

LLM-Drone: Aerial Additive Manufacturing with Drones Planned Using Large Language Models

Akshay Raman,* Chad Merrill,* and Amir Barati Farimani*

*Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave,
Pittsburgh, PA 15217, USA*

E-mail: akshayra@andrew.cmu.edu; camerril@andrew.cmu.edu; barati@cmu.edu

Abstract

Additive manufacturing (AM) has transformed the production landscape by enabling the creation of complex geometries with precision. However, AM faces limitations when applied to challenging environments, such as elevated surfaces and remote locations. Aerial additive manufacturing, facilitated by drones, presents a solution to these challenges by allowing construction in previously inaccessible areas. Despite advances in methods for the planning, control, and localization of drones, the accuracy of these methods is not good enough to additively manufacture parts. Recently, the emergence of LLMs has revolutionized various fields by introducing advanced semantic reasoning and real-time planning capabilities. This paper proposes the integration of LLMs with aerial additive manufacturing to optimize the planning and execution of construction tasks. Using the semantic understanding and adaptability of LLMs, drones can dynamically generate and adapt building plans on site, ensuring efficient and accurate construction even in constrained environments. This capability is extended to logistics tasks, such as sorting and palletizing in warehouses or shipyards, where LLMs use world knowledge to create semantic objects within a given coordinate system. We propose a textual prompt discretizing the build space for the LLM

to plan and send coordinates in which the aerial system will convert into locations to build. The LLM manages up to 90% build accuracy when given a semantic prompt and has shown great success in understanding the spatial environment even with tight planning constraints. The repeated feedback system of our LLM-Drone model also enables replanning using the LLM if manufacturing undergoes unforeseen errors. The semantic reasoning of the LLM allows for successful replanning of the planned build without complicated heuristics and evaluation functions. Our approach demonstrates the potential of combining LLMs with aerial additive manufacturing to not only overcome traditional manufacturing limitations but also to introduce innovative solutions in logistics and beyond.

Introduction

Additive manufacturing has emerged as a transformative technology, enabling the production of complex geometries and custom parts with high precision and minimal material waste.^{1–3} Its applications span across industries, from aerospace to healthcare, where intricate designs and efficient material usage are crucial. Despite its advantages, traditional additive manufacturing faces significant challenges when it comes to constructing structures in remote or constrained environments. Manufacturing is typically confined to specific locations, such as factories or controlled environments, making it difficult to implement on terrains such as disaster zones, mountainous regions, or extraterrestrial surfaces such as Mars.

To address these spatial limitations, drones have been proposed as a solution to extend the reach of additive manufacturing.^{4,5} Their mobility and ability to navigate diverse terrains present an opportunity to overcome location-based constraints. However, recent approaches that integrate drones with traditional additive manufacturing methods have encountered significant hurdles. The inherent instability of drones in flight poses difficulties in achieving the precision required for layer-by-layer fabrication, particularly in fluid-to-solid transitions commonly used in additive manufacturing processes.⁶

Recognizing these limitations, we introduce a novel approach that reimagines drone-assisted manufacturing. Rather than relying on drones to perform traditional additive manufacturing tasks, our method uses predefined geometries, specifically magnetically connecting blocks, that drones can transport and assemble. Using these standardized building blocks, we eliminate the need for precise in-flight material deposition, alleviating the challenges of drone stability and control.

In addition to solving the physical construction challenges, we leverage Large Language Models (LLMs) to enhance the design and planning phases of the process. LLMs, with their advanced reasoning and creative capabilities, can dynamically generate and adapt designs based on the current build state and user-defined requests. This combination of predefined geometries and intelligent design planning not only ensures accuracy in construction but also introduces a flexible, scalable method for drone-assisted manufacturing in diverse environments.

In this paper, we propose the LLM-Drone pipeline, which consists of 3 modules that highlight our approach to aerial additive manufacturing:

- LLM Planning Module - LLM pipeline that provides design based on current 3D build state and user defined design request
- Computer Vision Module - Vision module to align coordinate systems and estimate current state of build location
- Mechanical Module - Drone and block design allowing for pick-and-place of custom 3D printed blocks

Environment Setup

Consider a 3-D environment represented within the world as $W \subseteq \mathbb{R}^3$ with size H*W*L. The boundary of the environment is discretized into 1 cm sections represented by a numpy

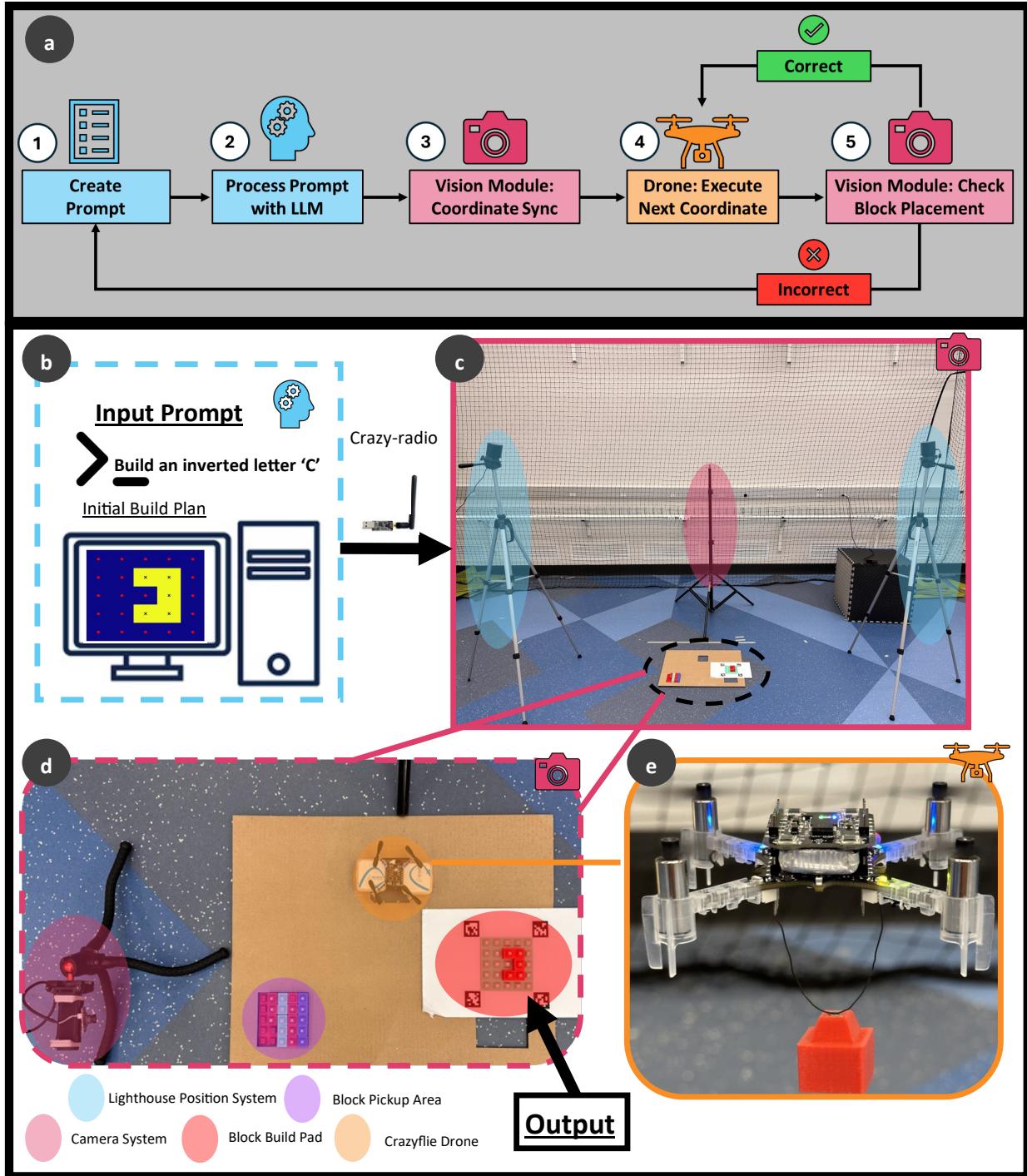


Figure 1: System Overview. (a) Highlights the main modules required for the additive manufacturing process. (b) A prompt is created from a default prompt template that includes current scene info and a design request. The LLM processes the prompt and outputs the coordinates needed to achieve the design. (c) The vision module aligns the Crazyflie coordinates with the LLM output coordinates using the Crazyflie lighthouse coordinates as world coordinates. (d) The drone executes a coordinate by picking up and placing a block. The vision module identifies the dropoff location of the block. If correct, the drone executes the next block placement. If incorrect, the current scene is based back to (b) for remprompt and get a new set of coordinates to finish the design. Based on the input prompt of creating an inverted letter 'c', the final output of our system is shown in d. (e) Visual example of Crazyflie drone transporting a building block from pickup to dropoff

array. The array delineates the environmental layout, classifying regions as obstacles or free space, which facilitates the navigation of drones. A roadmap can be constructed within this environment to traverse free space while avoiding obstacles during the construction process.

The experimental infrastructure utilizes Bitcraze’s Crazyflie ecosystem, including the Crazyflie Client, Crazyradio, and Lighthouse System, as illustrated in Figures 1b and 1c. This setup was used consistently across all data collection phases. Figures 1c, 1d and 1e highlights additional hardware including a Crazyflie 2.1 drone, a 1080p web cam, and the 3D printed blocks and building pads designed for the experiments. The Crazyflie ecosystem was selected for its robust built-in API, which seamlessly integrates with both the controller and the autonomous flight controller. This integration allowed our team to concentrate on developing a robust ’additive manufacturing’ system.

Methodology

The LLM-Drone pipeline⁷ combines three modules to autonomously construct a user-defined design request. Figure 1a highlights the integration of these modules in the design pipeline for the additive manufacturing process. First, the LLM Planning module captures the user design request to create the initial prompt. This prompt is fed into an LLM that outputs a set of coordinates as an action array that will accomplish the build task. Second, the Computer Vision module converts the action array into world (drone) coordinates using the coordinate sync algorithm. Third, the converted coordinate array is fed into the Crazyflie client, and the onboard controller executes a block pickup and dropoff for a single coordinate. Fourth, the Computer Vision module verifies the dropoff location of the block against the intended location. If the location is correct, the drone executes the next build coordinate. If the location is incorrect, a new prompt is generated based on which captures the current position of any blocks that have been placed. The LLM will then process the updated scene and generate coordinates to finish the design from the current state. We provide further

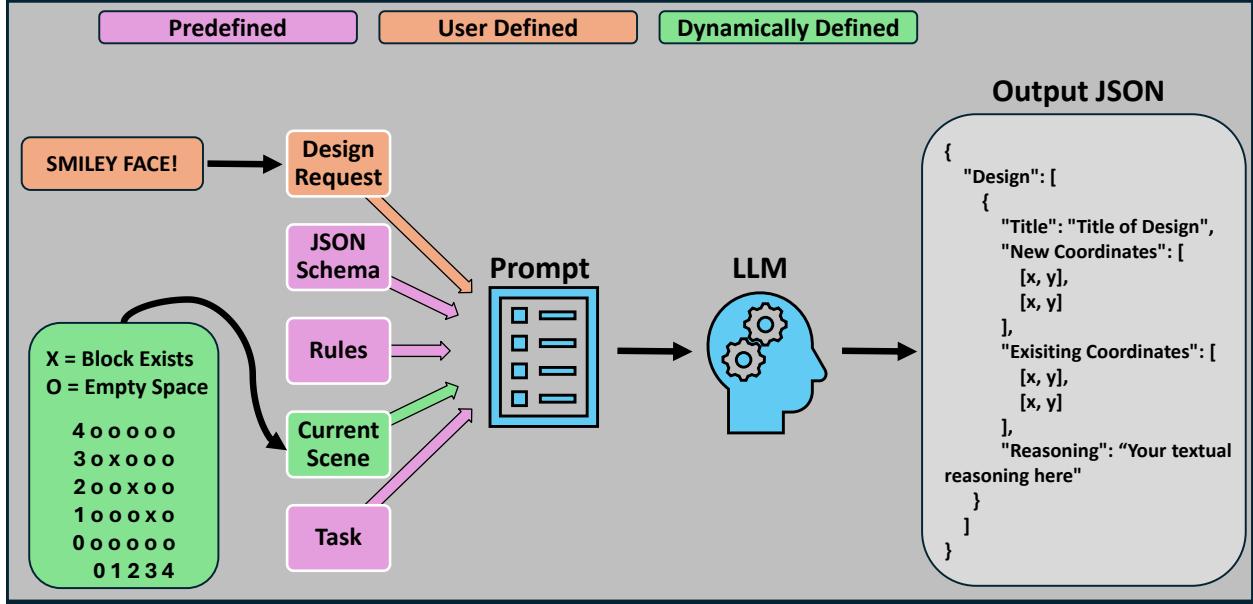


Figure 2: The prompt is broken into 5 parts. The Task, Rules, and JSON Schema are predefined and do not change. The Design Request is input by the user once at the start of the build process. The Current Scene is captured and presented each time a prompt is called.

detail into each of the 3 major modules in the remainder of this section.

LLM Planning Module

The LLM Planning Module includes the creation of the prompt and the processing of that prompt by an LLM. Figure 2 shows the structure for both the initial prompt and reprompt, respectively. The basic prompt structure remains the same for both types of prompts. The main components of the prompt are highlighted below.

- **Task:** The requirements of the task and what needs to be done. In our case, this is the definition of the coordinates that a drone will execute to build a desired design.
- **Design Request:** A text input from the user describing their desired design.
- **Current Scene:** This section begins by defining the dimensions and layout of the build space to the LLM. This includes a definition of the pad coordinate system. We then provide a textual representation of the grid system where o represents free spaces

and x represents spaces that a block already occupies, which is dynamically generated each time a prompt is created. During the drone execution phase, if an error is detected in the dropoff position by the Vision Module, this textual representation of the build space is updated in the reprompt. This ensures that LLM is prompted to redesign the required task using the existing block states as the initialization point.

- **Rules:** Defines rules the LLM must follow when selecting coordinates to create the requested design.
- **Output Schema:** Defines the JSON output schema with which the LLM should respond. The structure includes a title for the design, a set of coordinates for the drone to execute, any coordinates currently utilized on the build plate, and the LLMs' reasoning on how it created the design.

Computer Vision Module

The LLM-Drone Computer Vision Module integrates three submodules to enhance coordination and precision in drone operations. This section outlines the reasoning behind each vision state check and its integration within our application. The overall structure is presented in Algorithm ??.

Coordinate Sync

Although the LLM understands its environment based on prompts, there is no inherent alignment between the building pad and the coordinates within the internal frame of reference of the LLM. To minimize computational load on the LLM and prioritize the overall shape construction, the LLM output coordinates are relative to the build pad's local frame. The Coordinate Sync module translates these local coordinates from the LLM frame to the world coordinate system. In our setup, the lighthouse system defines the world coordinate frame.⁸

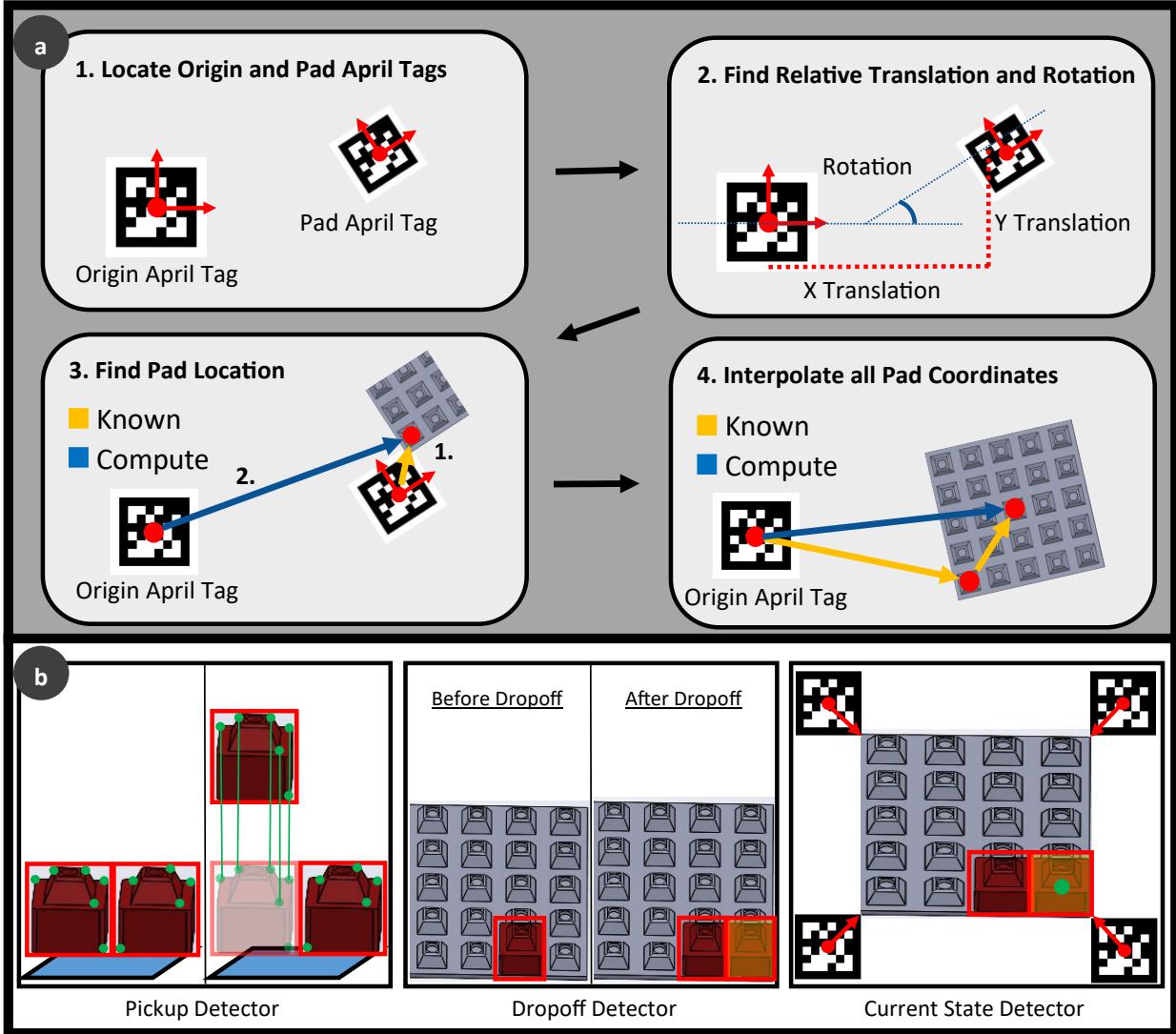


Figure 3: (a) Coordinate Sync algorithm overview. 1) Locate both origin AprilTag and Pad AprilTags. 2) Find relative transformation between origin and pad AprilTags. 3) Use Known Vector distances to compute blue vector position. 4) Interpolate any pad coordinate by adding known vector from origin to bottom left position of pad. (b) The Vision modules' 3 main purposes. Pickup Detector: Green points represents corner points for Lucas-Kanade to follow. If average of all tracking points within YOLO bounding box greater than threshold as shown in right, the block is deemed as moved. Dropoff Detector: A frame is captured before drone enters the region and drops a block. Background subtraction is compared with initial image to classify if a block has been placed. If the change is greater then a block has been placed. Current State Detector: The corners of the pad are interpolated from the location of the AprilTags. The center of spatial subtraction is given by green dot. The dot is interpolated to the closest pad location in x and y using AprilTags.

We propose two methods to determine the relationship between the pad's frame and the world coordinate frame.

(1) Hardcoding Method: Fix the location of the bottom right point on the pad and interpolate all other coordinates based on rotation and local positioning. This method provides higher accuracy, but requires that the build position be predetermined. (2) AprilTag-Based Method: Use AprilTags to dynamically compute the relationship between the world and the pad frames. This method offers flexibility in placing the pad at any location, allowing real-time computation of its position.

Figure 3 illustrates how coordinate synchronization aligns the local pad coordinates with the world frame and converts the LLM output action matrix into world coordinates.

Locate Necessary AprilTags: Figure 3 shows the two AprilTags detected by the calibrated stationary camera. These tags are used to establish their positions in the camera’s coordinate frame, C_f .⁹

Find Relative Translation and Rotation: Since the detected tags are represented in the camera frame, C_f , we compute the relative translation and rotation of the pad’s AprilTag with respect to the origin AprilTag. The AprilTag origin is aligned with the world frame. Thus, the relative translation vector (T_{rel}) and the relative rotation matrix (R_{rel}) in the world frame, W_f , can be calculated as follows:

$$T_{rel} = P_{origin_{C_f}} - P_{pad_{C_f}} \quad (1)$$

$$R_{rel} = R_{W_f C_f} * R_{C_f P_f} = R_{C_f W_f}^{-1} * R_{C_f P_f} \quad (2)$$

Compute Pad Location: This step is pre-computed as it is just the relational point from the origin of the AprilTag to the corner of the pad. This computation puts a relation between the pad AprilTag frame and the coordinate location of the first bottom left notch in dropoff pad.^{10,11} The location of this coordinate in world frame can be computed:

$$P_{notch1_{W_f}} = T_{rel} + R_{rel} * P_{notch1_{P_f}} \quad (3)$$

Interpolate All Pad Points: Each pad coordinate represented by LLM as integers

from bottom left (0,0) of pad to top right (environment size, environment size) can be interpolated based on the known location of the world frame position of (0,0). Figure 3 shows the computation required to find the pad coordinate (2,2) in world coordinates. The two red arrows are known parameters. Using the two vectors, the blue line can be computed. The computed coordinates are sent to the Crazyflie controller for execution.

Verify Action Completion and State Correctness

As depicted by Figure 3, the vision model includes verification of the completion of the pickup and drop-off action, as well as the current placement of the block within the build pad (correctness of state). Figure 3 shows the architecture behind the verifier.

Pickup Detector: YOLO-v8 is pre-trained with images of the final block iteration.^{12,13} The blocks are 3D printed with various colors to make it robust to detection at all lighting and colors. According to Table ??, there is a camera directly facing the pickup site. The frame angle allows the camera to see 1-2 layers behind for more robust detection. Inference is run on the custom trained YOLO-v8 model to track all building blocks in the frame. An ID is associated with each tracked block and the boundary box of the tracked block is masked.¹⁴ Within each masked out bounding box, the features of the top corner are detected and their movement is tracked using Lucas-Kanade as shown in Figure 3.¹⁵ Let $\mathbf{p}_i^{(t)}$ represent the position of the i -th feature point at time t . The vertical displacement of this point between frames t and $t + 3$ is given by:

$$\Delta y_i = p_{i,y}^{(t+3)} - p_{i,y}^{(t)}$$

where $p_{i,y}$ is the y -coordinate of the i -th point. The total vertical displacement ΔY is the sum of the vertical displacements of all points:

$$\Delta Y = \sum_{i=1}^N \Delta y_i = \sum_{i=1}^N \left(p_{i,y}^{(t+3)} - p_{i,y}^{(t)} \right)$$

where N is the number of feature points. Define a pickup threshold T . If ΔY exceeds this

threshold, determine that the block has been picked up:

$$\text{Pickup} = \begin{cases} \text{True} & \text{if } \Delta Y > T \\ \text{False} & \text{otherwise} \end{cases}$$

Dropoff Detector: Per ??, the dropoff depot camera is used for both state verifier and dropoff detector. Here, the bounds of the pad are estimated using AprilTags on all four corners. Unlike for coordinate sync, all 4 April are utilized to find the bounds of the block by finding the camera coordinates of the tips of the AprilTag that touches the pad. The pad is then masked from the image and a slight padding is applied around the masked region to account for the block leakage outside the pad. YOLO-v8 is applied to the masked region to track blocks.¹² If after placing a block the track count increases, then a block has successfully been dropped. However, sometimes YOLO-v8 is inconsistent. To verify, a frame before the drone appears and attempts to drop a block is captured, as well as a frame after the drone attempts a dropoff and clears the space. A simple static spatial background subtraction is applied to find additions between frames as shown by the highlighted region in Figure 3). A large addition could lead to the placement of a block as not captured by YOLO.^{16,17}

Current State Detector: The current state verifier is built on the dropoff detector. To determine the position of a block on the pad in the pad coordinate system and also to know whether a block is stacked or placed behind another block, we perform the following steps:

1. Pad Coordinate System Mapping

Detect the corners of the pad using AprilTags. Assume the pad corners are detected and ordered as follows: \mathbf{p}_0 (top-left), \mathbf{p}_1 (top-right), \mathbf{p}_2 (bottom-left), and \mathbf{p}_3 (bottom-right). Using these corners, map the block's position to the pad coordinate system.

2. Background Subtraction and Center of Change

Let I_{before} and I_{after} be the images before and after dropping the block. The difference image D is calculated as:

$$D = |I_{\text{after}} - I_{\text{before}}|$$

Identify the region of maximum change in D . Let Ω be the set of all pixels in this region.¹⁸

The centroid (c_x, c_y) of this region is given by:

$$c_x = \frac{1}{|\Omega|} \sum_{(x,y) \in \Omega} x, c_y = \frac{1}{|\Omega|} \sum_{(x,y) \in \Omega} y$$

3. Interpolating Pad Grid Coordinates

Using bilinear interpolation, calculate the grid coordinate:

$$x_{\text{grid}} = \left\lfloor \frac{c_x - p_{0,x}}{p_{1,x} - p_{0,x}} \cdot \text{pad_size} \right\rfloor$$

$$y_{\text{grid}} = \left\lfloor \frac{c_y - p_{0,y}}{p_{2,y} - p_{0,y}} \cdot \text{pad_size} \right\rfloor$$

4. Determine if Stacked or Placed Behind

Check if the new block is placed in a previously occupied position. Let ϵ_d be a distance threshold for considering the positions to be the same. Compute:

$$d_i = \|\mathbf{p}_{\text{new}} - \mathbf{p}_i\|$$

If $\min(d_i) < \epsilon_d$, check if the region of change is small. Let A be the area of the region of change and ϵ_A be the area threshold:

Stacked if $\min(d_i) < \epsilon_d$ and $A < \epsilon_A$

Placed Behind if $\min(d_i) > \epsilon_d$ or $A \geq \epsilon_A$

Mechanical Block Design

The primary objective of the block design was to develop a system that is viable within the payload capacity constraints of small drones such as a Crazyflie. Given the limited payload capacity of these drones, the design needed to be lightweight, compact, and capable of integrating well with other blocks to facilitate complex structures.¹⁹ The goal was to create a system that allows for simple pickup and drop-off mechanisms with minimal parts, making it feasible for small drones to operate efficiently. For larger drones, this concept can be expanded with more robust features, leveraging the additional payload capacity available.

Block Shape and Connectivity

The final block design was inspired by the interlocking capabilities of magnets. Magnets provide a straightforward and effective method for blocks to connect, allowing for relative inaccuracy in placement while still achieving a secure fit. This feature is particularly beneficial in aerial manufacturing, where maintaining stable hover positions without excessive movement is challenging. Magnetic interconnection ensures that the blocks can click together with ease, even if the positioning of the drone is not perfectly precise.^{20,21}

Each block features a magnetized interlocking mechanism that facilitates easy alignment and secure attachment. The blocks are designed with a hump at the top, serving as an auto-localizing feature to guide the alignment when stacking multiple blocks vertically. Figure ?? shows the shape and structure of the block on the left. This block can be easily printed in all colors.

Magnetic Transport

The pickup design leverages the magnet-based autolocking system to facilitate block handling. Each side of the drone incorporates a rotatable wire acting as a hinge, allowing

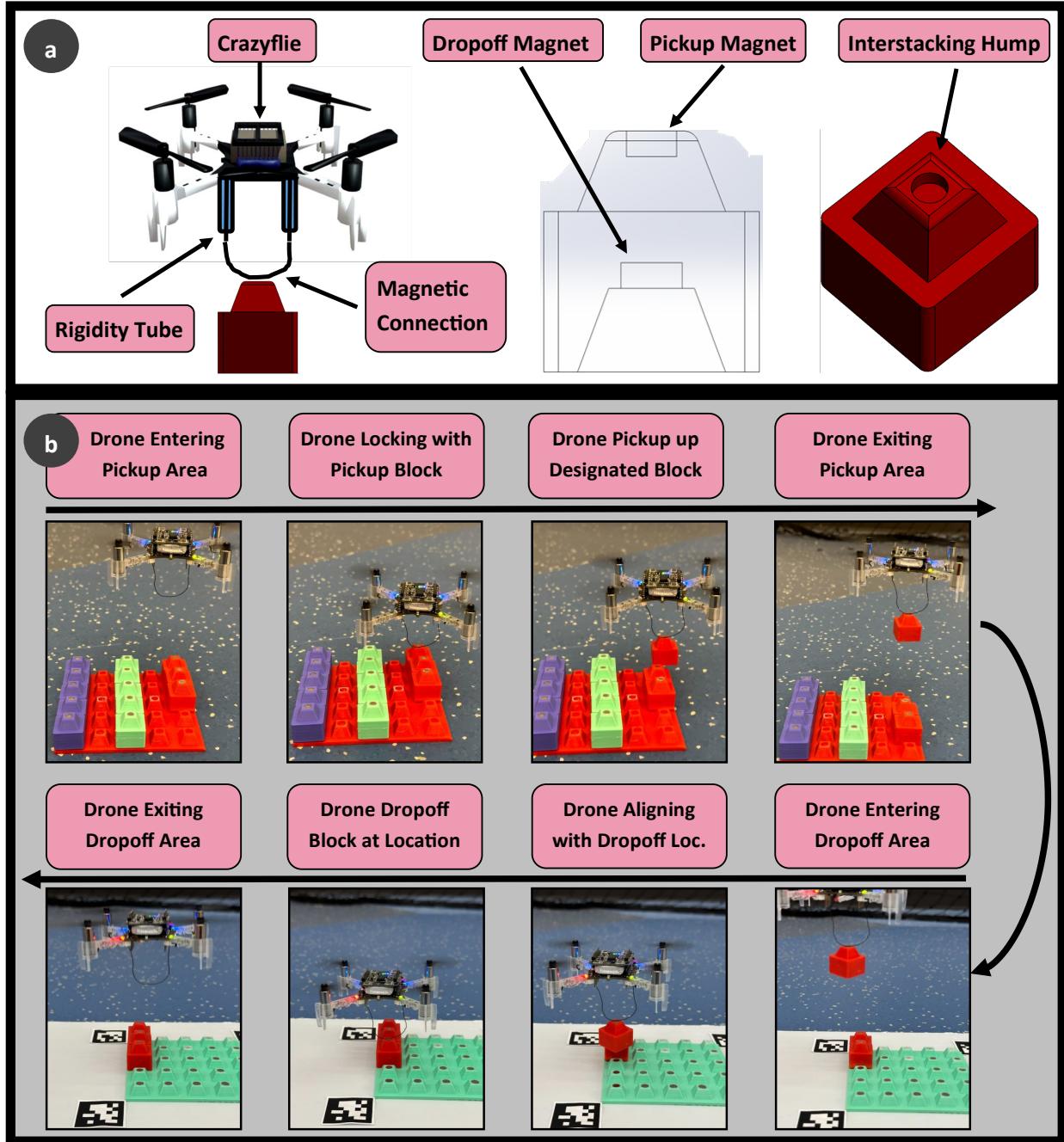


Figure 4: (a) Model of Crazyfly pickup apparatus. A rigid tube keeps the z-length of ferrous wire constant over numerous pickup/dropoff attempts. Wire has ability to move toward magnetic attraction from 'pickup magnet' on block. Wireframe image showing the placement of both pickup/dropoff magnet. 3D render of block with distinct hump for interlocking with other blocks. (b) Visual mechanical timelapse of the pickup and dropoff procedure. The stronger dropoff magnet allows the drone to click away from the building block once placed.

movement in one degree of freedom. The wire hangs free below the drone rotation towards the magnetic attraction of the block when near. Figure 4 highlights the key components of the mechanism below the drone. Once the block is locked, the drone can carry and position it at the desired location. A stronger magnet, integrated into the drop-off system, attracts the block more forcefully, allowing the drone to release it effortlessly because of the force differential.

Future Inspiration

This system is designed to be adaptable, with the potential for enhancements such as incorporating magnets that can be turned on and off. This capability would further optimize the performance and flexibility of the design, allowing for more precise control and a wider range of applications.^{20,21} This block design is intended for use in various settings, offering a versatile and scalable solution for construction and assembly tasks using drones. By building a system where the magnetic fields can be controlled, either through electromagnetic switches or other mechanisms, the functionality can be significantly improved, enabling more complex and precise operations.

Experiments

Testing Environments

To compare the robustness of the overall feasibility of the build environment as well as the LLMs capabilities, the algorithm was tested on two different platforms. The 3D additive manufacturing algorithm can be divided into two different problems, 1. Brain: LLM creativity and robustness to errors during manufacturing, 2. Body: The ability to take in commands from the LLM and convert them into correct actions and provide valid feedback to remedy any errors. Essentially, we design a virtual test scenario that validates the LLM performance and another physical scenario to validate hardware cooperation and the ability

to utilize LLM actions.

Metrics

We tested three different LLMs: Claude 3.5 Sonnet, OpenAI 4-o,²² and Gemini Pro 1.5²³ by comparing their performance in both quantitative and qualitative tests.

The first test consisted of a quantitative assessment of LLM designs on a 10x10 grid. For this test, we created fifteen "constrained prompts," which are prompts in which only one answer can be considered correct. The purpose of these constrained prompts was to restrict the output of the LLMs for precise accuracy measurement. We evaluated their performance using the Intersection Over Union (IoU) metric, comparing the generated designs with manually defined correct answers. Each LLM was tested five times per prompt and an average IoU was calculated across these responses. Figure 5A provides examples of six constrained prompts, their corresponding correct design plots, and the average IoU for each model on those specific prompts. Figure 5B provides the average IoU for the 15 prompts, the variance, the inference time in milliseconds, and the cost per 1000 tokens for each model.

The second test consisted of a qualitative comparison conducted on a smaller 5x5 grid. This test focused on open-ended design requests for simple geometric shapes, such as "star", "trapezoid", and "right triangle". We evaluated the feasibility and similarity of the generated designs, scoring each on a scale from 1 to 3, with 1 indicating both feasibility and recognizability, and 3 indicating neither. Figure 5C shows a bar chart showing the number of times each LLM scored 1, 2, or 3 on 30 different designs that we evaluated.

LLM Results

The quantitative results of our test, shown in Figures 5a and 5b, indicate that Claude 3.5 Sonnet achieved the highest average IoU at 89.5% in the fifteen constrained prompt tests. This suggests that Claude 3.5 Sonnet performed better than the other LLMs when tasked with scenarios that require strong spatial reasoning abilities. Furthermore, Claude 3.5 Sonnet

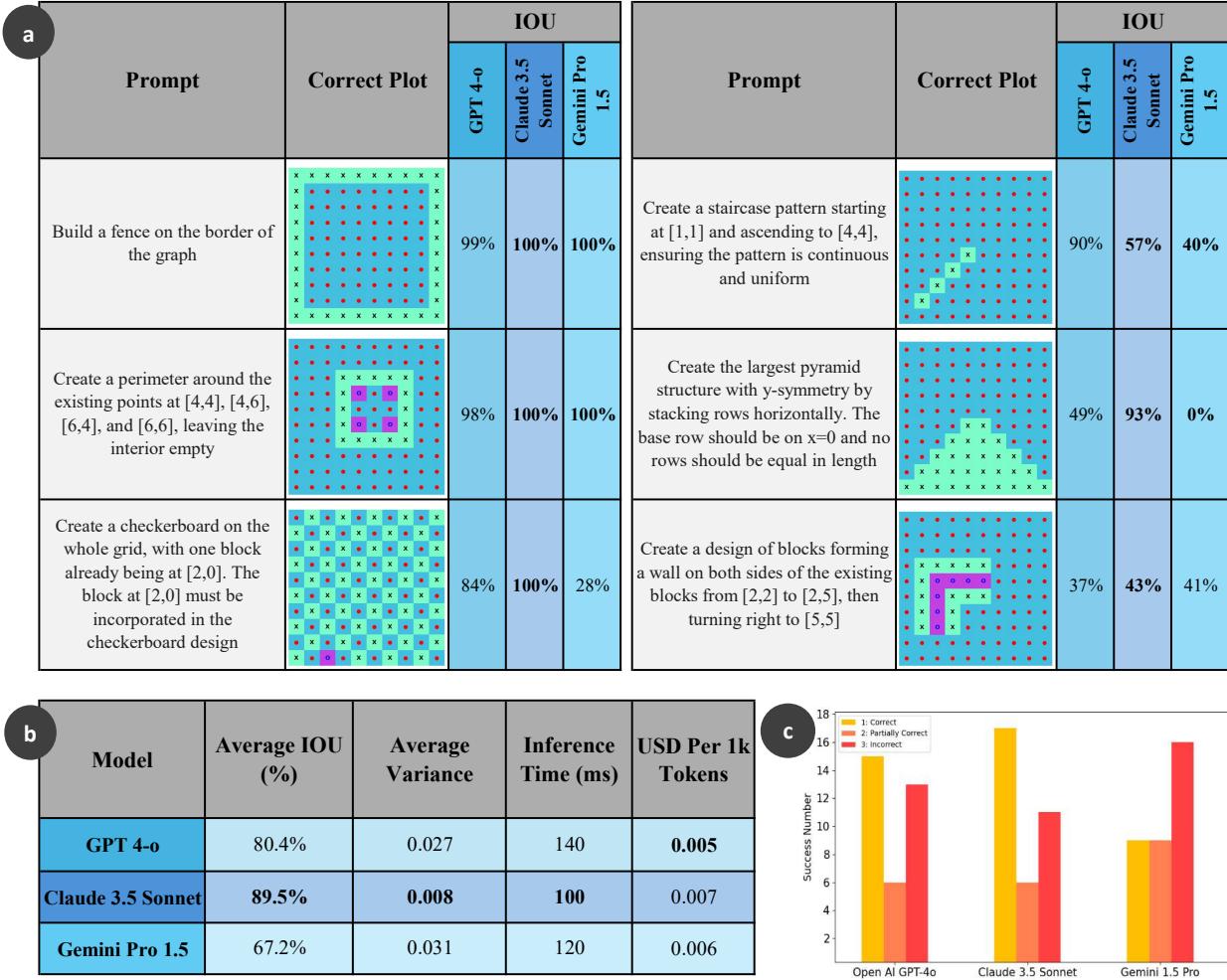


Figure 5: Performance of LLMs in the quantitative test (a and b) and qualitative test (c).

exhibited the lowest average variance in all its designs, indicating greater consistency in its responses, regardless of whether the answers were correct or incorrect.

The qualitative results, shown in Figure 5c, reveal that the designs generated by Claude 3.5 Sonnet were rated slightly higher than those of GPT 4-o, and both models outperformed Gemini Pro 1.5 by a significant margin. These findings suggest that both Claude 3.5 Sonnet and GPT 4-o demonstrate a developing capability to handle open-ended design tasks.

Overall, we find that Claude 3.5 Sonnet demonstrates the greatest ability for spatial reasoning within the framework of our task. Additionally, it has the fastest inference time of the three models, further enhancing its suitability for time-sensitive tasks, though at a

slightly higher cost.

Crazyflie Ecosystem Performance

We test the LLM-Drone pipeline with the Crazyflie ecosystem on a 5x5 grid build space. We execute the pipeline for several design requests, including a smiley face, cross, diamond, square, the letter L, and "two columns on the left and bottom right corner only." In order to evaluate the effectiveness of reprompting the LLM on placement errors by the drone, we execute runs for each of these designs both with and without the reprompting enabled. Figure 6a provides a step-by-step walkthrough of the pipeline executing the design of a smiley face with the reprompting enabled. Figure 6b provides a comparison of the initial LLM design plan (top), an execution without reprompting (middle), and an execution with reprompting (bottom). Note that in all images, we replace incorrectly placed blocks with purple blocks to aid in visualization. The results of reprompting versus no reprompting show that many designs can be recovered by the LLM after the drone incorrectly places a block. For example, for the cross design the LLM identifies that it can simply rotate the coordinates to accommodate the misplaced block. For the diamond and letter L designs, the LLM provides new layouts that incorporate misplaced blocks. Finally, for the square design, the LLM identifies that the misplaced block was simply an out-of-order placement, thus keeping the same overall design while telling the drone to return to the spot it previously missed. In general, these tests show that many errors can be overcome by providing the LLM with information about the current scene and allowing it to update the design to accommodate various errors.

Conclusion

The integration of Large Language Models (LLMs) with aerial additive manufacturing represents a transformative step forward in the field of construction and logistics, offering a

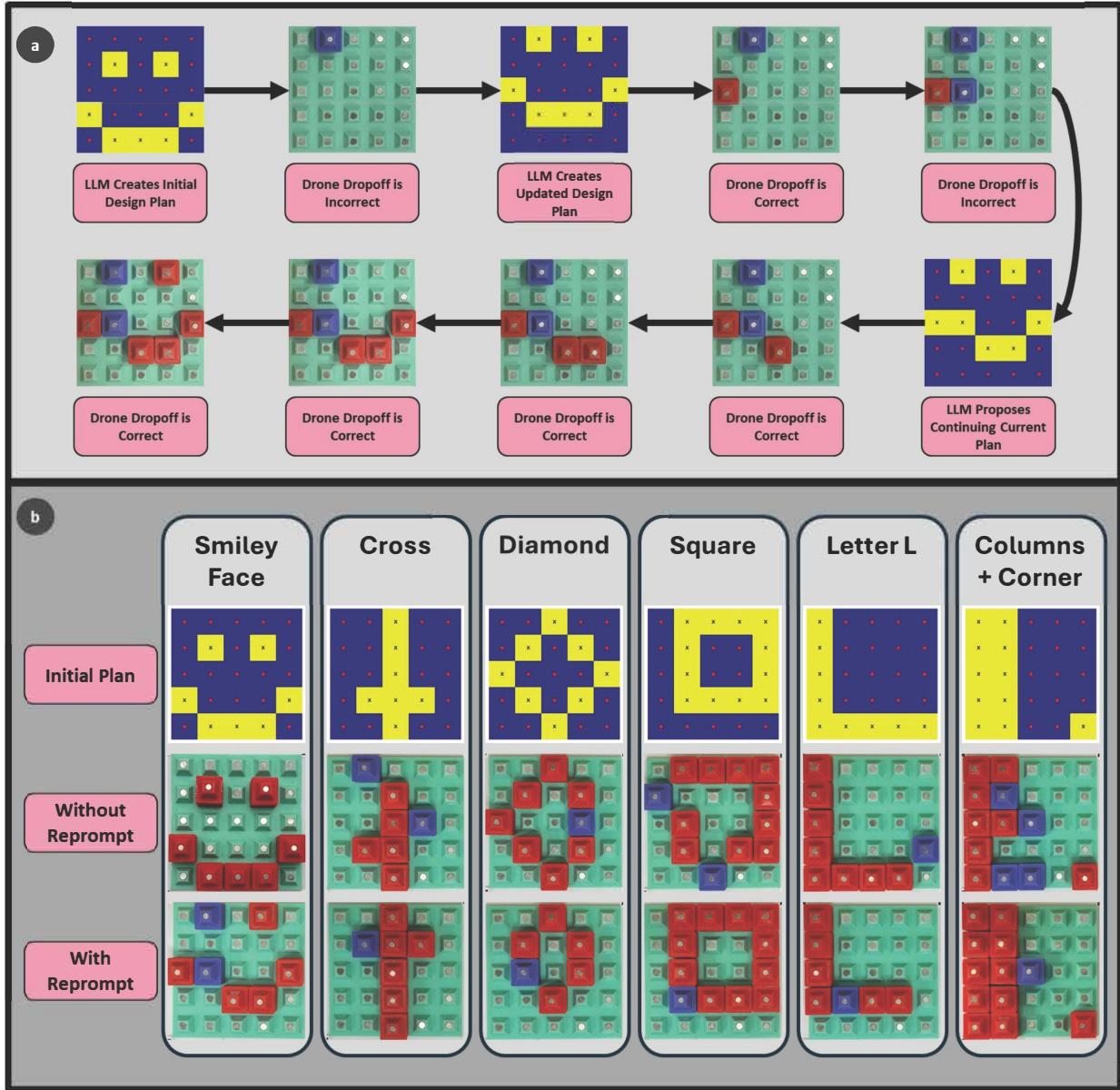


Figure 6: Crazyflie Ecosystem Results: a) Outlines a step-by-step design process with re-prompts enabled for the design of a Smiley Face. b) Provides an overview of 6 designs both with and without re-prompts.

dynamic solution to the challenges posed by inaccessible terrains and complex construction tasks. Through our research, we demonstrate the capabilities of LLM-driven drones to achieve high levels of precision and adaptability in real world settings. Our findings indicate that the LLM Planner can manage up to 90% build accuracy, enhancing both the planning and execution phases of aerial manufacturing tasks. We also conduct a performance compar-

ison across various LLM platforms, revealing that Claude 3.5 Sonnet outperformed others in spatial reasoning and subjective correctness of object construction.

Our research underscores the great potential of merging cognitive computing with physical manufacturing processes. The LLM-Drone system paves the way for innovative applications in various industries, including remote construction, warehouse automation, and emergency infrastructure deployment. Future work can explore taking LLM reasoning into 3D. While our current additive manufacturing platform allows for three-dimensional building, we are limited by the LLM planners ability to create 3D structures. Additionally, we aim to test the manufacturing process in various real-world testing platforms that mimic building in more confined regions to greater validate our proof-of-concept being modular across different build configurations.

References

- (1) Jadhav, Y.; Berthel, J.; Hu, C.; Panat, R.; Beuth, J.; Barati Farimani, A. Generative Lattice Units with 3D Diffusion for Inverse Design: GLU3D. *Advanced Functional Materials* **2024**, 2404165.
- (2) Jadhav, Y.; Farimani, A. B. Large language model agent as a mechanical designer. *arXiv preprint arXiv:2404.17525* **2024**,
- (3) Jadhav, Y.; Pak, P.; Farimani, A. B. LLM-3D Print: Large Language Models To Monitor and Control 3D Printing. *arXiv preprint arXiv:2408.14307* **2024**,
- (4) Augugliaro, F.; Lupashin, S.; Hamer, M.; Male, C.; Hehn, M.; Mueller, M. W.; Willmann, J. S.; Gramazio, F.; Kohler, M.; D'Andrea, R. The Flight Assembled Architecture installation: Cooperative construction with flying machines. *IEEE Control Systems Magazine* **2014**, 34, 46–64.

- (5) Latteur, P.; Goessens, S.; Breton, J.; Leplat, J.; Ma, Z.; Mueller, C. T. Drone-Based Additive Manufacturing of Architectural Structures. 2015.
- (6) Zhang, K.; Chermprayong, P.; Xiao, F.; Tzoumanikas, D.; Dams, B.; Kay, S.; Kocer, B. B.; Burns, A.; Orr, L.; Alhinai, T.; others Aerial additive manufacturing with multiple autonomous robots. *Nature* **2022**, *609*, 709–717.
- (7) Chen, Y.; Arkin, J.; Zhang, Y.; Roy, N.; Fan, C. Scalable Multi-Robot Collaboration with Large Language Models: Centralized or Decentralized Systems? *arXiv preprint arXiv:2309.15943* **2023**,
- (8) Evans, P. R. Rotations and rotation matrices. *Acta Crystallographica Section D: Biological Crystallography* **2001**, *57*, 1355–1359.
- (9) Kallwies, J.; Forkel, B.; Wuensche, H.-J. Determining and improving the localization accuracy of AprilTag detection. 2020 IEEE International Conference on Robotics and Automation (ICRA). 2020; pp 8288–8294.
- (10) Springer, J.; Kyas, M. Evaluation of Orientation Ambiguity and Detection Rate in April Tag and WhyCode. 2022 Sixth IEEE International Conference on Robotic Computing (IRC). 2022; pp 281–286.
- (11) Kalaitzakis, M.; Cain, B.; Carroll, S.; Ambrosi, A.; Whitehead, C.; Vitzilaios, N. Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers. *Journal of Intelligent & Robotic Systems* **2021**, *101*, 1–26.
- (12) Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction* **2023**, *5*, 1680–1716.

- (13) Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. *Sensors* **2023**, *23*, 7190.
- (14) Xianbao, C.; Guihua, Q.; Yu, J.; Zhaomin, Z. An improved small object detection method based on Yolo V3. *Pattern Analysis and Applications* **2021**, *24*, 1347–1355.
- (15) Baker, S.; Matthews, I. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision* **2004**, *56*, 221–255.
- (16) Xiong, J.; Wu, J.; Tang, M.; Xiong, P.; Huang, Y.; Guo, H. Combining YOLO and background subtraction for small dynamic target detection. *The Visual Computer* **2024**, 1–10.
- (17) Garcia-Garcia, B.; Bouwmans, T.; Silva, A. J. R. Background subtraction in real applications: Challenges, current models and future directions. *Computer Science Review* **2020**, *35*, 100204.
- (18) Kumar, S.; Yadav, J. S. Video object extraction and its tracking using background subtraction in complex environments. *Perspectives in Science* **2016**, *8*, 317–322.
- (19) Bitcraze AB Crazyflie 2.1 Datasheet. 2024; https://www.bitcraze.io/documentation/hardware/crazyflie_2_1/crazyflie_2_1-datasheet.pdf, Accessed: 2024-08-28.
- (20) Cho, D.-J.; Woo, D.-K.; Ro, J.-S.; Chung, T.-K.; Jung, H.-K. Novel electromagnetic actuator using a permanent magnet and an inter-locking mechanism for a magnetic switch. *IEEE Transactions on Magnetics* **2013**, *49*, 2229–2232.
- (21) Chung, D. G.; Kim, J.; Baek, D.; Kim, J.; Kwon, D.-S. Shape-Locking Mechanism of Flexible Joint Using Mechanical Latch With Electromagnetic Force. *IEEE Robotics and Automation Letters* **2019**, *4*, 2661–2668.

- (22) Bae, J.; Kwon, S.; Myeong, S. Enhancing Software Code Vulnerability Detection Using GPT-4o and Claude-3.5 Sonnet: A Study on Prompt Engineering Techniques. *Electronics* **2024**, *13*.
- (23) Team, G. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. 2024; <https://arxiv.org/abs/2403.05530>.