Language-Driven Agents for Drone Light Show Design

Nico Posner, Richard Huang, Sarthak Dhanke

Huanlin Dai, Binting Xia

Advisor: Stephen Barry

University of Chicago

Master of Science in Applied Data Science

Research Capstone Project

October 26, 2025

Abstract

This project investigates the role of large language models (LLMs) in the early-stage design of drone light shows. Rather than assume their utility, we explore whether language-based agents can meaningfully assist professionals in generating and refining swarm choreographies. Through literature review, prototype development, and evaluative experiments, we aim to identify where LLMs help, where they hinder, and how they can be reframed as collaborative tools. Our work contributes insights toward building more interpretable, practical, and user-centered systems — bridging the gap between creative intent and executable formations.

Keywords: drone swarm choreography, natural language interface, large language models, human-AI collaboration, creative robotics, prompt engineering, formation generation, interpretable AI, simulation tools, user-centered design

Executive Summary

Designing drone light shows is a complex process that combines creative choreography with technical precision. While commercial software platforms exist to support production, they are typically built for trained professionals and require manual specification of shapes, transitions, and safety constraints. As large language models (LLMs) become increasingly capable of interpreting natural language, researchers have proposed using them to translate high-level user prompts into drone swarm behaviors. However, current approaches typically use LLMs as translators — not as co-creative agents — and rarely engage with the realities of professional drone show design.

This project investigates whether language-based agents can meaningfully assist in the early stages of drone choreography, with a focus on feasibility, interpretability, and user alignment. Our goal is not to automate the design process, but to explore how LLMs can serve as collaborative tools that reduce the cognitive and operational overhead of translating intent into executable formations.

We begin by reviewing both academic papers and commercial tools. Prior work includes systems such as CLIPSwarm (Pueyo et al., 2024), SwarmGPT-Primitive (Jiao et al., 2023), Swarm-GPT (Vyas et al., 2024), FlockGPT (Lykov et al., 2024), and LLM-Flock (Li & Zhou, 2025). These systems demonstrate various architectures — from prompt-to-waypoint translation to decentralized plan generation — but none are grounded in actual design workflows or tested for usability by real users.

Commercially, tools like SPH Engineering's Drone Show Software and Verge Aero's Design Studio support professional use but lack intuitive or assistive interfaces for creative ideation. Our project situates itself in this gap: exploring how a prompt-based "design agent" could support iterative, interpretable, and rapid prototyping of drone choreographies.

To investigate this, we will:

1. Interview professionals (if possible) to understand current workflows and bottlenecks.

2. Develop a modular prototype agent that accepts user prompts, interprets them via LLMs,

and visualizes resulting formations.

3. Evaluate outputs based on interpretability, consistency, and alignment with user intent.

4. Compare structured prompt templates against open-ended input to assess design trade-offs.

Our expected outcome is not a production-ready deployment tool, but a critically evaluated prototype and a set of design insights. These may inform future tools that aim to bridge the divide between intuitive expression and technical execution in drone choreography — whether through language models or other interfaces.

Table of Contents

Abstract . . . . **i**

Executive Summary . . . . **ii**

List of Figures . . . . **v**

List of Tables . . . . **v**

List of Appendices . . . . **v**

Introduction **1**

Problem Statement 1

Analysis Goals 2

Scope 2

Background **4**

Commercial Tools 4

Academic Work 5

Identified Gaps 6

Data **6**

Methodology **6**

Workflow Analysis 6

Agent Development 7

Evaluation 7

Limitations 8

Findings **8**

Discussion **9**

Conclusion 9

Additional Data Analysis 12

Code Documentation 12

LLM Prompt Examples and Outputs 12

List of Figures

List of Figures

List of Tables

List of Tables

List of Appendices

Appendix A: Additional Data Analysis 12

Appendix B: Code Documentation 12

Appendix C: LLM Prompt Examples 12

Introduction

Problem Statement

Drone light shows have emerged as a visually striking form of large-scale entertainment, combining artistic design, precise swarm coordination, and real-time control of autonomous aerial vehicles. Despite their growing popularity in public events and commercial productions, the design and execution of such shows remains a labor-intensive and technically complex process. Creating choreographies that are not only visually appealing but also physically feasible requires expert input at multiple stages, from formation design and timing to collision avoidance and deployment logistics.

Recent research efforts have proposed using Large Language Models (LLMs) as intuitive interfaces for guiding drone swarm behaviors. The appeal of this approach lies in the promise of natural language interaction: a user simply describes the desired outcome (”make a spiral that transforms into a cube”) and the LLM generates the appropriate drone trajectories. However, a closer inspection of the current body of work reveals a critical limitation: LLMs are not being used to control swarms, but merely to generate intermediate representations (e.g., waypoints, motion primitives, alpha-shape outlines) that are then passed to conventional control or optimization algorithms. In effect, LLMs serve as translators, not choreographers.

While existing research on LLM-driven drone choreography primarily emphasizes technical feasibility and simulation results, there appears to be a lack of direct engagement with industry professionals or comprehensive evaluations comparing these approaches to established GUI-based tools and scripting systems. As a result, it remains unclear whether LLMs meaningfully improve the creative process, reduce production time, or resolve any specific pain points in real-world workflows.

This disconnect raises the central problem our project seeks to address: Can natural language interfaces meaningfully reduce the cognitive or operational load of drone show design, and under what conditions are LLMs actually useful? To answer this, we must move beyond proofof-concept demos and engage critically with the needs, constraints, and workflows of real-world drone light show designers. Without this grounding, efforts to integrate LLMs into choreography systems risk being performative rather than transformative.

Analysis Goals

The goal of this research is not to build yet another language-to-trajectory pipeline for drone swarms, but to critically examine the role, and limitations, of large language models (LLMs) in the process of drone light show design. Our analysis centers on the gap between the operational realities of drone choreography and the assumptions made in current LLM-driven frameworks.

To that end, we define the following guiding questions:

1. What are the actual constraints, tools, and creative workflows used by professionals who design drone light shows today?
2. What roles are LLMs currently being assigned in drone swarm research, and are those roles addressing real problems faced by practitioners?
3. Can we propose and prototype alternative interfaces, whether LLM-based or not, that are better aligned with practitioner needs?
4. What are the limits of LLMs in this context, in terms of creativity, safety, generalization, and interpretability?

Rather than treat LLMs as inevitable, this research treats them as a hypothesis: that they can act as meaningful collaborators in the design of drone light shows. Our aim is to test that hypothesis not by chasing performance benchmarks, but by understanding what matters to those who actually do this work, and how technology might truly assist them.

Scope

This project is intentionally scoped to explore the use of language-based interfaces for the design phase of drone light shows. Our focus is not on drone deployment or control, but on understanding whether and how Large Language Models (LLMs) can assist in the early stages of creative ideation, including formation design, sequencing, and user interaction.

Specifically, we do not aim to:

- Develop novel control algorithms for real-time drone navigation or formation stabilization.
- Train or fine-tune new LLMs from scratch.
- Benchmark LLMs as standalone models. However, we may compare outputs from multiple models to explore how reliably and interpretably they support design tasks.
- Conduct flight tests beyond limited-scale simulation or small-scale indoor platforms (e.g., Crazyflie).
- Replace industry-standard drone show design software or claim deployment readiness.

Instead, our work is framed around developing a prototype design agent, a lightweight, prompt-based assistant that can interpret user inputs (natural language or structured) and generate candidate drone choreographies, which can then be visualized, refined, or re-prompted. This agent is not meant to replace professional tooling, but to augment it, enabling faster prototyping and intuitive exploration.

Our focus includes:

- Studying the workflow, constraints, and tools currently used by professionals through interviews and secondary research.
- Prototyping modular interaction pipelines (e.g., prompt $\rightarrow$ LLM $\rightarrow$ shape/trajectory $\rightarrow$ visualization).
- Evaluating outputs for interpretability, consistency, and usability by both expert and novice users.

Laypeople can explore the tool, but the core value is for professionals and researchers seeking to reduce iteration time and lower the barrier between creative intent and technical implementation.

Ultimately, this research aims to lay the foundation for a new kind of collaborative interface, one that doesn't automate creativity, but facilitates it. We are not claiming to deliver a fully autonomous agent at this stage. However, the long-term goal is to evolve this system into a co-creative assistant that blends the strengths of human designers and generative language models in producing safe, expressive, and feasible drone light shows.

Background

Commercial Tools

Today, multiple companies offer professional-grade software platforms to support drone light show production. Tools such as SPH Engineering's Drone Show Software, Verge Aero's Design Studio, and Vimdrones Designer enable users to create and preview 3D formations, script movement paths over time, and integrate visual elements like lighting and branding. These platforms are typically timeline-based, GUI-driven, and built for production environments with specific drone hardware in mind.

While powerful, these tools assume a high degree of familiarity with the technical pipeline. They are not accessible to users outside the domain and require training, experience, and often vendor lock-in. More importantly, they provide little to no support for early-stage creative exploration. There is no natural language interface, no intelligent co-creation support, and limited ability to iterate fluidly based on rough conceptual prompts. This presents a missed opportunity: the most time-intensive and cognitively demanding part of the workflow, shaping the creative intent into executable sequences, remains largely manual.

Academic Work

A wave of academic interest has emerged around using Large Language Models (LLMs) to generate drone swarm behaviors. These efforts share a common goal: enabling intuitive interfaces for human-guided swarm design. However, the methods differ significantly, and each comes with critical limitations.

CLIPSwarm uses CLIP embeddings to map single-word prompts (e.g., "leaf") to 2D contour formations generated via alpha shapes. The formation is refined iteratively to improve CLIP similarity. While visually compelling, the system is limited to 2D silhouettes, lacks fine control, and shows semantic drift. Importantly, CLIPSwarm has only been tested in photo-realistic simulation, with no hardware demonstration.

SwarmGPT-Primitive generates a sequence of motion primitives (spiral, helix, wave) that align with musical beats. It includes a safety filter and self-correction for invalid LLM outputs. While this method improves reliability, it constrains creativity to a fixed library of motions and assumes prompt-to-primitive alignment can capture intent.

Swarm-GPT uses LLMs to generate waypoints for each drone, synchronized with beat timings from an audio track. These are passed through a safety optimizer before deployment. While more flexible, the approach suffers from verbosity, repetition, and difficulty modifying specific parts of the show. It also operates centrally, rather than distributing reasoning.

LLM-Flock explores the use of LLMs for decentralized formation planning. Each robot runs an LLM locally and adopts plans through an influence-based consensus protocol. This novel setup supports adaptability and demonstrates physical deployment with Crazyflie drones. However, the approach is currently limited to basic shapes (circle, triangle), with no support for creative choreography or sequence design.

FlockGPT introduces signed distance functions (SDFs) to define target formations. A central LLM converts natural language prompts into SDFs, which are then used to generate formations and allow real-time dialogue for revision. This is the most interaction-focused framework to date, but is limited to static geometry, lacking any notion of timing, motion, or music.

Identified Gaps

Across both commercial and academic approaches, several common limitations emerge:

- Studying the workflow, constraints, and tools currently used by professionals through interviews and secondary research.
- Prototyping modular interaction pipelines (e.g., prompt $\rightarrow$ LLM $\rightarrow$ shape/trajectory $\rightarrow$ visualization).
- Evaluating outputs for interpretability, consistency, and usability by both expert and novice users.

This work positions itself to address these gaps by building a prototype agent that emphasizes human-in-the-loop iteration, creative support, and workflow awareness, without overpromising full automation or technical generalization.

Data

[Your Data content here]

Methodology

Workflow Analysis

We begin by mapping the current state of drone show design workflows:

- Interviews with professional drone choreographers or companies (if accessible) to understand their tools, timelines, and pain points.
- Secondary research using documentation, training videos, and technical manuals from platforms like SPH Engineering or Verge Aero to reverse-engineer the design constraints and expectations of existing software.

This step ensures that the work is grounded in actual creative and operational needs, rather than assumptions derived from simulation-based research.

Agent Development

We will design and implement a lightweight language-driven design agent that enables users to generate and iterate on drone choreographies through natural language or structured inputs. This agent will consist of the following components:

- **Prompt Interface**: Accepts high-level textual commands such as "form a cube that rotates and then expands into a spiral."
- **LLM Processing Module**: Parses and interprets prompts using pre-trained language models (e.g., GPT-4, Claude, or LLaMA). We may test outputs across different models to assess consistency and interpretability, but we do not aim to benchmark them as standalone systems.
- **Formation Generator**: Translates LLM output into intermediate representations such as waypoints, motion primitives, or SDF-based geometries.
- **Visualizer**: Renders the generated formations and motion sequences using 2D or 3D plotting libraries or simulation environments (e.g., Matplotlib, Plotly, Unity).

This pipeline is designed to be modular, allowing us to replace or isolate components for controlled evaluation.

Evaluation

Rather than focusing solely on model performance, we will evaluate the system along qualitative and user-centered dimensions:

- **Prompt consistency**: Does the same prompt yield semantically equivalent outputs across sessions?
- **Editability**: Can users easily refine outputs via re-prompts or structured input tweaks?

- **Recognizability**: Do generated shapes align with user intent (visually and semantically)?
- **Transparency**: Can users understand how outputs were generated, or are they opaque?
- **Time savings / iteration speed**: Where possible, compare against baseline design processes.

These evaluations will be documented through user feedback (if pilot-tested), annotated logs, and internal analyses.

Limitations

Throughout the project, we will document and reflect on:

- **Failure cases**: e.g., ungrounded outputs, hallucinations, ambiguity in instructions.
- **Gaps between LLM output and operational feasibility**.
- **Where structured templates outperform creative prompts**.
- **Where non-LLM alternatives might be preferable**.

This stage is critical for avoiding unjustified generalizations and for surfacing design recommendations that may extend beyond our own system.

Findings

This research does not aim to prove that language models can autonomously design drone shows. Instead, we aim to identify where they help, where they fall short, and under what conditions they might be useful. From that perspective, our expected findings fall into three broad categories:

1. **Insight into the Design Process:**

   We expect to develop a clearer understanding of how drone show designers currently translate creative intent into executable formations, where manual effort and iteration slow the process, and what types of tasks could benefit most from partial automation.

2. **Assessment of LLM Utility:**
    - What kinds of inputs are LLMs good at interpreting?
    - When do LLMs generate usable formations, and when do they fail?
    - Do structured prompts lead to more consistent, editable outputs than open-ended ones?
    - How does LLM behavior vary across models, and what does that imply for design?
3. **Design Recommendations for Human-LLM Collaboration:**
   We expect to generate recommendations on prompting strategies, interface patterns, and representation formats that improve usability, even if the LLM itself is not always reliable.

Even if the final system is limited in scope or capability, we expect this work to produce a clearer map of the problem space, the potential role of language-based agents, and the design trade-offs involved. The goal is not to validate a specific solution, but to refine the question itself, and give others a better place to start.

Discussion

[Your Discussion content here]

Conclusion

This project began not with the goal of building a novel system, but with a question: Does using large language models to guide drone swarm choreography meaningfully help anyone, and if so, how? Through a critical review of existing academic literature and commercial tools, we identified a recurring pattern: LLMs are typically used as translators for natural language, generating intermediate representations like waypoints or geometric shapes. However, these systems rarely consider how real-world professionals design drone shows, nor do they evaluate whether the outputs are usable, interpretable, or worth trusting.

Our proposed work addresses that gap. By developing a modular, prompt-driven agent and studying how users interact with it, we aim to uncover not just whether LLMs are technically capable, but whether they are practically valuable, for whom, in which contexts, and under what constraints.

We are not claiming that LLMs will revolutionize drone choreography. Instead, we aim to:

- Clarify where they genuinely reduce creative friction.
- Identify when structured design tools are preferable.
- Explore what an intelligent assistant for formation design might actually look like, not someday, but now, in modest and testable ways.

In doing so, we hope to contribute not just a prototype, but a set of grounded insights that future researchers, tool-builders, and designers can build on. Even if the output is imperfect, the process, rooted in dialogue, curiosity, and iteration, offers a different kind of value: a framework for building more human-centered, interpretable systems in creative robotics.

References

Jiao, A., Patel, T. P., Khurana, S., Korol, A.-M., Brunke, L., Adajania, V. K., Culha, U., Zhou, S., & Schoellig, A. P. (2023). Swarm-gpt: Combining large language models with safe motion planning for robot choreography design. *arXiv preprint arXiv:2312.01059*. https://arxiv.org/abs/2312.01059

Li, P., & Zhou, L. (2025). Llm-flock: Decentralized multi-robot flocking via large language models and influence-based consensus. *arXiv preprint arXiv:2505.06513*. https://arxiv.org/abs/2505.06513

Lykov, A., Karaf, S., Martynov, M., Serpiva, V., Fedoseev, A., Konenkov, M., & Tsetserukou, D. (2024). Flockgpt: Guiding uav flocking with linguistic orchestration. *arXiv preprint arXiv:2405.05872*. https://arxiv.org/abs/2405.05872

Pueyo, P., Montijano, E., Murillo, A. C., & Schwager, M. (2024). Clipswarm: Generating drone shows from text prompts with vision-language models. *arXiv preprint arXiv:2403.13467*. https://arxiv.org/abs/2403.13467

Vyas, V., Schuck, M., Dahanaggamaarachchi, D. O., Zhou, S., & Schoellig, A. P. (2024). Swarmgpt-primitive: A language-driven choreographer for drone swarms using safe motion primitive composition. *arXiv preprint arXiv:2412.08428*. https://arxiv.org/abs/2412.08428

Additional Data Analysis

[Additional analysis content]

Code Documentation

[Code documentation]

LLM Prompt Examples and Outputs

[LLM prompts and outputs]