at randomly selected locations along the greenhouse rows, without prior selection or scene curation, to evaluate the decision-making and task-selection behavior of the robot under diverse, unstructured conditions. Therefore, the robot encountered a broad range of naturally occurring scenarios, including varying degrees of occlusion, fruit clustering, leaf interference, and plant morphology differences.

Experiments were conducted during the tomato production season, in which harvesting, fruit thinning, and leaf removal tasks co-occur. Trials were conducted in December, a period when ripe fruits, immature fruits, and excessive foliage coexist in the canopy. All experiments were performed between 10:00 and 15:00 to ensure stable daylight illumination while capturing typical, realistic variations in lighting and shadow conditions in a greenhouse environment.

Hardware. The hardware configuration combines a commercial dual-arm platform with a forward-facing RGB-D sensor mounted on the greenhouse work cart (Fig. 6). All trials were executed on the Rainbow Robotics RB-Y1 dual-arm platform (RB-Y1a; Rainbow Robotics, Republic of Korea). This study uses the two 7-DoF articulated manipulators mounted on the RB-Y1 body, each equipped with a 1-DoF parallel-jaw gripper (Rainbow Robotics, Republic of Korea), for plant interaction and object handling. For all greenhouse experiments, the system was operated as a fixed-base dual-arm robot. The mobile base (if available) was stationary, and all action generation and evaluation focused on arm-level manipulation rather than base navigation or whole-body locomotion.

A forward-facing RGB-D stream was acquired using an Intel RealSense D435i (Intel Corporation, USA), providing hardware-synchronized RGB and depth frames with inertial measurements from the onboard inertial measurement unit. The depth modality was employed to back-project 2D pixel measurements onto 3D metric points under the camera frame to construct action primitives (e.g., *approach* and *push*) from image-space selections. Given a pixel coordinate (u, v) and its depth value d, the corresponding 3D point $\mathbf{p}_c = [x\ y\ z]^\top$ was recovered using standard pinhole back-projection with calibrated camera intrinsics and then transformed into the robot reference frame using camera-to-robot extrinsic calibration when generating end-effector targets.

Table 4
Deterministic evaluation criteria for each axis (trial level).

Axis	Metric	Deterministic criterion (summary)
Validity	PSR	Plan preconditions supported by FACTs
Validity	FnE	IK/planning success, collision/safety constraints satisfied
Validity	CAS	Primitive ordering consistent with intent (e.g., push→approach)
Decision	TPA/CDA/TSA	Agreement with expert reference label
Execution	ASR/successful-and-safe execution	Completed sequence + intended effect; successful-and-safe requires no safety risk

4.2. Evaluation protocol and metrics

Reference labels were assigned using a predefined rule sheet rather than free-form judgment. Harvesting was labeled when at least one ripe fruit required collection in the observed workspace; pruning was labeled when foliage removal was necessary to improve safe access or reduce obstruction; and thinning was labeled when unripe fruits formed a locally dense cluster requiring removal. Cooperative control was labeled when the primary target was obstructed by an object judged to require secondary-arm intervention under the same scene state. When multiple candidate targets were present, the reference target was defined as the object that best matched the selected task under the local canopy context and immediate feasibility constraints.

This study organizes the evaluation along four axes that mirror the following stages: **perception** (quality of SGG triplets and derived FACTs), **logical validity** (FACT-consistency and executability of the planned actions), **decision accuracy** (DA; agreement with expert reference decisions for the task, cooperation, and target), and **action execution** (physical success and successful-and-safe execution during manipulation). Each trial is logged with binary indicators for these axes, enabling raw rates and validity-filtered analyses (e.g., DA computed only when the plan is logically valid). Although the reasoning backend is implemented using an LLM, the overall system is formulated as a vision-grounded language reasoning framework because all planning is conditioned on visual evidence derived from the SGG module. Table 4 summarizes the evaluation flags and their deterministic criteria.

Perception metrics: SGG and FACT quality. This study presents an evaluation of scene-graph perception using the standard SGG settings widely adopted in the literature. Predicate classification assumes ground-truth (GT) object boxes and labels and evaluates predicate prediction. Scene-graph classification assumes GT boxes but predicts object labels and predicates. Scene-graph detection (SGDet) jointly predicts object detections and predicates. For each setting, the relationship recall at K (R@K) and mean recall at K (mR@K) are reported to capture the overall retrieval and class-balanced performance under long-tailed predicate distributions.

In addition to the relationship metrics, the object- and attribute-level correctness values are logged when available in the evaluation protocol because the downstream FACT construction depends on the precision of the node (object/attribute) and edge semantics (predicate). This perception axis evaluates the *structured evidence* (triplets and attributes), not raw pixels. The goal is to quantify how reliably the canopy is converted into grounded symbolic evidence for reasoning.

Logical-validity metrics: FACT-consistent and executable planning. Logical validity assesses whether the planned action sequence is *consistent with* the grounded FACT set and *executable*, given immediate physical constraints. Each trial assesses three complementary aspects: (i) *PSR*, which determines whether the preconditions claimed by the plan (e.g., an object is occluding the target or an obstacle is classified as rigid/flexible) are supported by FACTs; (ii) *FnE*, which flags physically infeasible plans given kinematic reachability, collision risk, or workspace limits; and (iii) *CAS*, which verifies that the proposed ordering of primitives (e.g., push-then-approach for occlusion clearing) is logically coherent with the stated intent and constraints. All logical-validity metrics (PSR, FnE, and CAS) were assessed by a single experimenter following a predefined and deterministic scoring protocol applied consistently across all trials. The experimenter did not intervene with the robotic behavior and merely acted as an observer applying fixed criteria to logged execution outcomes and observable robotic states. Particularly, FnE was marked as a failure if the inverse kinematics failed, the motion planner failed, collision or contact with nontarget rigid structures was detected, an emergency stop occurred, or a controller aborted due to safety constraints. No inter-rater agreement analysis was conducted because only one evaluator applied the scoring protocol; this is a limitation of the present study and should be addressed in future validation with multiple raters.

This study logs hallucination as *evidence-inconsistency* with respect to the grounded FACT set. We define a hallucination event when a generated plan references any entity ID, attribute token, or relation token that is not present in the current FACT set $\mathcal{F}_t$. In reporting, we use a *trial-level hallucination rate* (HR), where a trial is marked as hallucinated if *any* generation attempt (up to three repair attempts) produces an evidence-inconsistent plan before a valid plan is obtained or the trial terminates. This definition matches the practical deployment risk: a hallucinated attempt indicates that the generator can produce unsupported visual claims unless the verifier blocks it. In this setting, hallucination is operationalized as a plan justification unsupported by *FACTs* rather than a purely linguistic phenomenon, aligning with the established definitions of hallucination as content not supported by evidence.

The proposed validity gate is intended to prevent physically unsafe executions, and its effect is quantified by the *unsafe execution prevention rate (UEPR)*:

$$\mathrm{UEPR} = \frac{|\{\text{invalid plans that are aborted before execution}\}|}{|\{\text{invalid plans}\}|}. \tag{33}$$

A high unsafe execution prevention rate indicates that the system blocks physically risky actions (e.g., collision-prone approaches or rigid-structure interference) before the robot executes them.

Decision-agreement metrics: Agreement with expert references. Decision accuracy must be interpreted component-wise because task selection, cooperation decision, and target selection can fail independently within a single trial. The DA metric measures agreement between the planner outputs and expert reference labels for task class selection, cooperative-control necessity, and target instance selection. This study records three trial-level agreement flags: *task prediction agreement* (TPA), *cooperative decision agreement* (CDA), and *target selection agreement* (TSA). When multiple decisions are required in a single trial (e.g., selecting a target among multiple ripe fruits and deciding whether to push an occlusion), these flags determine which decision component succeeded or failed.

Because high-level decisions in robotic manipulation in greenhouses can be partially subjective (e.g., alternative targets may both be acceptable), a validity-filtered view of the decision quality is presented. *Validity-filtered DA (VF-DA)* is computed by restricting the DA statistics to trials whose plans are logically valid (where PSR, FnE, and CAS are satisfied), preventing physically impossible or FACT-inconsistent plans from being counted as incorrect decisions when the true failure mode is invalid reasoning or grounding.

Execution metrics: Physical success and safety. Action execution evaluates whether the motion-primitive sequence can be executed by the robot and whether it achieves the intended access or occlusion-management outcome in the canopy. This study logs the *action success rate* (ASR) as a trial-level indicator of whether the robot completed

the planned sequence without aborting and achieved the intended manipulation effect, such as displacing an occlusion or reaching the target access pose. Separately, this study reports the *successful-and-safe execution rate*, which counts only trials that both succeeded and completed without an observed safety risk. Aborted trials are counted as ASR failures and are not counted as successful-and-safe executions; however, conservative aborts are also reported separately because they indicate that the validity or feasibility gate prevented potentially unsafe execution rather than allowing uncontrolled physical interaction.

This study also reports the *validity-to-execution consistency* (VEC) to connect reasoning quality to embodied outcomes, defined as the fraction of strict accepted plans (PSR, FnE, CAS, and no hallucination flag) that lead to successful and safe execution. The VEC quantifies the practical reliability of the reasoning-to-action interface under severe greenhouse occlusion, independent of subjective task-label disagreements.

4.2.1. Relation-error attribution from field logs

The task-stratified statistics in Table 8 are descriptive and cannot, by themselves, establish that erroneous relational predictions cause faulty robot decisions. To provide direct evidence, a decision-level attribution analysis is conducted over the N=50 field trials using three per-trial quantities recorded during the campaign: the ground-truth cooperation requirement of the canopy state (annotated under the rule sheet of Section 4.2), the cooperation decision of the generated plan, and the plan-level hallucination and action-sequence-consistency flags used for HR and CAS.

A *cooperation flip* is a trial whose planned cooperation decision disagrees with the ground-truth cooperation requirement. Flips are decomposed into *missed cooperation* (MC: cooperation required but not planned) and *unnecessary cooperative manipulation* (UCM: a cooperative *push* planned although no occlusion required clearance). The attribution then follows deductively from the FACT-closed architecture. Because the planner is restricted to the serialized FACT set, a cooperation claim in a hallucination-free plan must be supported by an `occluded` FACT; a UCM without hallucination therefore implies a *false* `occluded` relation admitted into the FACT set. Likewise, an MC in a plan that is hallucination-free and action-sequence-consistent implies that the required `occluded` FACT was *absent*, i.e., a missed relation. Flips in trials flagged for hallucination are attributed to the reasoning channel instead. The association between flips and the hallucination flag is tested with Fisher's exact test, which is appropriate for the small cell counts at N=50, to verify that flips are driven by the relational FACT content rather than by ungrounded generation.

This attribution is deductive rather than interventional: the per-trial serialized FACT sets and prompts were not retained after the field campaign, so counterfactual replay of edited FACT sets and oracle re-planning with ground-truth FACTs could not be performed. The corresponding decision-level headroom is therefore reported as a bound implied by the attribution, and interventional replay under full FACT logging is identified as future work in Section 5.6.

4.3. Field trial overview

The field trials indicate that the proposed framework achieves relatively high plan-level consistency, while a clear gap remains between logical validity and physical execution robustness. Fig. 7 summarizes the end-to-end performance across evaluation axes, and Table 5 presents the 50 conducted field trials. After expert annotation and task consolidation, the GT task distribution for evaluation comprised 11 harvesting trials (H), 14 leaf-removal/pruning trials (P), and 25 thinning trials (T). In addition, 14 trials required cooperative control according to the GT annotations ($CC = True$), whereas 36 trials were noncooperative ($CC = False$).

Thus, the reported trials evaluate the complete perception–reasoning–action loop. The logged outcomes include not only perception and reasoning metrics, but also physical action success, successful-and-safe outcomes, and abort events. This logging design prevents plan-level validity from being conflated with real-world manipulation success.

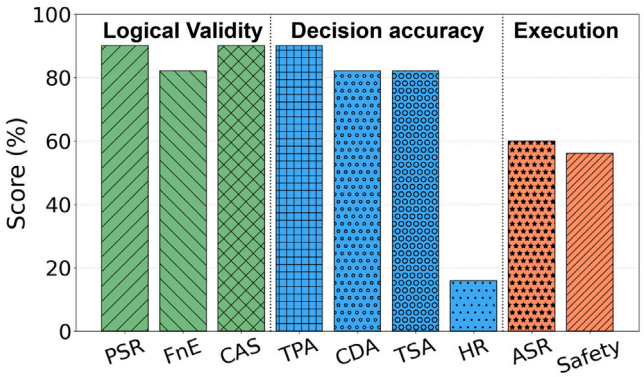

Fig. 7. Summary plots of end-to-end evaluation: logical validity (precondition satisfaction rate, PSR; feasibility and executability, FnE; and correct action sequence, CAS), decision accuracy (task-planning agreement, TPA; cooperative decision agreement, CDA; target selection agreement, TSA; and hallucination rate, HR), and execution (action success rate, ASR; successful-and-safe execution rate; and execution time).

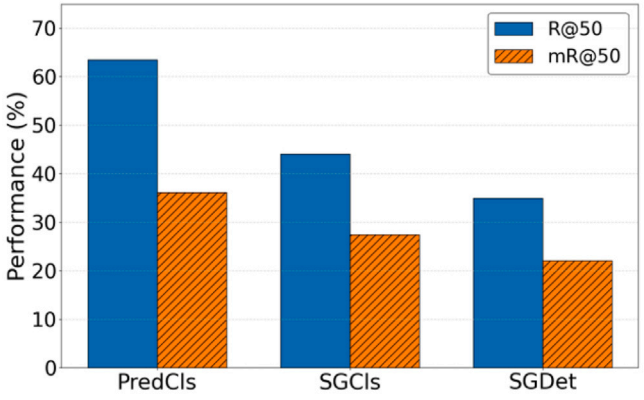

Fig. 8. Scene-graph generation performance in predicate classification (PredCls), scene-graph classification (SGCls), and scene-graph detection (SGDet) settings. Relationship recall (R@50) and mean recall (mR@50) are reported to assess the relational grounding quality.

4.4. Perception results: Scene-graph quality for grounded FACT extraction

The perception stack provides useful but still imperfect relational grounding, with the largest degradation appearing in the full SGDet setting. Table 6 reports the quantitative SGG results employed as the basis for grounded FACT extraction. Under the standard SGG evaluation modes, the relationship recall (Rel R@50) and mean recall at K=50 (Rel mR@50) were 63.5 and 36.2, respectively, for predicate classification and 44.1 and 27.5, respectively, for scene-graph classification. For SGDet, the object detection and object class accuracy were 67.8 and 75.2, respectively, whereas the relationship performance reached 35.0 (Rel R@50) and 22.1 (Rel mR@50), respectively. Attribute recognition evaluated using SGDet achieved 64.0 (Attr R@50) and 47.0 (Attr mR@50). Fig. 8 summarizes the SGG performance across standard evaluation modes.

These results also provide an evidence-level sensitivity analysis for the reasoning framework. The relation scores decrease from PredCls (Rel R@50/mR@50 = 63.5/36.2) to SGCls (44.1/27.5) and SGDet (35.0/22.1), showing that detection and classification errors substantially affect the relation FACTs used by the reasoning module. Therefore, the overall framework is not independent of SGG quality. It is particularly sensitive to task-critical relation predicates, such as *occluded*, *attached_to*, and *near*, because these predicates can change target

Table 5
Field evaluation results ($N = 50$ trials).

Category	Metric	Value
I. Perception	(Table 6)	–
II. Logical validity	PSR (Precondition satisfaction rate)	0.90 (45/50)
	FnE (Feasibility and executability)	0.82 (41/50)
	CAS (Correct action sequence)	0.90 (45/50)
	V_{plan} (PSR ∧ FnE ∧ CAS)	0.82 (41/50)
	V_{strict} ($V_{\text{plan}} \wedge \neg$HR)	0.78 (39/50)
III. Decision accuracy	DA (Full tuple agreement)	0.58 (29/50)
	TPA (Task-planning agreement)	0.90 (45/50)
	CDA (Cooperation decision agreement)	0.82 (41/50)
	TSA (Target selection agreement)	0.82 (41/50)
	HR (Hallucination rate)	0.16 (8/50)
IV. Action execution	ASR (Action success rate)	0.60 (30/50)
	Time (Avg. execution time; cooperative/noncooperative)	9.8/15.2 s
	Successful-and-safe execution rate	0.56 (28/50)
	UEPR (invalid logical plans blocked before execution)	1.00 (12/12)

Table 6
Scene-graph generation (SGG) performance and evidence-quality sensitivity for grounded FACT extraction.

Metric	OD	OCA	Rel R@50	Rel mR@50	Attr R@50	Attr mR@50
PredCls	GT	GT	63.5	36.2	GT	GT
SGCls	GT	85.8	44.1	27.5	GT	GT
SGDet	67.8	75.2	35.0	22.1	64.0	47.0

OD: object detection, OCA: object class accuracy,
Rel: relationship, Attr: attribute, GT: ground truth

selection, cooperation decisions, and the selection of *push* or *approach* primitives.

This discussion also clarifies the scope of the technical contribution. Recent SGG models address important perception-level problems, including predicate bias, relation imbalance, causal relation modeling, multimodal fusion, and relation uncertainty. These advances could improve the quality of the FACT set used by the proposed framework. However, even a stronger SGG model does not by itself define how visual relations should be checked, rejected, repaired, or converted into robot actions. The proposed framework therefore addresses a different interface-level problem: how to use imperfect structured perception as auditable evidence for symbolic reasoning and dual-arm execution.

The degradation from PredCls to SGDet also indicates how perception errors can propagate to downstream reasoning. In PredCls, object boxes and object labels are given, so the relation score mainly reflects predicate recognition. In contrast, SGDet jointly depends on object detection, object classification, and relation prediction. The decrease from 63.5/36.2 in PredCls to 35.0/22.1 in SGDet therefore shows that the FACT set used by the reasoning module is sensitive to missing or incorrectly classified objects as well as incorrect relations. This is important for action planning because predicates such as *occluded*, *attached_to*, and *near* can change target selection, cooperation decisions, and whether the robot selects a direct *approach* primitive or a cooperative *push–approach* sequence.

The proposed framework handles these errors conservatively rather than fully disambiguating them. The FACT-consistency checker prevents the VLM planner from inventing objects, attributes, or relations that are absent from the current FACT set. However, if an incorrect relation is already present in the FACT set, the verifier treats it as available evidence; if a true relation is missing from the FACT set, the verifier cannot recover it. Downstream protection is therefore provided by feasibility and safety gating: plans whose preconditions, kinematic feasibility, or primitive ordering are inconsistent with the current FACTs or robot constraints are rejected or aborted before unsafe execution. This design reduces the propagation of unsupported VLM-generated claims, but it does not eliminate perception-level uncertainty. Future work should propagate relation confidence from SGG to FACTs and use uncertainty-aware reasoning, active re-observation, or tactile feedback to disambiguate critical relations before execution.

This strength–limitation trade-off of relation-dependent reasoning is directly observable in the qualitative examples (Fig. 10). The successful case shows that scene-graph relations can provide a compact and inspectable bridge from visual perception to cooperative action selection. Because the detected relation is serialized as a FACT, the VLM reasoning output can be checked against the current scene evidence before execution. This makes the reasoning chain more auditable than an unconstrained image-to-action generation process.

At the same time, the limitation case shows that the framework remains dependent on the correctness of the SGG evidence. FACT consistency prevents the VLM from inventing visual relations that are absent from the FACT set, but it does not determine whether the FACT set itself is correct. Thus, missed or false-positive relations should be treated as uncertain visual evidence at the system level. This motivates future work on confidence-aware FACTs, uncertainty-aware SGG, active re-observation, and tactile or force feedback during execution.

4.5. Logical validity results

The proposed framework maintains high plan-level logical validity across the field trials. Table 5 presents the logical validity results. The PSR was 0.90 (45/50), indicating that the plans typically followed the required preconditions for the selected task and manipulation context. The FnE score was 0.82 (41/50), reflecting cases where the reasoned plan could be executed under the platform and scene constraints. The CAS score also reached 0.90 (45/50), suggesting that the high-level ordering of motion primitives was consistent with the intended manipulation logic when the plans were generated (see Table 7).

This diagnostic analysis separates the validation criteria that contribute to the final logical-validity score. The combination of PSR, FnE, and CAS was satisfied in 41 of 50 trials, whereas adding the hallucination filter reduced the accepted logically valid set to 39 of 50 trials. This indicates that unsupported visual claims were not the dominant failure mode, but FACT-consistency checking still prevented hallucination-flagged outputs from being counted as logically valid plans. This table should be interpreted as a post-hoc validation-stage diagnostic rather than a full counterfactual ablation of the repair loop.

Table 7
Post-hoc validation-stage diagnostic analysis using the 50 field-trial logs.

Validation criterion	Passed trials	Rate
Precondition satisfaction (PSR)	45/50	90.0%
Feasibility/executability (FnE)	41/50	82.0%
Correct action sequence (CAS)	45/50	90.0%
Hallucination flag (HR)	8/50	16.0%
PSR ∧ FnE ∧ CAS	41/50	82.0%
PSR ∧ FnE ∧ CAS ∧¬HR	39/50	78.0%

Table 8
Task-stratified downstream performance from the field-trial logs.

Task	N	TPA	CDA	TSA	HR	ASR	Successful+safe
Harvesting	11	90.9	81.8	90.9	18.2	81.8	81.8
Leaf-removal/pruning	14	85.7	85.7	92.9	0.0	78.6	71.4
Thinning	25	92.0	80.0	72.0	24.0	40.0	36.0
Overall	50	90.0	82.0	82.0	16.0	60.0	56.0

4.6. Decision accuracy results

Decision agreement is high at the component level, but substantially lower for the full decision tuple. Table 5 summarizes decision-level agreement with expert references. We distinguish *full-tuple decision agreement* (DA) from *component-wise agreement* to avoid ambiguity. Let the reference decision tuple be $\mathbf{d}_i^{\mathrm{ref}} = (\tau_i^{\mathrm{ref}}, c_i^{\mathrm{ref}}, \eta_i^{\mathrm{ref}})$ and the model output be $\mathbf{d}_i = (\tau_i, c_i, \eta_i)$, where τ is the task label, c is the cooperation flag, and η is the target identity. Full agreement is defined as

$$\mathrm{DA} \triangleq \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\mathbf{d}_i = \mathbf{d}_i^{\mathrm{ref}}\right]. \tag{34}$$

In addition, we report component-wise agreement rates: task prediction agreement (TPA), cooperation decision agreement (CDA), and target selection agreement (TSA), each computed as a marginal agreement on the corresponding element of $\mathbf{d}_i$.

Decision-level agreement with the expert reference was high for task-type inference but lower for cooperation and target selection (Table 5). The TPA metric reached 0.90 (45/50), demonstrating that the inferred task type matched the GT task label in most trials. The CDA was 0.82 (41/50), indicating the correct identification of whether cooperative control was required. The TSA was also 0.82 (41/50), reflecting agreement between the selected target and the GT target definition. The HR was 0.16 (8/50), indicating that a nonnegligible subset of trials produced reasoning outputs that were inconsistent with the vision-derived FACTs. At the same time, the gap between the full-tuple DA (0.58) and the component-wise agreement rates suggests that disagreement with the reference policy does not always imply an invalid or unusable decision, particularly when multiple targets or task strategies remain acceptable in the same canopy state.

The task-stratified results suggest that downstream performance is most vulnerable in relation- and target-sensitive cases (Table 8). Thinning maintained high task-level agreement (92.0%), but showed lower target selection agreement (72.0%), higher hallucination rate (24.0%), lower ASR (40.0%), and lower successful-and-safe execution rate (36.0%) than harvesting and leaf-removal/pruning. This pattern does not prove that all thinning failures were caused solely by SGG errors, because physical execution difficulty and target ambiguity also contribute to failure. However, it indicates that tasks requiring finer target discrimination and occlusion reasoning are more sensitive to imperfect relational evidence in the FACT set.

4.7. Relation-error attribution results

The planned cooperation decision agreed with the ground-truth requirement in 41 of the 50 field trials (0.82), and each of the nine disagreements is attributable to an identifiable error source (Table 9). The nine flips comprise six missed cooperations and three unnecessary cooperative manipulations. All three unnecessary cooperative *push* decisions occurred in hallucination-free plans, so each is directly attributable to a false `occluded` FACT admitted by the SGG module. These three trials are concrete instances in which an erroneous relational prediction induced an unnecessary environmental interaction. Of the six missed cooperations, four occurred in plans that were both hallucination-free and action-sequence-consistent, implying that the required `occluded` FACT was absent from the evidence set; the remaining two coincided with hallucinated content and are attributed to the reasoning channel. In total, seven of the nine cooperation flips (77.8%) are attributable to erroneous relational FACTs and two to hallucinated reasoning. Flips also showed no association with the hallucination flag (2/8 in hallucinated versus 7/42 in fully grounded plans; Fisher's exact test, $p = 0.62$): cooperation errors arise from the relational *content* of the FACT set, not from ungrounded generation.

The attribution also quantifies the decision-level benefit of higher-quality relational perception. Because the seven perception-attributable flips are each tied to an individual false or missing `occluded` FACT, correcting these relational errors would remove the perception channel entirely, bounding cooperation agreement at 48/50 (0.96) versus the observed 41/50 (0.82); the two residual errors lie in the reasoning channel, which is the channel that the verification-and-repair layer targets. Task-wise, flips concentrated in thinning (5/25) relative to harvesting (2/11) and leaf-removal/pruning (2/14), consistent with the task-stratified pattern in Table 8 and with the qualitative limitation case in Fig. 10(b), where a relation error propagates to an incorrect cooperation decision.

4.8. Action execution results

Action execution was evaluated after the reasoning and validation stages to determine whether the symbolic plan could be physically realized by the dual-arm robot in the greenhouse. This metric therefore captures real-world closed-loop performance rather than offline reasoning correctness alone.

Physical execution remains the main bottleneck of the proposed framework, even when plan-level validity is relatively high. Table 5 presents the physical action execution results. Although the raw ASR was 0.60, this value alone does not fully characterize the execution behavior of the system because the framework was designed to reject invalid or unsafe plans before or during execution. In the proposed framework, ASR is logged as 1 only when the robot completes the planned sequence without aborting and achieves the intended manipulation effect; accordingly, all aborted trials are counted as ASR failures, even when the abort itself reflects conservative safety enforcement rather than uncontrolled execution. Among the trials with invalid plans, a substantial portion was safely aborted before physical execution,

Table 9
Attribution of the 50 field-trial cooperation decisions. Perception-channel attribution requires a hallucination-free plan (UCM) or a hallucination-free, action-sequence-consistent plan (MC), so that the flip is implied by the relational FACT content under the FACT-closed planner. MC: missed cooperation; UCM: unnecessary cooperative manipulation.

Cooperation outcome	*n*	Missing `occluded` FACT	False `occluded` FACT	Hallucinated reasoning
Agreement	41	–	–	–
Missed cooperation (MC)	6	4	–	2
Unnecessary cooperation (UCM)	3	–	3	0
All cooperation flips	9	4	3	2

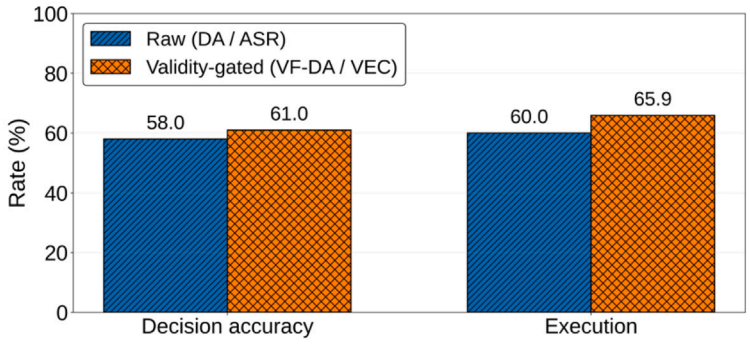

Fig. 9. Comparison of raw and validity-gated outcomes. Left: Decision accuracy before and after applying logical validity filtering. Right: Action success rate compared with validity-enforced execution consistency. Validity gating excludes logically inconsistent or physically infeasible plans before execution, improving the reliability of deployed autonomy.

Table 10
Conditional comparison between rule-based reference decisions and grounded language-based reasoning outcomes.

Evaluation	Result
Trials with same action (rule-based = proposed)	38/50
Decision accuracy (TPA/CDA/TSA)	0.81
Execution outcome (ASR/successful-and-safe execution rate)	0.62
Trials with different action (rule-based ≠ proposed)	12/50
Decision accuracy	Not applicable (rule disagreement)
Execution outcome	Not directly comparable
Logical validity (PSR/FnE/CAS)	0.92 (valid alternative decisions)

preventing potential collisions with rigid plant structures. Therefore, lower ASR and successful-and-safe execution rates partially reflect conservative safety enforcement rather than uncontrolled execution failure. The average execution time was 9.8 s for cooperative trials and 15.2 s for noncooperative trials in the logged episodes. This counterintuitive ordering reflects the fact that several cooperative trials were short access-clearing motions, whereas some noncooperative trials required longer target approaches in cluttered workspaces; the values should therefore be interpreted as descriptive field-log statistics rather than a general claim that cooperative execution is faster.

Plan-level logical validity alone is insufficient to guarantee robust physical manipulation under severe occlusion. These execution-level results motivate the discussion in Section 5 on tactile feedback and execution-aware autonomy. Fig. 9 visualizes the effect of validity gating on the DA and execution safety.

Most execution failures were blocked before unsafe physical interaction occurred, indicating that the validity gate functioned conservatively as intended. Among the 20 aborted trials, 12 were caused by invalid logical plans (schema or FACT inconsistency), and the remaining 8 resulted from physical infeasibility under kinematic or safety constraints. Under the UEPR definition in , all 12 invalid logical plans were blocked before execution (UEPR = 1.00), while the remaining 8 aborts reflect feasibility or safety screening at the motion layer. These results indicate that the proposed validity and feasibility gating effectively prevent unsafe executions, and that the lower raw action success rate should be interpreted partly as a consequence of conservative autonomy rather than purely as evidence of execution instability.

Table 11
Execution outcome breakdown ($N = 50$).

Outcome category	Count
Successful and safe execution	28
Successful, but safety risk detected	2
Aborted due to invalid logical plan	12
Aborted due to physical infeasibility	8

Agreement with the rule-based reference policy is informative only when both methods select the same action, whereas logically valid alternative decisions require a different interpretation. Table 10 summarizes the conditional comparison protocol. When the inferred action matched the rule-based reference policy, decision agreement and execution outcomes were directly comparable. When the inferred action differed, agreement-based metrics became insufficient for interpretation; therefore, logical validity was used to assess whether the alternative decision remained fact-consistent, physically feasible, and actionable under the same scene conditions (see Tables 11 and 12).

4.9. Qualitative end-to-end pipeline examples

The qualitative cases confirm that the evaluation is closed-loop rather than offline (Fig. 10). The outputs of SGG, FACT serialization, VLM reasoning, validation, and action-primitive mapping are shown together with the final robot outcome. The successful case demonstrates

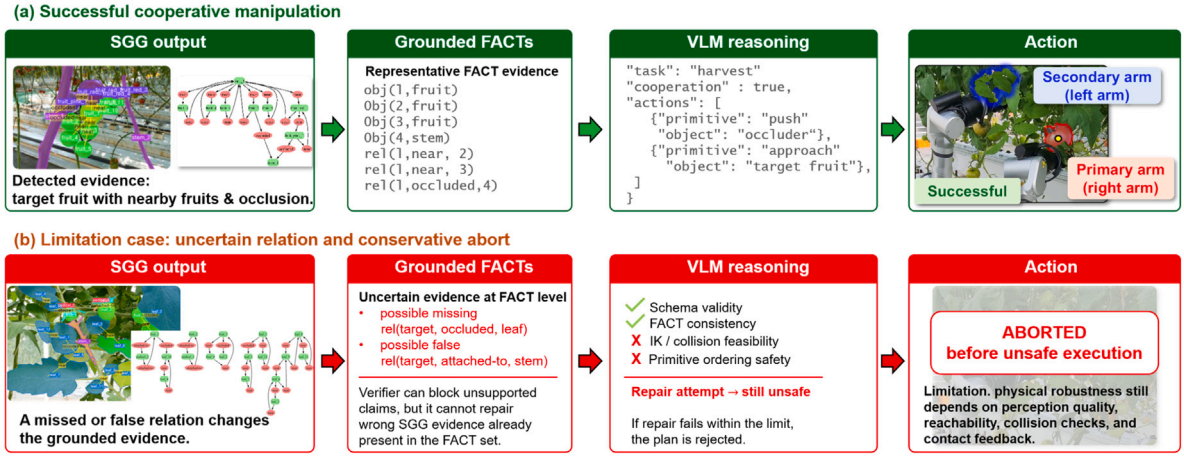

Fig. 10. Qualitative end-to-end examples of the proposed pipeline. (a) In a successful cooperative case, the SGG output is serialized into FACTs, the VLM planner generates a schema-valid and FACT-consistent plan, and the robot executes the resulting *push–approach* sequence. (b) In a limitation case, an incomplete or erroneous SGG relation changes the FACT set and affects downstream planning or feasibility checking. The verifier can block unsupported language claims and reject infeasible plans, but it does not correct erroneous relations already present in the FACT set.

that verified reasoning can be physically executed, whereas the limitation case shows that perception uncertainty or feasibility constraints can still lead to aborts.

The limitation case shows the remaining dependence on SGG quality. If a task-critical relation is missing or incorrectly predicted, the corresponding FACT set becomes incomplete or misleading. The verifier prevents the VLM from introducing unsupported claims beyond the FACT set, but it does not correct erroneous SGG evidence already present in the FACTs. Therefore, the system may select an unnecessary cooperative action, omit a useful push primitive, or abort through feasibility and safety gating. This qualitative result is consistent with the quantitative observation that SGDet relation performance is limited and that downstream performance is lower in relation- and target-sensitive cases.

5. Discussion

The results show that the proposed framework is most effective at the plan-validity level, whereas perception quality and physical execution remain the two main bottlenecks. This interpretation is important because the proposed system should not be evaluated as a generic SGG model, an end-to-end VLA policy, or a purely rule-based agricultural planner. Instead, it should be assessed as a modular reasoning-to-action framework that uses scene-graph-derived FACTs to constrain language-based planning before robot execution. Accordingly, the following discussion compares the results with three related lines of work: SGG-centered visual reasoning, grounded VLM-based robot planning, and agricultural robotic manipulation under occlusion.

The revised results clarify that the framework was evaluated beyond offline data processing. The closed-loop field trials connect perception, reasoning, validation, and physical action execution on a dual-arm robot in a greenhouse. This is important because high plan-level validity does not guarantee physical success: the ASR and abort results show that contact, reachability, and plant-structure uncertainty remain major bottlenecks after symbolic reasoning is complete. Therefore, the real-world validation supports a narrower but more deployment-relevant conclusion: FACT-grounded reasoning can produce inspectable and executable action plans, but physical robustness still depends on perception quality, feasibility checking, and manipulation capability.

5.1. Perception and SGG-dependent reasoning

The effectiveness of the proposed framework depends directly on the quality of relational grounding provided by the SGG module. The sensitivity analysis clarifies the role and limitation of the verification layer. The verifier prevents the VLM planner from referring to objects, attributes, or relations that are absent from the current FACT set. However, it does not correct the SGG module itself. If a true occlusion relation is missed by SGG, the verifier cannot recover that missing evidence. Conversely, if an incorrect relation is included in the FACT set, the verifier treats it as available visual evidence. Therefore, the framework is conditionally robust with respect to language-model hallucination given the FACT set, but it remains sensitive to the SGG model through task-critical relation quality. In the Goheung field trials, the SGG performance for grounded FACT extraction reached a relationship mR@50 of 22.1 under the SGDet setting (Table 6). This value should be interpreted cautiously. It does not indicate that the SGG module reaches the level of recent generic SGG methods on benchmark datasets; rather, it shows that relation prediction in tomato canopies remains difficult because objects are thin, overlapping, deformable, and frequently occluded. This is consistent with the broader SGG literature, where relationship prediction is more difficult than object recognition because predicates depend on object context, spatial configuration, and dataset bias (Krishna et al., 2017; Zellers et al., 2018).

For the proposed framework, the practical issue is not only whether SGG relations are correct, but whether relation errors are critical for action selection. A missing *occluded* relationship can suppress a required *push* action, whereas a false-positive *occluded* relationship can introduce unnecessary cooperative motion. Similarly, an incorrect *attached_to* or *near* relationship can change whether the system treats an object as a manipulable target or an obstacle. Thus, the SGG output acts as uncertain visual evidence even though the current FACT representation is deterministic. The verifier prevents the VLM reasoning module from inventing unsupported relations, but it cannot correct a false relation already produced by the SGG module or recover a relation that was missed before FACT construction.

This distinction explains why the reported perception and reasoning results should be interpreted together. A high plan-validity rate under imperfect SGG indicates that the verification layer can block unsupported language outputs, but it does not imply that the perception module is solved. In this sense, better SGG is both a perception improvement and an autonomy enabler: as the FACT set becomes more complete and accurate, the reasoning module can reduce conservative fallbacks and generate shorter, more goal-directed action sequences. This requirement is now addressed empirically: Section 4.7 attributes every cooperation-decision error observed in the field trials to either the relational FACT content or the reasoning channel, so that SGG

Table 12
Interpretation of the present results relative to related studies.

Related line of work	Main focus	Relevance to our results
Generic SGG and visual relationship modeling (Krishna et al., 2017; H. Li et al., 2024)	Predicting objects, attributes, and pairwise relations in images	Greenhouse SGG remains difficult under occlusion, and relation errors affect robot decisions.
Rule-based agricultural scene-graph planning (Park and Son, 2025)	Deterministic mapping from visual relations to task and cooperation decisions	Rule agreement is useful but does not cover all feasible manipulation strategies.
Grounded VLM-based robot planning (Brohan et al., 2023b)	Combining language-model reasoning with skill or affordance grounding	FACT grounding gives scene-specific evidence, but verification is still required.
End-to-end vision-language-action policies (Brohan et al., 2023a)	Learning direct mappings from vision and language to robot actions using large-scale robot data	Our modular design favors inspectability and deployment with limited field data.
Contact-rich robotic manipulation and tactile sensing (Dahiya et al., 2010; Kappassov et al., 2015)	Robust physical interaction under uncertain contact conditions	The gap between plan validity and ASR motivates force- or tactile-aware execution.

quality is assessed not only by R@K and mR@K but also by how often a relation error changes the cooperation decision.

The relation-error attribution makes this dependency explicit rather than associative. Seven of the nine cooperation flips in the field trials trace to the relational FACT content itself: all three unnecessary cooperative pushes originated from false `occluded` FACTs, and four missed cooperations from absent ones, while flips showed no association with hallucination (Section 4.7). Correct relational FACTs would thus have bounded cooperation agreement at 0.96, isolating the residual errors in the reasoning channel that the verification layer targets. This is consistent with the qualitative limitation case in Fig. 10(b).

The use of Neural Motifs as the SGG backbone is a deliberate design choice rather than a claim of perceptual novelty. The reasoning-to-action interface consumes only detected objects, attributes, and relations, so the framework is agnostic to the SGG architecture. Within this interface, Motifs was selected because its frequency-bias formulation, recomputed from the greenhouse annotations, matches the strong predicate regularities of canopy scenes and remains trainable in the limited-data regime, and because its mature, lightweight implementation suits on-robot deployment. Recent transformer-based one-stage SGG models can be substituted without changing the interface; the relation-error attribution in Section 4.7 bounds the decision-level benefit such a substitution could provide.

5.2. Logical validity and comparison with grounded language planning

The principal finding is that grounded language-based reasoning can produce logically valid robot plans when its outputs are constrained by scene-derived FACTs, but unverified generation remains unsafe for deployment. This finding is consistent with grounded VLM-based robot planning studies such as SayCan, where language-model proposals are constrained by real-world skill affordances (Brohan et al., 2023b). The present work differs in the grounding mechanism: instead of grounding candidate skills through learned affordance values, it grounds every generated object, attribute, relation, and action argument against a closed FACT set extracted from the current greenhouse observation.

These results also clarify why the proposed framework was motivated by deployment constraints rather than by the general trend of language-model planning. The VLM reasoning module alone is not the main contribution; its outputs become useful for greenhouse manipulation only when they are constrained by scene-specific FACTs and screened by feasibility and safety checks. The observed hallucination and abort cases support this design choice: flexible reasoning must remain auditable before it can be trusted for physical interaction with dense plant structures.

This design is supported by the reported logical-validity components. This study decomposes logical validity into three binary checks: precondition satisfaction (PSR), feasibility/executability (FnE), and correct action sequence (CAS). To avoid ambiguity between plan validity and hallucination-filtered acceptance, two validity predicates are defined:

$$v_i^{\text{plan}} = \mathbf{1}\left[\text{PSR}_i \wedge \text{FnE}_i \wedge \text{CAS}_i\right], \quad v_i = \mathbf{1}\left[\text{PSR}_i \wedge \text{FnE}_i \wedge \text{CAS}_i \wedge \neg\,\text{HR}_i\right]. \tag{35}$$

Here, v_i^{plan} measures whether the generated plan is structurally coherent, FACT-supported, and physically executable, whereas v_i denotes a strict accepted plan after excluding hallucination-flagged trials. In the field trials, the component rates were PSR = 90.0%, FnE = 82.0%, and CAS = 90.0%, yielding V_{plan} = 82.0% and V_{strict} = 78.0% (Table 13). The lower FnE compared with PSR indicates that syntactically well-formed plans can still contain infeasible action arguments or fail physical constraints; therefore, schema validation alone is insufficient.

The hallucination rate further supports the need for explicit verification. The observed HR of 16.0% indicates that a nonnegligible portion of generated outputs attempted to use unsupported or inconsistent scene information. This is consistent with the broader NLG literature, which identifies hallucination as a persistent failure mode of neural generation (Ji et al., 2023). Therefore, the results do not support a claim that FACT grounding eliminates hallucination. Instead, they support a narrower and more defensible claim: FACT grounding plus validation and repair makes hallucination observable, rejectable, and preventable from propagating directly to robot execution.

The validation-stage diagnostic analysis indicates that logical validity is affected by both action feasibility and evidence consistency. The difference between PSR ∧ FnE ∧ CAS and PSR ∧ FnE ∧ CAS ∧¬HR shows that some otherwise executable plans still contained hallucination-flagged visual claims. Thus, FACT-consistency checking acts as a conservative filter before a symbolic plan is considered logically valid. However, this analysis should not be interpreted as a full counterfactual ablation of the repair loop. Such an ablation requires raw pre-repair generations and violation messages, which should be archived in future deployments.

5.3. Computational considerations

The VLM introduces a noticeable but bounded high-level planning delay rather than a real-time control burden. During field operation, VLM reasoning required approximately 3–14 s per decision from prompt submission to the final accepted symbolic plan, including any verification-triggered repair attempts. The lower end corresponds

primarily to first-pass valid plans, whereas longer generation or repair increased the delay toward the upper end.

The remaining FACT construction, schema validation, FACT-consistency checking, and reasoning-to-action conversion stages are deterministic operations and do not introduce another learned inference model. Because the VLM is invoked before motion execution rather than during the low-level control cycle, the observed latency affects task-level responsiveness but not arm-control stability. Detailed per-call latency distributions, peak memory, and separate timings for generation and repair were not retained; therefore, the study reports the observed operational range rather than a full computational benchmark.

Structurally, the VLM is invoked once per decision step, plus up to three verification-triggered repair calls in the worst case, before motion execution, so its latency lies outside the real-time control loop and is incurred once per trial rather than per control cycle. Recording per-module latency and peak memory is planned for future deployments (Section 5.6).

5.4. Decision accuracy and comparison with rule-based planning

The comparison with the rule-based reference should be interpreted as an assessment of validity-aware flexibility rather than unconditional superiority. The rule-based system in Park and Son (2025) is a deterministic reference policy for relational agricultural manipulation, but it is not a unique ground truth for every canopy state. Dense greenhouse scenes may admit multiple acceptable decisions; for example, the robot may select a different ripe fruit, remove a different flexible occluder first, or choose a noncooperative approach when cooperative manipulation is feasible but unnecessary. Consequently, full-tuple agreement is informative but incomplete: a plan that differs from the reference can remain operationally valid if it is FACT-consistent, feasible, and executable. The main empirical claim is therefore that the proposed framework supports validity-aware alternative decisions, not that it universally outperforms the rule-based baseline.

The current results demonstrate this limitation. The full-tuple DA was 58.0%, whereas the component-wise agreement rates were higher: TPA reached 90.0%, CDA reached 82.0%, and TSA reached 82.0%. This gap indicates that many disagreements were not complete reasoning failures but partial deviations in target or cooperation selection. This is why the conditional analysis in Table 10 is necessary. When the proposed method selected the same action as the rule-based policy, direct agreement-based comparison was meaningful. When it selected a different action, the decision had to be evaluated using logical validity and feasibility rather than rule agreement alone.

To quantify this point, this study uses validity-aware decision and execution metrics. Building on the strict accepted-validity indicator v_i, two validity-aware indicators are defined as

$$\text{VF-DA} = \frac{\sum_{i=1}^{N} v_i\, a_i}{\sum_{i=1}^{N} v_i}, \tag{36}$$

$$\text{VEC} = \frac{\sum_{i=1}^{N} v_i\, e_i}{\sum_{i=1}^{N} v_i}, \tag{37}$$

where a_i denotes the decision-agreement indicator, and e_i represents the execution-success indicator. VF-DA and VEC are not introduced to inflate performance; they evaluate a deployable operating mode in which invalid plans are rejected before execution and either human intervention or replanning is required. In the current results, VF-DA increased from DA = 58.0% to VF-DA = 61.0%, and execution success increased from ASR = 60.0% to VEC = 65.9% (Table 13).

The autonomy-realization indicator further shows why a strict rule-agreement metric can underestimate acceptable behavior. Let the reference decision tuple be $\mathbf{d}_i^{\text{ref}} = (\tau_i^{\text{ref}}, c_i^{\text{ref}}, \eta_i^{\text{ref}})$ and the model output be $\mathbf{d}_i = (\tau_i, c_i, \eta_i)$. Full agreement is defined as $a_i = \mathbf{1}[\mathbf{d}_i = \mathbf{d}_i^{\text{ref}}]$. The autonomy-realization indicator (ARI) is defined as

$$\text{ARI} = \frac{\sum_{i=1}^{N} v_i\, e_i\, (1 - a_i)}{\sum_{i=1}^{N} v_i\, e_i}, \tag{38}$$

which measures the fraction of successful and valid executions that deviate from the reference rules. In the trials, ARI = 29.6% (Table 13). This indicates that nearly one-third of valid and successful executions would be penalized under DA-only evaluation, even though they were fact-consistent, feasible, and physically successful. The result supports the proposed evaluation framework, but it does not imply that the VLM planner is universally superior to the rule-based planner. The more defensible conclusion is that rule-based agreement and validity-aware execution capture different aspects of deployable autonomy.

These results should be interpreted as a comparison with a rule-based reference policy rather than as a claim of universal superiority over all state-of-the-art planning methods. The full decision-tuple agreement of 58.0% indicates that the proposed framework does not simply imitate the rule-based planner. However, the component-wise agreement rates for task selection, cooperation decision, and target selection were higher, and several rule-disagreement cases remained logically valid and executable. Therefore, the empirical advantage demonstrated here is validity-aware flexibility: the framework can produce FACT-consistent alternative plans while preserving explicit verification before execution.

5.5. Action execution and comparison with physical manipulation studies

The execution results show that plan-level correctness does not guarantee physical success in contact-rich greenhouse manipulation. Although the raw ASR was 60.0%, this value includes aborted trials caused by invalid logical plans and physical infeasibility under kinematic or safety constraints. Among the 20 aborted trials, 12 were caused by invalid logical plans and 8 resulted from physical infeasibility. Thus, the execution results reflect both the limitations of the reasoning pipeline and the conservative safety policy required for real plant interaction.

This finding is consistent with contact-rich manipulation studies emphasizing that visual information alone is insufficient for robust interaction with deformable or cluttered objects. In the present system, rigidity is represented as a planning-oriented visual/category attribute, but actual physical interaction depends on mechanical stiffness, contact state, entanglement, and local compliance. Tactile-sensing studies identify contact feedback as an important mechanism for safe and dexterous manipulation when visual information is insufficient (Dahiya et al., 2010; Kappassov et al., 2015). The gap between V_{strict} = 78.0% and ASR = 60.0% therefore identifies the next bottleneck: not only better reasoning, but also execution-aware sensing and control.

A practical implication is that future systems should integrate the current FACT-grounded reasoning layer with tactile or force-aware execution. Tactile signals could estimate stiffness, modulate push distance and velocity, detect unexpected resistance, and trigger replanning when contact conditions violate the expected action model. Simulation-to-real learning can also complement this direction because diverse contact interactions are costly to collect in greenhouses, whereas simulation can generate interaction data under controlled parameter variations (Tobin et al., 2017). The proposed reasoning-to-action interface can serve as a structured scaffold for such future learning: high-level plans remain interpretable and verifiable, while low-level controllers adapt primitives during contact.

5.6. Limitations and future work

The comparative interpretation above clarifies the limits of the present evidence. First, the proposed framework does not claim state-of-

Table 13
Validity-aware reinterpretation of the Goheung field trials ($N = 50$). V_{plan}: PSR ∧ FnE ∧ CAS; V_{strict}: $V_{plan} \wedge \neg$HR (Eq. (35)); DA: full agreement with the reference decision tuple; VF-DA and VEC are defined in Eqs. (36) and (37), respectively; ARI is defined in Eq. (38).

	PSR	FnE	CAS	V_{plan}	V_{strict}	DA	VF-DA	ASR	VEC	ARI	$P(v \wedge e)$
Rate (%)	90.0	82.0	90.0	82.0	78.0	58.0	61.0	60.0	65.9	29.6	54.0

the-art generic SGG performance. The ontology is restricted to tomato-greenhouse objects, attributes, and relations that are directly relevant to harvesting, leaf-removal/pruning, thinning, and occlusion management. Extension to other crops, tools, and crop-management tasks will require ontology expansion, additional scene-graph supervision, and new field validation.

Second, the current reasoning module operates on deterministic FACTs and does not propagate calibrated SGG uncertainty into planning. As a result, the verifier can reject VLM-generated claims that are absent from the current FACT set, but it cannot recover a true relation missed by SGG or correct an incorrect relation already converted into a FACT. This limitation is especially important for *occluded*, *attached_to*, and *near* predicates because they directly affect target selection, cooperation decisions, and the choice between direct *approach* and cooperative *push–approach* sequences. Future work should attach calibrated confidence scores to relation FACTs and trigger re-observation, human confirmation, or conservative replanning when task-critical relations are uncertain.

Third, the action interface is intentionally limited to *approach* and *push* primitives. This improves interpretability and safety, but it restricts flexibility in highly cluttered scenes where grasping, cutting, repositioning, or multi-step clearing may be required. Fourth, the present study does not train an end-to-end VLA policy. Integrating VLA policies may improve generality, but it would require larger robot-action datasets, simulation-to-real validation, and explicit safety certification beyond the scope of the current field study.

Fifth, this revision reports a post-hoc validation-stage diagnostic rather than a complete counterfactual ablation of the repair loop. The original field logs contain final validation outcomes, but not all raw pre-repair generations, violation messages, retry traces, and SGG confidence scores needed to replay the pipeline with individual components disabled. Similarly, the study does not provide a controlled comparison among multiple SGG backbones, grounded planning methods, or end-to-end VLA policies under the same greenhouse robot setting. Future deployments should archive all intermediate generations, validation failures, repair attempts, module-wise latency, peak memory, and SGG confidence scores to enable controlled ablations and direct model-level comparisons. For the same reason, the per-trial serialized FACT sets and prompts were not retained, so the relation-error attribution in Section 4.7 is deductive rather than interventional; future deployments will archive the complete FACT sets and prompts to enable counterfactual FACT editing and oracle re-planning for interventional validation of the attribution.

Finally, physical execution remains the main bottleneck. The current framework can verify symbolic plans before execution, but robust interaction with deformable plants also requires contact-aware sensing and control. Future work should integrate tactile or force feedback, confidence-aware FACT propagation, and a richer but still verifiable primitive library for contact-rich manipulation in dense canopies.

6. Conclusion

This study demonstrates that scene-graph-derived FACTs can provide an auditable bridge between relational perception and action planning for multipurpose greenhouse robots. By constraining language-based reasoning to explicit object, attribute, and relationship evidence and screening candidate plans for schema compliance, FACT consistency, and execution feasibility, the framework supports task and cooperative action selection while preventing unsupported plans from reaching the robot. The principal contribution is therefore the verified SGG-to-action interface, rather than any individual perception, language, or control component.

The closed-loop field evaluation also indicates that logically valid planning and successful physical execution are distinct requirements. Explicit grounding and validity gates make decisions inspectable and enable the conservative rejection of unsupported or infeasible plans, but they do not correct erroneous scene-graph evidence or resolve contact uncertainty during manipulation. The central take-home message is that structured grounding and explicit verification provide a practical foundation for auditable agricultural robot autonomy, while further improvements depend on uncertainty-aware perception and more robust physical interaction.

CRediT authorship contribution statement

Yonghyun Park: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Changjo Kim:** Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Gangmin Kim:** Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Hyoung Il Son:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of generative AI use

During the preparation of this manuscript, generative language tools were used for limited language refinement. All technical content, experimental design, and conclusions were developed and verified by the authors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-02216206); by a Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIR) (RS-2024-00406796, HRD Program for Industrial Innovation); and by the Regional Innovation System and Education (RISE) Glocal University 30 Program through the Gwangju RISE Center, funded by the Ministry of Education (MOE) and the Gwangju Metropolitan City, Republic of Korea (2026-RISE(Glocal University 30)-05-011)

Code and data availability

The annotation and preprocessing code used to construct the scene-graph annotations and convert them into the SGG/FACT-compatible format is publicly available at https://github.com/parkoh25/SGG_VLM. The raw RGB-D sequences and full field-trial logs are not publicly available because they were collected in a third-party commercial greenhouse and may contain facility layout, crop production status, robot deployment locations, timestamps, and site-specific operational information.

References

Abbasi, R., Martinez, P., Ahmad, R., 2022. The digitization of agricultural industry—A systematic literature review on agriculture 4.0. Smart Agric. Technol. 2, 100042. http://dx.doi.org/10.1016/j.atech.2022.100042.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Gonzalez Arenas, M., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.W.E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., Zitkovich, B., 2023a. RT-2: Vision-language-action models transfer web knowledge to robotic control. http://dx.doi.org/10.48550/arXiv.2307.15818, arXiv:2307.15818.

Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al., 2023b. Do as i can, not as i say: Grounding language in robotic affordances. In: Conference on Robot Learning. Pmlr, pp. 287–318. http://dx.doi.org/10.48550/arXiv.2204.01691.

Dahiya, R.S., Metta, G., Valle, M., Sandini, G., 2010. Tactile sensing—From humans to humanoids. IEEE Trans. Robot. 26 (1), 1–20. http://dx.doi.org/10.1109/TRO.2009.2033627.

Hernández, H.A., Mondragón Bernal, I.F., Gonzalez Bautista, S.R., Pedraza, L.F., 2025. Reconfigurable agricultural robotics: Control strategies, communication, and applications. Comput. Electron. Agric. 234, 110161. http://dx.doi.org/10.1016/j.compag.2025.110161.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P., 2023. Survey of hallucination in natural language generation. ACM Comput. Surv. 55 (12), 1–38. http://dx.doi.org/10.1145/3571730.

Jo, Y., Park, Y., Seol, J., Pak, J., Kim, B., Kim, C., Son, H.I., 2025. A review on dual-arm manipulation in agriculture. IEEE Access 13, 150379–150399. http://dx.doi.org/10.1109/ACCESS.2025.3602396.

Kappassov, Z., Corrales, J.-A., Perdereau, V., 2015. Tactile sensing in dexterous robot hands—Review. Robot. Auton. Syst. 74, 195–220. http://dx.doi.org/10.1016/j.robot.2014.10.011.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L., 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. 123 (1), 32–73. http://dx.doi.org/10.1007/s11263-016-0981-7.

Li, X., Bao, X., Chen, Z., Li, X., 2026. REL-SF4PASS: Panoramic semantic segmentation with REL depth representation and spherical fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27676–27685. http://dx.doi.org/10.48550/arXiv.2601.16788.

Li, X., Miao, P., Li, S., Li, X., 2024. MLMG-SGG: Multilabel scene graph generation with multigrained features. IEEE Trans. Image Process. 33, 1549–1559. http://dx.doi.org/10.1109/TIP.2022.3199089.

Li, X., Wu, T., Zheng, G., Yu, Y., Li, X., 2023. Uncertainty-aware scene graph generation. Pattern Recognit. Lett. 167, 30–37. http://dx.doi.org/10.1016/j.patrec.2022.12.011.

Li, X., Zheng, G., Yu, Y., Ji, N., Li, X., 2025. Relationship-incremental scene graph generation by a divide-and-conquer pipeline with feature adapter. IEEE Trans. Image Process. 34, 678–688. http://dx.doi.org/10.1109/TIP.2024.3384096.

Li, H., Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., Shen, P., Zhao, X., Shah, S.A.A., Bennamoun, M., 2024. Scene graph generation: A comprehensive survey. Neurocomputing 566, 127052. http://dx.doi.org/10.1016/j.neucom.2023.127052.

Liu, L., Yang, F., Liu, X., Du, Y., Li, X., Li, G., Chen, D., Zhu, Z., Song, Z., 2024. A review of the current status and common key technologies for agricultural field robots. Comput. Electron. Agric. 227, 109630. http://dx.doi.org/10.1016/j.compag.2024.109630.

Pallottino, F., Violino, S., Figorilli, S., Pane, C., Aguzzi, J., Colle, G., Nerio Nemmi, E., Montaghi, A., Chatzievangelou, D., Antonucci, F., Moscovini, L., Mei, A., Costa, C., Ortenzi, L., 2025. Applications and perspectives of generative artificial intelligence in agriculture. Comput. Electron. Agric. 230, 109919. http://dx.doi.org/10.1016/j.compag.2025.109919.

Park, Y., Kim, C., Son, H.I., 2024. Fast and stable pedicel detection for robust visual servoing to harvest shaking fruits. Comput. Electron. Agric. 220, 108863. http://dx.doi.org/10.1016/j.compag.2024.108863.

Park, Y., Pak, J., Kim, C., Son, H.I., 2025. 3D point cloud-based 6D pose estimation using the pedicel morphological features of fruits and vegetables. Robot. Auton. Syst. 194, 105151. http://dx.doi.org/10.1016/j.robot.2025.105151.

Park, Y., Seol, J., Pak, J., Jo, Y., Kim, C., Son, H.I., 2023. Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector. Comput. Electron. Agric. 212, 108116. http://dx.doi.org/10.1016/j.compag.2023.108116.

Park, Y., Son, H.I., 2025. Visual scene understanding-based task planning for an efficient multipurpose agricultural robot system. IEEE Robot. Autom. Lett. 10 (12), 13241–13248. http://dx.doi.org/10.1109/LRA.2025.3629964.

Ryan, M., 2023. Labour and Skills Shortages in the Agro-Food Sector. OECD Food, Agriculture and Fisheries Papers 189, OECD Publishing, Paris, France, http://dx.doi.org/10.1787/ed758aab-en.

Sánchez-Molina, J.A., Rodríguez, F., Moreno, J.C., Sánchez-Hermosilla, J., Giménez, A., 2024. Robotics in greenhouses. Scoping review. Comput. Electron. Agric. 219, 108750. http://dx.doi.org/10.1016/j.compag.2024.108750.

Seol, J., Park, Y., Pak, J., Jo, Y., Lee, G., Kim, Y., Ju, C., Hong, A., Son, H.I., 2024. Human-centered robotic system for agricultural applications: Design, development, and field evaluation. Agriculture 14 (11), 1985. http://dx.doi.org/10.3390/agriculture14111985.

Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H., 2020. Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3716–3725. http://dx.doi.org/10.1109/CVPR42600.2020.00377.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P., 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 23–30. http://dx.doi.org/10.1109/IROS.2017.8202133.

Yang, J., He, Y., Yang, J., Yang, L.T., Gao, Y., Dai, C., 2026. Modality-aligned anchor learning based on multi-level fusion for accurate scene graph generation. Inf. Fusion 127, 103755. http://dx.doi.org/10.1016/j.inffus.2025.103755.

Zellers, R., Yatskar, M., Thomson, S., Choi, Y., 2018. Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5831–5840. http://dx.doi.org/10.1109/CVPR.2018.00611.

Zhou, H., Luo, T., Zhang, J., Liu, L., 2025. Exploring the essence of relationships for scene graph generation via causal features enhancement network. IEEE Trans. Pattern Anal. Mach. Intell. 47 (8), 6616–6630. http://dx.doi.org/10.1109/TPAMI.2025.3559995.

Zhou, H., Zhang, J., Luo, T., Yang, Y., Lei, J., 2023. Debiased scene graph generation for dual imbalance learning. IEEE Trans. Pattern Anal. Mach. Intell. 45 (4), 4274–4288. http://dx.doi.org/10.1109/TPAMI.2022.3198965.