Autonomous Orbital Assembly via Vision-Language-Action Models: A Software Framework for Robotic Manufacturing in the Arkisys Bosuns Locker

Ankur Senapati[1], Elaine Lee[2], Ojas Chaturvedi[3], Ritwik Jayaraman[3], and Ekansh Agarwal[3]
Elizabeth Kung[4] and Harsha Malshe[5]

[1] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA
[2] School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907, USA
[3] Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA
[4] Advisor, New Glenn Systems Engineering, Blue Origin, Kent, WA 98032, USA
[5] Mentor, 129 South Street, Boston, MA 02111, Merlin Labs

Abstract

Current in-space servicing, assembly, and manufacturing (ISAM) robotic systems depend on continuous teleoperation or pre-scripted motion sequences, introducing communication latency constraints, limiting mission cadence, and preventing scalable orbital manufacturing. This paper presents a platform-agnostic software framework that uses a Vision-Language Model (LLaVA) for natural language task reasoning, coupled with classical computer vision and motion planning, to enable autonomous multi-step assembly. The system accepts high-level operator commands (e.g., "tighten a screw"), reasons about the required tool, localizes the target via OpenCV-based detection and depth-buffer deprojection, and commands a robotic arm through inverse kinematics and OMPL collision-free path planning. A trade study comparing the VLA approach against an open-vocabulary object detector (OV-DINO) is presented; VLA was selected for superior task-level reasoning despite higher compute requirements. Pick-and-place is demonstrated in CoppeliaSim simulation on a UR5 arm with 4 mm end-effector positioning accuracy and 32-second cycle time from a natural language command. Three interchangeable tool heads (gripper, self-piercing riveter, and screwdriver) connect via a proprietary twist-lock mechanism, forming a complete assembly capability chain. The payload is designed for the Arkisys Bosuns Locker hosted payload interface ($15.75 \times 15.75 \times 35.45$ in, 400 kg, 300 W), with significant mass and power margins. This work represents the first application of vision-language-action reasoning to ISAM assembly task planning.

Nomenclature

BBox = Bounding Box
CLIP = Contrastive Language-Image Pre-training
CONOPS = Concept of Operations
C3 = COSMIC Capstone Challenge
DINO = Self-Distillation with No Labels
DLS = Damped Least Squares
DOF = Degrees of Freedom
HSV = Hue-Saturation-Value (color space)
IK = Inverse Kinematics
ISAM = In-Space Servicing, Assembly, and Manufacturing
LEO = Low Earth Orbit
NDE = Non-Destructive Evaluation
NL = Natural Language
OMPL = Open Motion Planning Library
ORU = Orbital Replaceable Unit
OV-DINO = Open-Vocabulary DINO
PDR = Preliminary Design Review
SPR = Self-Piercing Riveting
TRL = Technology Readiness Level
VLA = Vision-Language-Action
VLM = Vision-Language Model
ZMQ = ZeroMQ (Communication Protocol)

I. Introduction

The 2022 National In-Space Servicing, Assembly, and Manufacturing (ISAM) Implementation Plan identifies autonomous on-orbit assembly as a critical enabler for next-generation space infrastructure [1]. Current approaches to orbital robotic operations, from the International Space Station's Canadarm2 to commercial servicing vehicles, rely on either continuous teleoperation or pre-scripted motion sequences. These paradigms are fundamentally limited by com-

munication latency, the cost of extravehicular activity (6+ hours per EVA session), and the inability to adapt to unanticipated scenarios without ground operator intervention [10].

Terrestrial robotics is undergoing a paradigm shift driven by foundation models. Vision-Language-Action (VLA) architectures such as RT-2 [11] and OpenVLA [13] demonstrate that large pretrained models can map natural language commands and visual observations directly to robot actions. Separately, Vision-Language Models (VLMs) such as LLaVA [14] enable multimodal reasoning about scenes from images and text. These advances have not yet been applied to ISAM, where the combination of constrained workspaces, communication delays, and the need for multi-step task sequencing makes autonomous reasoning particularly valuable.

The Consortium for Space Mobility and ISAM Capabilities (COSMIC) established a challenge to advance ISAM concepts through university design competitions [2,5]. Teams are challenged to design a payload for the Arkisys Bosuns Locker hosted payload interface that demonstrates a chain of three or more discrete operations important for orbital manufacturing or assembly [4]. Our team (Astrobotics, Purdue University) addresses this challenge with a software-first approach: rather than developing novel manipulator hardware, we develop an intelligent software layer that proposes any robotic arm to execute assembly tasks from natural language supervision. This paper makes three contributions:

1. A VLA pipeline, combining LLaVA for task reasoning with OpenCV perception and IK/OMPL motion planning; demonstrated for autonomous pick-and-place in simulation with 4 mm accuracy from natural language commands.
2. A trade study comparing VLA (LLaVA) against open-vocabulary object detection (OV-DINO [16]) for ISAM applications.
3. A conceptual payload design packaging three chained assembly operations (pick-and-place, self-piercing riveting, screwing) inside the Bosuns Locker, with custom-designed tooling and a proprietary twist-lock mechanism.

The remainder of this paper is organized as follows. Section II reviews related work in ISAM robotics and foundation models. Section III presents the system design including the software pipeline, hardware architecture, and tooling. Section IV describes the concept of operations. Section V presents trade studies. Section VI reports simulation results. Sections VII and VIII address risk assessment and connection to COSMIC high-value missions. Section IX discusses lessons learned, Section X outlines the path to Preliminary Design Review (PDR), and Section XI concludes.

II. Background and Related Work

A. ISAM Missions and Robotic Systems

On-orbit robotic assembly has been demonstrated in progressively more capable missions. DARPA's Orbital Express (2007) demonstrated autonomous rendezvous, docking, and component transfer between two spacecraft, establishing the feasibility of autonomous servicing [6]. Northrop Grumman's Mission Extension Vehicle (MEV-1, MEV-2) successfully docked with aging commercial satellites to extend their operational lives, demonstrating commercial viability [7]. NASA's OSAM-1 (formerly Restore-L) aims to demonstrate robotic servicing of a satellite not designed for servicing [8]. On the ISS, the Canadarm2/Dextre system performs routine assembly and maintenance tasks through teleoperation from the ground or from within the station [10].

A common thread across these systems is their dependence on either human-in-the-loop control or preprogrammed trajectory sequences. No operational on-orbit system currently uses learned perception-action models for autonomous assembly task execution. This constrains mission cadence, requires continuous ground support, and prevents adaptation to novel or unanticipated assembly scenarios.

B. Foundation Models for Robotics

The application of large pretrained models to robotic control has accelerated rapidly. RT-2 [11] demonstrated that a 55-billion-parameter vision-language model, fine-tuned on robot demonstrations, can directly output motor commands from images and language instructions. Octo [12] provides an open-source generalist robot policy based on transformer architectures. OpenVLA [13] offers a 7-billion-parameter open-source VLA that maps vision and language inputs to robot joint actions end-to-end.

LLaVA [14] takes a different approach: rather than directly outputting robot actions, it provides multimodal visual reasoning, answering questions about images and making decisions based on visual content. This reasoning capability can be integrated into a robotic pipeline as a task-level planner, with lower-level perception and control handled by classical methods.

Table 1: Bosuns Locker compliance matrix.

Param.	Spec	Design	Margin
Vol. (in^3)	15.75×15.75×35.45	14×14×32	Positive
Mass (kg)	400	∼80	320 kg
Sust. (W)	300	∼180	120 (40%)
Peak (W)	1000	∼265	735 (74%)

C. Open-Vocabulary Object Detection

Self-supervised vision models provide an alternative perception pathway. DINOv2 [15] produces rich patch-level features without supervised training. OV-DINO [16] extends this approach to open-vocabulary object detection, accepting free-form text queries and returning bounding boxes around matching objects. These models offer lower computational requirements than full VLA architectures while providing language-conditioned spatial localization.

D. Gap in Current Work

No prior work has applied VLA-class reasoning or open-vocabulary detection to ISAM assembly tasks. No trade study exists comparing these approaches under the compute, power, and volume constraints of space-rated hardware. This paper addresses both gaps by implementing and evaluating both approaches in a simulated orbital assembly scenario.

III. System Design

A. Host Vehicle: Arkisys Bosuns Locker

The Arkisys Bosuns Locker is a hosted payload interface on the Arkisys Port Module, providing a standardized volume for ISAM payloads [2]. Table 1 summarizes the Bosuns Locker specifications and the compliance of our conceptual design.

The 6-DOF arm, three tool heads, the toolbox housing, and the compute unit package within the allocated volume with positive margin. The payload mass estimate of approximately 80 kg leaves a 320 kg margin, and sustained power draw of approximately 180 W provides a 40% margin against the 300 W allocation.

B. 6-DOF Robotic Arm

The flight design employs a 6-DOF robotic arm architecture, selected for its balance of dexterity and mechanical simplicity. This configuration, integrated with a locomotion architecture featuring identical end effectors at each terminus, provides the necessary maneuverability for obstacle avoidance within the constrained Bosun's Locker volume. By allowing the arm to reposition its base rather than relying on a fixed mounting point, the system effectively maximizes its reachable workspace.

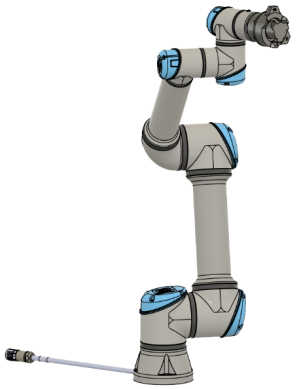

Figure 1: CAD rendering of the 6-DOF robotic arm.

Each end effector integrates a camera with LED illumination for workspace perception, pogo pins for electrical connection to interchangeable tool heads, and a proprietary twist-lock mechanism for secure mechanical coupling. The arm stows in a folded configuration within the Bosuns Locker volume and unfolds upon activation.

For the ground testbed, a Universal Robots UR5 6-DOF arm operating in CoppeliaSim [19] serves as a proxy for the flight robotic arm. The UR5 provides a representative kinematic chain for validating the software pipeline while the flight arm design matures in parallel.

C. Interchangeable Tooling Suite

Three custom-designed tool heads provide the three assembly operations:

Claw Assembly (pick-and-place): A parallel-jaw gripper for grasping and transporting structural components. Mass: 1.087 kg (38.35 oz); volume: 394.9 cm^3 (24.09 in^3).

Riveter Assembly (self-piercing riveting): A low-force SPR tool that joins structural components without pre-drilled holes. Mass: 0.502 kg (17.72 oz); volume: 183.8 cm^3 (11.22 in^3). SPR was selected over welding and adhesive bonding (see Section V) for its low thermal signature, zero debris generation, and vibration resistance.

Electric Screwdriver Tooling (reversible fastening): A rotary tool for automated screw insertion and torque application. Mass: 0.822 kg (29.01 oz); volume: 247.1 cm^3 (15.08 in^3). Provides reversible

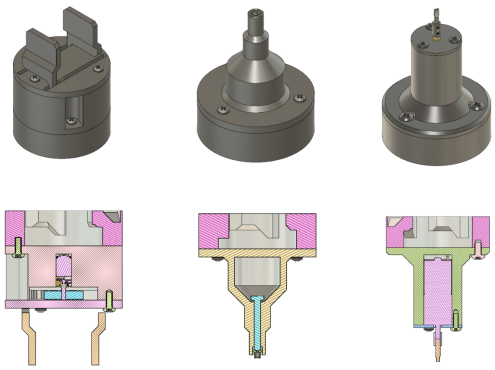

Figure 2: Three interchangeable tool heads: claw assembly (left), riveter (center), and electric screwdriver (right).

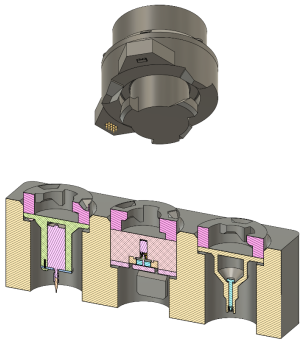

Figure 3: End effector with twist-lock mechanism (top) and toolbox cross-section showing storage bays for the three tool heads (bottom).

fastening for modular, serviceable structures.

Twist-Lock Mechanism: Tool heads attach to the end effector via a proprietary twist-lock mechanism. To minimize kinematic complexity during tool changeovers, the framework employs a **Universal Attach Orientation**. Regardless of the specific tool geometry, the end-effector adopts a fixed world-frame orientation derived from the screwdriver's attach dummy. This ensures consistent mechanical engagement and pogo-pin alignment across the entire suite without requiring tool-specific approach trajectories. Simultaneously, pogo pins on the side of the interface establish electrical connectivity for power and data. Three limit switches confirm that the tool head is fully seated and locked before the system proceeds with operations. This mechanism eliminates loose fasteners during tool change, a critical consideration for microgravity operations where dropped components become debris.

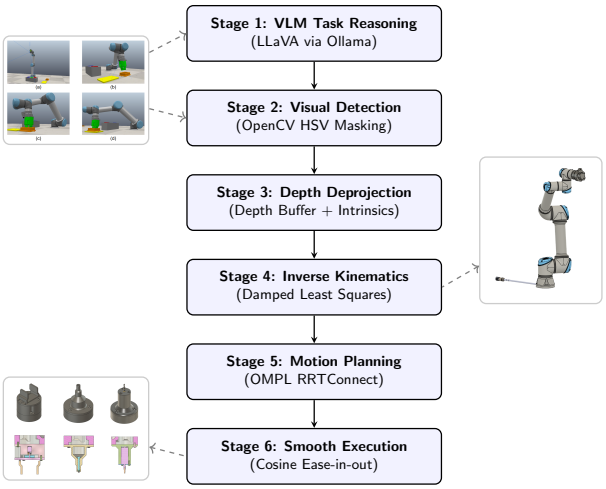

Figure 4: The 6-stage Vision-Language-Action software pipeline. Visual inputs (left) are processed into semantic targets, while planned trajectories drive the physical or simulated manipulator and tool suite (right).

D. Software Architecture: The VLA Pipeline

The core contribution of this work is the software pipeline that converts a natural language command into autonomous arm motion. The pipeline comprises six stages, illustrated in Fig. 4.

Stage 1: VLM Task Reasoning. The operator issues a natural language command (e.g., "I need to tighten a screw"). LLaVA [14], running locally via the Ollama framework, receives the command along with a camera image of the workspace. A structured prompt maps visual objects to tool semantics (e.g., colored blocks representing tools). LLaVA returns a tool selection decision, which is parsed via regex with robust multi-match handling: if the reply starts with a color word, it is trusted directly; otherwise, whole-word frequency counting with last-mentioned tiebreaking selects the answer. On an Apple M3/M4 MacBook Pro, LLaVA inference completes in 3–5 seconds.

Stage 2: Visual Detection. OpenCV converts the camera frame to HSV color space. Per-target HSV masks isolate the selected object. The largest contour by area is identified, and its centroid pixel coordinates are extracted. HSV thresholds were tuned to discriminate target objects from visually similar elements in the scene (e.g., separating deep-blue blocks from the UR5's pastel-cyan joint caps by enforcing a saturation floor of 150).

Stage 3: Depth-Based Deprojection. The vision sensor's depth buffer provides per-pixel depth. With camera intrinsics (FOV $= 70°$, resolution 512×512, focal length $f_{\text{pix}} = \text{RES}/\tan(\text{FOV}/2)$), the pixel is back-projected to camera-frame 3D coordinates,

accounting for the CoppeliaSim sensor's horizontal mirror convention. The camera's 4×3 world-frame pose matrix transforms the result to world coordinates. Deprojection error, validated against ground-truth block positions, is less than 0.1 cm.

Stage 4: Staged Inverse Kinematics. To prevent local minima divergence during high-displacement transits, the framework utilizes a **staged DLS IK approach**. The target pose is interpolated across $N = 20$ intermediate waypoints, with the solver re-seeded at each stage. This ensures a continuous joint-space path even during complex wrist re-orientations.

Stage 5: Motion Planning. The Open Motion Planning Library (OMPL) [18] RRTConnect algorithm plans a collision-free joint-space trajectory from the current configuration to the IK goal. Collision pairs include self-collision (arm vs. itself) and arm vs. floor. Per-joint directional constraints prevent winding motions. A custom state space bounds each joint to a $\pm\pi$ window centered on the start configuration, extended toward the goal.

Stage 6: Smooth Execution. Motion is driven by **Cosine Ease-in-out Interpolation**, which minimizes mechanical jerk by smoothing acceleration at the trajectory boundaries. For contact operations, the system transitions to a fine-stepping loop with a 0.008 s delay between iterations to maintain high tracking fidelity.

E. Compute Considerations for Space Deployment

LLaVA is a 7-billion-parameter model requiring GPU-class compute for inference. For space deployment, several mitigation strategies are under consideration:

1. **Model quantization:** 4-bit and 8-bit quantized variants reduce memory requirements by 4×–8× with modest accuracy loss, potentially enabling inference on edge AI accelerators.
2. **Smaller VLMs:** Models such as TinyLLaVA and MobileVLM offer reduced parameter counts optimized for edge deployment.
3. **Hybrid ground/onboard architecture:** The VLM performs task-level reasoning (infrequent, latency-tolerant), while classical computer vision handles real-time perception on edge compute. This exploits the asymmetry between the task-reasoning cadence (∼once per assembly operation) and the perception-control cadence (∼10 Hz).
4. **Ground-based inference:** For non-time-critical tasks, the VLM can run on ground infrastructure with action plans uplinked. The system's store-and-forward architecture (Section IV) naturally supports this mode.

Target space-rated hardware includes NVIDIA Jetson Orin (hardened variants), Qualcomm Snapdragon Space processors, and Xilinx Versal AI Edge FPGAs. The VLM inference call is a one-time per-task event (3–5 seconds), not a continuous real-time loop, which relaxes the compute constraint relative to end-to-end VLA architectures that require inference at control-loop rates.

IV. Concept of Operations

The mission lifecycle comprises eight stages from launch through verification.

1. Launch. The payload is stowed inside the Bosuns Locker in a folded configuration with the toolbox secured. It launches as a hosted payload on the Arkisys Port Module.

2. Deploy. The Port Module reaches the target LEO orbit. The Bosuns Locker is activated, the arm unfolds from its stowed configuration, and self-checkout diagnostics verify joint functionality, camera operation, and tool-change mechanism integrity.

3. Initialize. The compute unit boots and loads the VLM into memory. Camera systems capture workspace imagery to verify the scene. The system reports ready status to the ground via the Port Module communications link.

4. Command. The operator issues a high-level natural language task command via uplink (e.g., "Assemble strut assembly Alpha"). Command size is approximately 1 KB. No joint-level or trajectory-level commanding is required.

5. Pick and Place. The VLM interprets the command and identifies the required component. The vision system localizes the target via OpenCV detection and depth deprojection. The arm autonomously plans and executes a collision-free path to grasp, transport, and place the component at the designated assembly location.

6. Rivet. The arm approaches the toolbox, engages the riveter tool head via the twist-lock mechanism (confirmed by three limit switches), and navigates to the joining location. SPR force is applied with closed-loop force feedback to verify joint integrity.

7. Screw. The arm swaps to the screwdriver tool head via the twist-lock mechanism, locates the fastener point, inserts the fastener, and torques to specification with closed-loop force monitoring.

8. Verify and Report. The vision system inspects the completed assembly. Telemetry (joint positions, force readings, model confidence) and verifi-

Table 2: Trade study comparison between VLA (LLaVA) and OV-DINO perception architectures.

Criterion	VLA (LLaVA)	OV-DINO
Architecture	VLM + CV + depth deproj.	Open-vocab det. + IK
Task reasoning	Full NL understanding	Object queries only
Output	Tool selection	Bounding box (BBox)
Positioning	4 mm (CV + depth)	BBox tolerance
Inference	3–5 s (M3 Mac)	Edge-native, fast
Parameters	∼7B	∼300M
Space HW	Requires quantization	Edge-feasible
Status	System demonstrated	BBox tolerance error

cation images are stored onboard. Data is downlinked during the next communications window. The operator reviews the results and authorizes the next task or commands corrective action.

Operator Role. The operator functions in a supervisory capacity: issuing task-level commands, reviewing post-task verification imagery, and authorizing subsequent operations. This is consistent with the C3 "limited remote commands" definition [2]. No real-time joystick control or joint-level commanding is required. The system is designed for full autonomy after the initial command, entering a safe-hold state on communications timeout and storing telemetry for delayed downlink.

Communications. The system operates in a store-and-forward mode with periodic real-time telemetry windows. Uplink requirements are minimal: approximately 1 KB per natural language command. Downlink requirements are moderate: approximately 5–10 MB per assembly cycle, comprising compressed verification images (∼500 KB each) and telemetry logs (∼50 KB/min).

V. Trade Studies and Analysis

A. Vision Architecture: VLA (LLaVA) vs. OV-DINO

Both approaches were implemented and evaluated on the CoppeliaSim testbed. Table 2 summarizes the comparison.

Decision: VLA selected. The VLA architecture provides task-level reasoning that is essential for the supervisory autonomy concept. Given a command such as "I need to tighten a screw," LLaVA reasons about the scene to select the appropriate tool (screwdriver), whereas OV-DINO can only localize objects when given explicit queries ("find the green block"). The ability to interpret arbitrary natural language commands without pre-programming each task-tool mapping is a fundamental architectural advantage.

OV-DINO was successfully implemented using the open-source repository [16] and demonstrated object detection from natural language queries. However, positioning the arm to the bounding box centroid introduced tolerance error compared to the OpenCV + depth deprojection pipeline, which achieved 4 mm accuracy via sub-pixel contour centroiding and direct depth measurement. OV-DINO remains a viable perception backbone for future work, particularly for unstructured environments where HSV-based detection is insufficient.

The compute challenge of deploying LLaVA in space is mitigated by the hybrid architecture described in Section III.E: the VLM is invoked once per assembly task (not at control-loop rates), and classical CV handles real-time perception on edge compute.

Table 3: Joining method trade study.

Criterion	Weld	SPR	Adhesive
Heat gen.	Extreme	Low	None
Debris (μg)	High	None	Low
Power req.	Very high	Moderate	Low
Vib. resist.	High	High	Moderate
μg compat.	Poor	Good	Good
TRL (terr.)	9	9	9

B. Joining Method

Table 3 compares three joining methods for orbital assembly.

Decision: SPR selected. Self-piercing riveting generates no debris (no pre-drilling required), produces a low thermal signature compatible with the Bosuns Locker power budget, and provides vibration-resistant permanent joints. SPR is mature at TRL 9 in the automotive industry [20, 21] and requires only adaptation for microgravity application.

C. Reversible Fastening

Screwing was selected over adhesive bonding. Screwing provides reversible, serviceable joints with well-characterized torque profiles, while adhesives introduce outgassing risk in vacuum, are not reversible, and have uncertain cure times under thermal vacuum conditions. Reversible fastening is essential for modular, serviceable orbital structures aligned with ISAM goals.

Table 4: Power budget.

Subsystem	Sust. (W)	Peak (W)
Compute (VLM)	25	45
Arm actuators (7 joints)	100	150
Cameras + LEDs	10	15
End effectors (rivet/screw)	30	40
Housekeeping / comms	15	15
Total	**180**	**265**
Bosuns Locker limit	**300**	**1000**
Margin	**120 (40%)**	**735 (74%)**

D. Arm Kinematics and Resource Budgets

A 6-DOF robotic arm architecture was selected to provide the necessary dexterity for maneuvering within the constrained Bosun's Locker volume, while base locomotion enables repositioning to maximize the reachable workspace. The UR5 arm in the current testbed validates the software pipeline, with the flight hardware following this same kinematic structure.

Table 4 presents the power budget allocation across all subsystems. The compute subsystem draws an estimated 25 W sustained during VLM inference, which occurs only once per assembly task and can be duty-cycled between tasks to reduce thermal load. Arm actuator power dominates the budget at 100 W sustained across seven joints, consistent with similarly sized space-rated manipulators. The end effector allocation of 30 W sustained covers both the SPR tool (which requires brief force application) and the screwdriver (continuous rotary drive during fastener insertion). Total sustained power of 180 W provides a 40% margin against the 300 W Bosuns Locker allocation, and peak power of 265 W leaves a 74% margin against the 1000 W peak limit, ensuring that simultaneous compute inference and arm actuation remain well within the available power envelope.

VI. Results and Testbed Validation

A. Simulation Environment

The testbed consists of a Universal Robots UR5 6-DOF arm operating in CoppeliaSim (formerly V-REP) [19] with the Bullet 2.78 physics engine. An overhead vision sensor (512×512 px, 70° FOV) provides RGB images and a depth buffer. Three colored blocks (red, green, blue) at known positions represent interchangeable tools. The VLM (LLaVA via Ollama) runs on an Apple M3 MacBook Pro; motion planning uses the CoppeliaSim OMPL plugin.

Table 5: Testbed quantitative results.

Metric	Value
End-effector accuracy	4 mm
Deprojection XY error	< 2 cm
VLM inference (M3 Mac)	3–5 s
Total pick cycle time	32 s
Trajectory samples	863
OMPL planning time	< 5 s
VLM task accuracy	3/3 correct

B. Pick-and-Place Demonstration

A complete pick-and-place cycle was demonstrated from natural language command to tool acquisition and repositioning. Key quantitative results are summarized in Table 5.

The trajectory is divided into transit and contact phases. During placement, the system executes an autonomous **Contact-Driven Descent**. Utilizing the simulator's `checkDistance` API as a proxy for tactile feedback, the arm performs a rapid descent to a 10 mm clearance gap, followed by a fine-stepping phase (2 mm increments) that terminates immediately upon contact ($d \leq 1$ mm). This closed-loop approach allows for 4 mm positioning accuracy without pre-scripted height requirements. Cosine ease-in-out interpolation at approximately 3° per sub-step provides smooth motion throughout.

VLM Task Reasoning. LLaVA correctly mapped all tested natural language commands to the appropriate tool: "I need to drive a nail" → red (hammer), "I need to tighten a screw" → green (screwdriver), "I need to loosen a bolt" → blue (wrench). The structured prompt with explicit tool-color mappings and the robust regex parser (frequency counting with last-mentioned tiebreaking) achieved deterministic correct responses.

Deprojection Validation. Ground-truth block positions were validated against deprojected coordinates for all three blocks. The depth-buffer deprojection pipeline, using the sensor's actual perspective angle queried at runtime (not a hard-coded constant), achieved XY errors below 2 cm across the workspace.

C. OV-DINO Evaluation

OV-DINO [16] was evaluated as an alternative perception backbone. The model successfully produced bounding boxes from natural language queries (e.g., "red block") and the arm navigated to bounding box coordinates via IK. The bounding box centroid provided sufficient accuracy to bring the end effector near the target, at which point a centering refinement could be applied. However, the bounding box tolerance introduced positioning error relative to the sub-

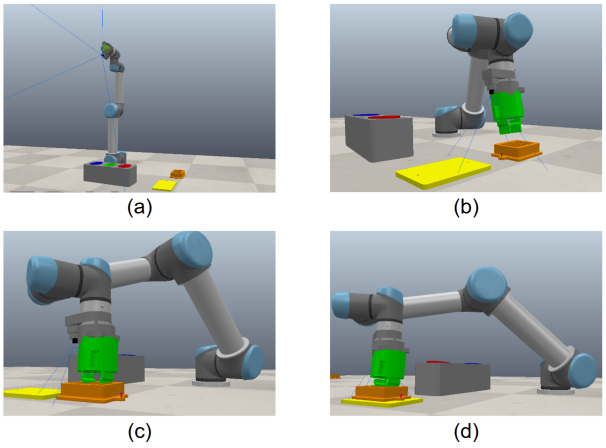

Figure 5: Pick-and-place sequence in CoppeliaSim: (a) home position, (b) OMPL path to above block, (c) descent and grasp, (d) lift and rotate to delivery position.

pixel centroiding of the OpenCV + depth pipeline. OV-DINO's lower compute requirements (∼300M parameters vs. LLaVA's ∼7B) make it attractive for edge deployment, and it remains under consideration as a complementary perception layer for unstructured environments.

VII. Risk Assessment

Table 6 identifies the top five risks, their assessed likelihood and impact, and planned mitigations.

Microgravity-specific considerations include: riveting reaction forces are absorbed through the robotic arm attachment points to the workspace structure; SPR generates no chips or debris (no drilling involved); screw operations use magnetic retention on fastener heads to prevent loss; and the twist-lock tool change mechanism eliminates loose fasteners.

VIII. Connection to COSMIC High-Value Missions

The autonomous assembly capability demonstrated in this work directly advances four of COSMIC's prioritized ISAM use cases [3]:

Use Case #6: Autonomous Payload Swap on Persistent Platforms. The pick-and-place plus fastening chain directly enables removing and replacing orbital replaceable units (ORUs) on persistent platforms such as the Arkisys Port. The VLM interprets swap commands without pre-programming each sequence.

Use Case #7: Assembly of Large Persistent Platforms. The software framework enables robotic arms to autonomously join modular structural components using both permanent (riveting) and reversible (screwing) fastening, scaling from the Bosuns Locker demonstration to full platform assembly.

Use Case #2: Upgrade/Replacement of Instruments. The same capability chain applies to instrument replacement on client spacecraft. The platform-agnostic software deploys on any servicing vehicle's robotic arm.

Use Case #12: In-Space Assembly of Modular Spacecraft. Assembling modular spacecraft in orbit requires precisely the multi-step manipulation demonstrated: identify component, transport, align, join permanently (rivet) or reversibly (screw), and verify.

IX. Lessons Learned

A. Most Innovative Concepts Considered

Three innovative concepts were explored during the design process:

1. Sim-to-real transfer for zero-shot orbital deployment. Training the vision model entirely in a digital twin of the Bosuns Locker environment, then deploying directly to flight hardware with no on-orbit retraining. This would dramatically reduce commissioning time but requires closing the sim-to-real gap for lighting, texture, and sensor noise.

2. Multi-arm cooperative assembly via shared VLM. A single VLM could coordinate two or more robotic arms, providing coordinated action outputs from a shared perception backbone, a virtual second pair of hands for tasks requiring simultaneous holding and fastening.

3. Self-supervised anomaly detection during assembly. Leveraging the VLM's visual feature space to detect assembly anomalies (misaligned parts, incomplete rivets, foreign object debris) without explicit defect training data.

B. Most Important Technology Gaps

1. Space-qualified ML inference hardware. No radiation-hardened processor currently supports transformer-based vision models at speeds needed for responsive robotic control. Closing this gap, likely through commercial hardening of edge AI chips from NVIDIA or Qualcomm, would enable an entire class of AI-driven space robotics beyond our specific application.

2. Standardized robotic tool-change interface for microgravity. No industry-standard quick-change

Table 6: Risk identification and mitigation.

Risk	L	I	Mitigation
VLM inference latency on space-rated compute	H	H	Model quantization (4/8-bit); smaller VLMs (TinyLLaVA); hybrid ground/onboard architecture; VLM invoked once per task, not at control rate
Thermal management of edge compute in vacuum	M	H	Passive heat sinks with thermal interface material to Bosuns Locker structure; conduction-only cooling path; compute duty-cycled to limit sustained thermal load
Tool changeover reliability in μg	M	M	Proprietary twist-lock mechanism with force confirmation; three limit switches verify engagement; pogo pin electrical connection; no loose fasteners during swap
Vision accuracy under variable lighting	M	M	Onboard LED illumination provides consistent lighting; HSV detection thresholds tuned to workspace; multiple capture frames averaged
Communication loss during operation	L	H	Full autonomy after initial command; safe-hold state on comm timeout; telemetry stored for delayed downlink; no real-time ground dependency

tool interface exists for microgravity robotic arms. Our proprietary twist-lock addresses the immediate need, but a standardized connector analogous to industrial quick-change plates (e.g., ATI tool changers) would benefit the ISAM ecosystem.

3. On-orbit non-destructive evaluation (NDE). After riveting or screwing, no established method exists for autonomously verifying joint integrity in orbit. Force feedback provides partial confidence, but a compact ultrasonic or thermographic sensor integrated with the end effector would dramatically increase mission assurance.

C. Biggest Challenges Encountered

1. VLM compute feasibility (Technical). Deploying a 7-billion-parameter model on space-rated hardware is a significant challenge. This drove investigation of quantization, smaller models, and the hybrid ground/onboard architecture. The key insight is that VLM inference is a one-time-per-task event, not a continuous control loop, which fundamentally relaxes the compute constraint.

2. Sim-to-real gap in visual detection (Technical). HSV-based detection required careful tuning to discriminate target objects from visually similar scene elements. For example, the UR5's cyan joint caps have similar hue to blue target blocks; resolution required tightening saturation thresholds (floor of 150) and narrowing hue ranges. This challenge will intensify for flight deployment with real-world lighting variations.

3. Cross-disciplinary team coordination (Programmatic). Integrating ECE, CS, and Aerospace team members required establishing shared coordinate frame conventions and interface definitions early. An initial misalignment between the CAD model coordinate system and the vision pipeline's frame cost several weeks of rework.

X. Path Forward

A four-phase, 24-month path to Preliminary Design Review (PDR) is planned:

Phase 1: Model Optimization (Months 1–6). Quantize LLaVA for target edge hardware (NVIDIA Jetson Orin class). Benchmark alternative VLMs (TinyLLaVA, MobileVLM). Validate on 50+ assembly scenarios in simulation. Test OV-DINO as a complementary real-time perception layer. Benchmark autonomous performance against human teleoperation baseline.

Phase 2: Hardware Integration (Months 7–12). Integrate the compute unit with a flight-representative 6-DOF robotic arm. Design and manufacture the Bosuns Locker structural adapter. Qualify the twist-lock tool-change mechanism under repeated cycling. Build the flight-like end effector suite with integrated cameras and pogo pins.

Phase 3: Environmental Testing (Months 13–18). Thermal vacuum testing of the compute unit and arm assembly. Vibration and shock testing

to Falcon-class launch loads. EMI/EMC testing for Bosuns Locker interface compatibility.

Phase 4: System Integration and PDR (Months 19–24). Full system integration with a Bosuns Locker mockup. End-to-end autonomous assembly demonstration covering all three operations. Complete CDR-level documentation. Interface verification with the Arkisys Port architecture.

Future work beyond PDR includes physical testbed validation with a hardware-in-the-loop arm, sim-to-real transfer pipeline development, and multi-arm coordination.

XI. Conclusion

This paper presented the first application of vision-language-action reasoning to in-space servicing, assembly, and manufacturing task planning. A software pipeline combining LLaVA for natural language task reasoning, OpenCV for visual detection, depth-buffer deprojection for 3D localization, and IK/OMPL motion planning was demonstrated for autonomous pick-and-place in CoppeliaSim simulation with 4 mm end-effector accuracy and 32-second cycle time. A trade study comparing the VLA approach against OV-DINO object detection found that VLA's task-level reasoning capability justifies its higher compute requirements, which are addressable through model quantization and hybrid ground/onboard architectures. Three custom-designed interchangeable tool heads, connected via a proprietary twist-lock mechanism, provide pick-and-place, self-piercing riveting, and screwing capabilities that chain into a complete autonomous assembly workflow. The conceptual payload design fits within the Arkisys Bosuns Locker constraints with 40% sustained power margin and 80% mass margin. This platform-agnostic, software-first approach is deployable across different robotic systems, offering a path toward scalable autonomous orbital manufacturing.

Acknowledgments

The authors thank Elizabeth Kung (advisor) and Harsha Malshe (mentor) for their guidance throughout this project. The authors acknowledge COSMIC and the Aerospace Corporation for organizing the Capstone Challenge, Arkisys for the Bosuns Locker platform specifications, and Purdue University for institutional support.

References

[1] White House, "In-Space Servicing, Assembly, and Manufacturing (ISAM) National Strategy," National Science and Technology Council, Washington, D.C., 2022.

[2] COSMIC, "2025-26 COSMIC Capstone Challenge Information Packet," COSMIC-E01-WD005-2025-B, Oct. 2025.

[3] COSMIC, "Prioritized Use Cases for In-Space Servicing, Assembly, and Manufacturing," Consortium for Space Mobility and ISAM Capabilities, 2025.

[4] Rome, J. and Goyal, V. K., "Overview of the ISAM Design Challenge and Competition," AIAA Paper 2024-0628, AIAA SciTech Forum, Jan. 2024.

[5] Heying, J. and Rome, J., "The COSMIC Capstone Challenge," AIAA Paper 2025-0803, AIAA SciTech Forum, Jan. 2025.

[6] Friend, R. B., "Orbital Express Program Summary and Mission Overview," *Proceedings of SPIE*, Vol. 6958, 2008.

[7] Northrop Grumman, "Mission Extension Vehicle (MEV): In-Orbit Satellite Life Extension," Northrop Grumman Space Systems, 2020.

[8] Reed, B. et al., "On-Orbit Servicing, Assembly, and Manufacturing (OSAM-1)," NASA Goddard Space Flight Center, 2020.

[9] Obenchain, T. et al., "Technology Roadmap for Orbital Smallsat Factory," AIAA Paper 2024, AIAA SciTech Forum, 2024.

[10] Flores-Abad, A., Ma, O., Pham, K., and Ulrich, S., "A Review of Space Robotics Technologies for On-Orbit Servicing," *Progress in Aerospace Sciences*, Vol. 68, 2014, pp. 1–26.

[11] Brohan, A. et al., "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," *arXiv preprint arXiv:2307.15818*, 2023.

[12] Ghosh, D. et al., "Octo: An Open-Source Generalist Robot Policy," *arXiv preprint arXiv:2405.12213*, 2024.

[13] Kim, M. J. et al., "OpenVLA: An Open-Source Vision-Language-Action Model," *arXiv preprint arXiv:2406.09246*, 2024.

[14] Liu, H. et al., "Visual Instruction Tuning," *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[15] Oquab, M. et al., "DINOv2: Learning Robust Visual Features without Supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[16] Wang, H. et al., "OV-DINO: Unified Open-Vocabulary Detection with Language-Aware Selective Fusion," *arXiv preprint arXiv:2407.07844*, 2024.

[17] Radford, A. et al., "Learning Transferable Visual Models from Natural Language Supervision," *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[18] Sucan, I. A., Moll, M., and Kavraki, L. E., "The Open Motion Planning Library," *IEEE Robotics and Automation Magazine*, Vol. 19, No. 4, 2012, pp. 72–82.

[19] Rohmer, E., Singh, S. P. N., and Freese, M., "CoppeliaSim (V-REP): A Versatile and Scalable Robot Simulation Framework," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[20] He, X., Pearson, I., and Young, K., "Self-Pierce Riveting for Sheet Materials: State of the Art," *Journal of Materials Processing Technology*, Vol. 199, No. 1–3, 2008, pp. 27–36.

[21] Li, D., Chrysanthou, A., Patel, I., and Williams, G., "Self-Piercing Riveting—A Review," *International Journal of Advanced Manufacturing Technology*, Vol. 92, 2017, pp. 1777–1824.