Shape n' Swarm: Hands-on, Shape-aware Generative Authoring for Swarm UI using LLMs

Matthew Jeung
University of Chicago
Chicago, Illinois, USA
mkjeung@uchicago.edu

Anup Sathya
University of Chicago
Chicago, Illinois, USA
anups@uchicago.edu

Michael Qian
University of Chicago
Chicago, Illinois, USA
michaelq@uchicago.edu

Steven Arellano
University of Chicago
Chicago, Illinois, USA
stevenarellano@uchicago.edu

Luke Jimenez
University of Chicago
Chicago, Illinois, USA
lukejimenez@uchicago.edu

Ken Nakagaki
University of Chicago
Chicago, Illinois, USA
knakagaki@uchicago.edu

Figure 1: Shape-aware generative authoring process with Shape n' Swarm. Left to right: The user arranges a stick figure with the robots. The user instructs the system with speech. The system interprets the user-manipulated shape and generates a walking animation.

ABSTRACT

This paper introduces a novel authoring method for swarm user interfaces that combines hands-on shape manipulation and speech to convey intent for generative motion and interaction. We refer to this authoring method as *shape-aware generative authoring*, which is generalizable to actuated tangible user interfaces. The proof-of-concept Shape n' Swarm tool allows users to create diverse animations and interactions with tabletop robots by hand-arranging the robots and providing spoken instructions. The system employs multiple script-generating LLM agents that work together to handle user inputs for three major generative tasks: (1) thematically interpreting the shapes created by users; (2) creating animations for the manipulated shape; and (3) flexibly building interaction by mapping I/O. In a user study (n = 11), participants could easily create diverse physical animations and interactions without coding. To lead this novel research space, we also share limitations, research challenges, and design recommendations.

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Actuated Tangible User Interfaces, Swarm User Interfaces, Tangible Interaction, Large Language Models

ACM Reference Format:
Matthew Jeung, Anup Sathya, Michael Qian, Steven Arellano, Luke Jimenez, and Ken Nakagaki. 2025. Shape n' Swarm: Hands-on, Shape-aware Generative Authoring for Swarm UI using LLMs . In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28-October 1, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3746059.3747781

UIST '25, September 28-October 1, 2025, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/2025/09.
https://doi.org/10.1145/3746059.3747781

1 INTRODUCTION

Researchers in Tangible and Shape-Changing Interfaces [52] and Reconfigurable Robotics [74] have long envisioned materials that dynamically reconfigure in response to user intent. Visions such as Radical Atoms [23] and Programmable Matter [17] imagine malleable physical interfaces that interpret user inputs and adapt their form in real-time, enabling seamless, expressive interaction.

In pursuit of these visions, HCI researchers have developed a range of actuated tangible user interfaces (A-TUIs) using self-reconfiguring hardware and materials [15, 31, 40, 72]. Complementing these advances, researchers have proposed various authoring methods to enable user control over actuation. Notably, tangible manipulation techniques – where users manually shape the interface to record gestures and trigger actuation – have offered intuitive modes of interaction [43, 51, 61]. However, these approaches often fall short of the adaptive, intent-interpreting responsiveness envisioned in speculative materials like Radical Atoms or Programmable Matter.

To illustrate our vision, consider a speculative scenario involving a nine-year-old child without technical experience. The child interacts with a soft, clay-like material embedded with sensing and actuation capabilities. As they intuitively sculpt the material into the form of a dog, the material continuously senses changes to its shape. When the child gives a simple verbal command – "Wag his tail!" – the system draws on both the shape and the linguistic input to disambiguate the child's intent. By interpreting the manipulated form in the context of the speech instructions, the system identifies the relevant region and activates it to simulate tail-wagging behavior.

This scenario highlights the importance of *shape awareness* – incorporating speech with the direct shape-based manipulation for the system to interpret and respond to the user's goals. In this context, the user does not use shape in isolation to communicate intent to the system; instead, they combine it with speech as part of a broader interactive dialogue through which intent is expressed and interpreted. Based on this vision, we introduce *shape-aware generative authoring* as a generalizable authoring method and concept that integrates direct shape manipulation and speech to convey intent for generative motion and interaction for A-TUIs (Figure 2).

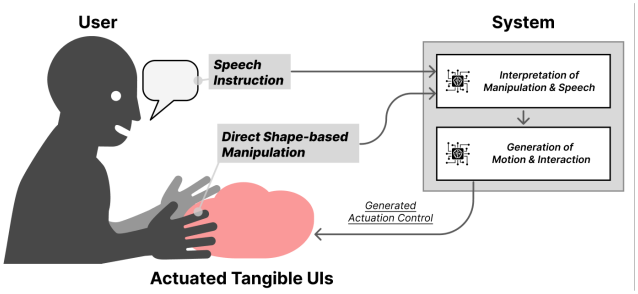

Figure 2: Concept behind Shape-aware Generative Authoring. The user inputs speech instructions and shape-based manipulation, which the system interprets for generating actuation.

In this visionary scenario, we theorize that the combination of shape and speech is an intuitive workflow for users, allowing the shape itself to carry semantic intent for robot actuation. To test this hypothesis, we introduce *Shape n' Swarm*, a proof-of-concept Swarm User Interface (SUI) authoring tool. Shape n' Swarm enables users to arrange tabletop robots and issue spoken commands to generate responsive, adaptive behaviors such as motion and interaction.

The proof-of-concept uses toio robots, an established platform in HCI research on SUIs [34, 41, 63], which are self-reconfiguring and easily arranged by hand on 2D tabletop surfaces – making them well-suited to our tangible authoring paradigm. We use a multi-agent Large Language Model (LLM) architecture [48, 69], where dedicated agents interpret both the physical arrangement of robots and natural language input to generate appropriate actuation via script-generation.

Through a user study ($n = 11$), participants used Shape n' Swarm to author expressive animated characters and scenes [10], mathematical and geometric physicalizations, and remote I/O control over physical objects. Our studies reveal that direct tangible shaping with speech offers unique benefits to user ideation, lowers the barrier of entry to SUI authoring, and enables expressive, diverse outcomes.

Through flexible interpretation of shape and speech, the concept of *shape-aware generative authoring* advances the vision of programmable interactive materials [17, 23] and opens new directions for reconfigurable TUIs that seamlessly blend user-oriented semantic shaping and system-oriented generative actuation.

1.1 Contributions

This paper makes the following contributions:

- We introduce a novel authoring approach for actuated tangible user interfaces (A-TUIs) based on shape-aware generative authoring, which combines shape-based direct manipulation and speech input to convey intent for generative actuation.
- We present a proof-of-concept authoring tool that integrates a multi-robot platform (Toio) with a multi-agent LLM architecture, demonstrating the feasibility of shape-aware generative authoring.
- We report findings from a user study ($n = 11$) demonstrating the unique strengths of our authoring approach, as well as the current limitations and areas for improvement of the developed system.

2 RELATED WORK

Shape n' Swarm builds upon prior works in (1) (Tangible) Authoring of A-TUIs and SUIs, (2) Multi-modal Authoring Tools, and (3) LLM-based Systems for A-TUIs and SUIs.

2.1 (Tangible) Authoring of A-TUIs and SUIs

To incorporate dynamic actuation into static TUIs [24], A-TUI research [47] has investigated shape-changing hardware, taking form factors including pin-based shape displays [15, 20, 59], shape-changing lines [39, 40], space-distributed mobile robots [31, 75], morphing sheets [46, 55], and pneumatic devices [18, 71]. Researchers have sought to enable intuitive user control over the actuation of

these hardware systems through various interaction modalities, without requiring users to have programming skills. Such examples include specialized GUI software [7, 25, 73, 75], hand or body-based gestures [2, 13, 28, 43, 75], sketching [12, 27], or direct tangible interaction [2, 16, 43, 51, 61].

Particularly, researchers have explored direct tangible interaction, or shaping, as an intuitive method for users to convey their intent for actuation. One such approach is 'kinetic memory' presented in Topobo [51], which allows users, even children, to 'record and play' motions to make tangibly-defined locomotive or expressive movements. Researchers have applied this 'record-and-play' hands-on approach to instruct movements for pin-based shape displays [43], plush toys [61], and tabletop robots [16]. Moreover, past studies have leveraged the affordance of a group of tangible blocks, enabling users to construct custom geometries through arrangement or assembly [22, 31, 32, 35]. This research highlights the intuitive nature of shaping as a user input modality. Specifically, shaping allows users to rapidly experiment while lowering the barrier of entry [51]. Our research seeks to harness these advantages explored in prior works.

While our approach similarly focuses on direct tangible manipulation and constructive assembly, we extend it by integrating speech-based instructions and LLMs. By utilizing LLMs to interpret user-manipulated shapes and speech to generate actuation, our approach incorporates adaptive intent-interpretation in contrast to 'record and play,' offering generative actuation behavior while preserving the affordance of tangible manipulation.

Further, Shape n' Swarm builds on prior research for authoring Swarm User Interfaces (SUIs). A subclass of A-TUIs, SUIs [31] capitalize on the reconfigurability of tabletop multi-robot systems, typically equipped with few wheels, to enable physical display and interaction. (In contrast to the term 'swarm robots', SUIs refer to multi-robot systems more broadly, without necessitating decentralized control). SUIs have been proposed for diverse applications in haptic design [29], constructive assembly [76], display with storytelling [10], shape-changing capabilities [64], and reconfigurable physical environments [41, 42, 75]. Researchers are actively exploring authoring methods for intuitive control over tabletop robot clusters, including AR-based sketching [27, 62], gesture control [28], and object-oriented, pre-scripted multi-modal control [68]. While a few recent papers presented the usage of LLMs for generative interactions with SUIs [19, 70, 77] (Section 2.3), our approach employs the affordance of SUIs, allowing users to flexibly arrange robots on 2D tabletop surfaces.

2.2 Multi-modal Authoring Tools

HCI researchers have sought to integrate speech with input modalities like gesture and touch for authoring digital systems or media [3, 30, 54]. For example, in DrawTalking, Rosenberg et al. [54] incorporate hand-drawing with speech to allow users to flexibly author animations. These approaches leverage the affordance of other input modalities (e.g., touch, gesture, drawing) combined with speech instruction, allowing for more intuitive authoring than speech alone. In contrast to our approach, where users generatively author new movements and interactions, DrawTalking mapped pre-programmed movement patterns tied to keywords.

Recently, to expand the flexibility of natural language (speech and text) as an approach to authoring digital systems and content, researchers have explored using LLMs to generatively respond to natural language inputs. LLMs have demonstrated potential in interpreting text-based user intent to advance tools for creating and authoring digital content [4, 37, 38], reducing the need for programming experience. For example, LLM-based systems have used user natural language prompts for creative coding in 2D digital art [1], sketching interactive storylines [5], enabling keyframe animation workflows [65], and creating and manipulating objects in virtual reality [9]. In this space, we observe the utilization of LLM-chaining and script generation [1, 9, 38, 48] to generatively translate user speech or text instructions into digital content, informing our approach.

Recent research in AI and HCI has sought to provide multimodal inputs to LLM systems beyond natural language [14, 21, 57, 67], merging research in multimodal digital authoring and LLM-based authoring tools. These approaches include the interpretation of mouse inputs [21], vision [14], and tangible interaction with robotic arms [67]. As part of our contributions, we explore how LLMs can interpret user manipulation of an A-TUI's shape, particularly tabletop robots, as another form of multimodal input to LLMs.

2.3 LLM-based Systems for A-TUIs and SUIs

Further, we build on prior research on applying LLMs to A-TUIs and SUIs. A closely related work to our approach is SHAPE-IT [48], which applies LLM tools to A-TUIs by enabling text-to-shape generative authoring on a pin-based shape display. It employs LLM-chaining with multiple agents to generate scripts that control the actuated hardware. In contrast to pure user-typed text input, we make the advancement of incorporating shape-based tangible manipulation as a means for users to author the behavior of A-TUIs.

In the context of swarm robotics and SUIs, researchers are increasingly incorporating LLMs for flexible authoring and control methods. Recent research in swarm robotics has sought to enable natural language as a method to coordinate multi-unit swarms [26, 36, 60]. Similarly, in SUI research, researchers have explored speech to generatively author motion and interaction for tabletop robots [70], including tabletop games [19]. TangibleNegotiation [77] combines speech with robot arrangement to generate images for art education, using simple motion for tangible feedback rather than shape-aware actuation. In contrast, utilizing LLM agents to interpret user arrangements of robots for generative actuation has yet to be explored, advancing research at the intersection of LLMs and SUIs.

Notably, our overarching idea of combining shape and speech for authoring does not center on the usage of LLMs. Rather, our LLM architecture enables our specific implementation, with the intent of proving the feasibility of our broader authoring method.

3 CONCEPT AND AUTHORING WORKFLOW

This section introduces an alternative approach to authoring swarm user interface. Instead of writing code to define robot positions, behaviors, and interactions, users physically arrange robots and describe their intentions aloud. We describe this approach as *shape-aware generative authoring*, a tangible and conversational authoring

framework designed to support embodied, incremental authoring of behaviors.

To illustrate this workflow (Figure 3), we walk through a hypothetical interaction between a young child and the Shape n' Swarm authoring tool. This narrative surfaces key authoring steps for the user and the underlying, behind-the-scenes system architecture and behavior, highlighting how different LLM agents collaborate to enable each step.

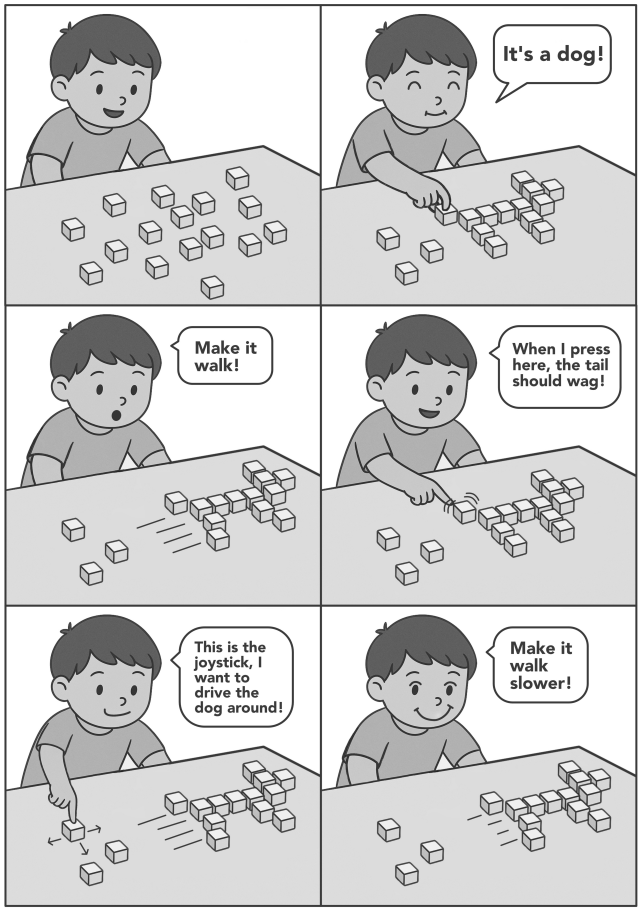

Figure 3: A child authors SUI behaviors using Shape n' Swarm. The child intuitively arranges robots into a dog shape. They animate the dog, make the tail wag when it is pressed, and designate a separate robot as a joystick. The joystick is used to move the dog formation and the animation is altered to make it slower, demonstrating tangible and verbal authoring without code.

3.1 Walkthrough

A child user approaches a table filled with small robots and begins to play. They arrange the robots into a familiar silhouette – a few at the front for the head, two sets of two for legs, a longer segment for the body, and one at the back for a tail.

Step 1: Defining the Configuration. While the system can sense the spatial arrangement of robots, it does not yet know what they represent for the user. The child announces, "It's a dog!".

System Behavior - The *Prompt Helper Agent* transcribes and reformats this speech and forwards it to the *Manipulation Interpretation Agent*, which combines the verbal description with the physical layout to label different robot groupings (e.g. head, legs, tail). These semantic labels enable meaningful behavior to be layered onto the formation.

Step 2: Creating Animation. The child says, "Now make the dog walk!"

System Behavior - This instruction is handled by the *Animation Agent*, which draws on the semantic structure established in Step 1. Recognizing that the request applies to the entire dog formation, the agent generatively defines a "walk" behavior and tailors it to the identified parts of the configuration – coordinating leg movement, body oscillation, and slight head bobbing. The animation script is then distributed across the relevant robots, bringing the dog to life with a rhythmic, synchronized gait.

Step 3: Creating Discrete Interaction. The child claps in delight. Next, wanting to directly interact with the dog, the child taps the robot at the tail and says, "When I press here, the tail should wag."

System Behavior - This is interpreted by the *Button-trigger Interaction Agent*, which identifies the pressed robot as a physical input and links it to the "tail" group. The agent generates a script that triggers a wagging motion in response to touch. The system binds this script to the designated robot, enabling playful, embodied, in-situ interactivity.

Step 4: Creating Continuous Interaction. The child then picks up another robot and places it beside the dog. Pointing to it, they say, "This is a joystick. I want to drive the dog around!"

System Behavior - Here, the *Input-mapping Interaction Agent* takes over. It recognizes that the user intends to use this newly designated robot as a continuous control input. Based on previous grouping data and the current prompt, it establishes a real-time mapping: movement of the joystick robot now controls the position of the entire dog formation. As the child pushes the joystick robot around the table, the dog-shaped group of robots follows accordingly, simulating locomotion.

Step 5: Refining Behavior. The child, noticing the dog is walking too quickly, follows up: "Make it walk slower."

System Behavior - The *Prompt Helper Agent* maintains the context of previous interactions and routes this refinement to the *Animation Agent*, which adjusts the gait parameters of the walking script. The dog now trots more slowly, responding to the child's preferences in real time.

3.2 Reframing Swarm User Interface Authoring

This hands-on, shape-aware authoring process transforms SUI behavior design into a tangible, iterative, and expressive activity. Instead of writing code, they physically compose robot layouts and describe behaviors in natural language. The multi-agent LLM architecture supports this fluid interaction with dedicated agents tailored to different interaction goals – spatial interpretation, animation, event binding, continuous input mapping, and prompt refinement.

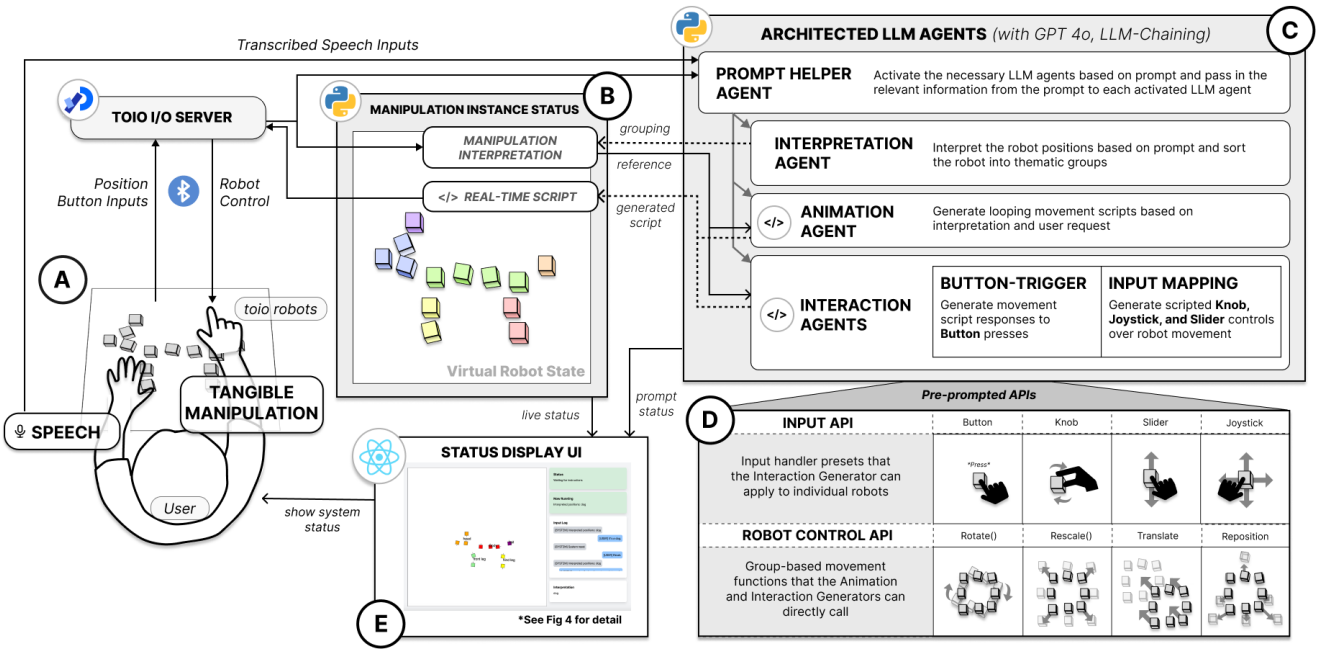

Figure 4: System Diagram: (A) Tabletop interaction between user and robots (B) Manipulation Instance Status (C) Architected LLM Agents (D) Pre-prompted APIs (E) Status Display UI

While we have illustrated this through the lens of a child creating a robotic pet using tabletop robots, the same framework enables broader use cases. For instance, an educator can set up a classroom demonstration on traffic management by arranging robots into an intersection and describing behaviors like, "Make cars stop when the light turns red." The system interprets the layout and verbal instructions, allowing for rapid prototyping and real-time iteration.

By enabling embodied interaction and incremental design, Shape n' Swarm makes SUI programming accessible to novices and experts alike, encouraging playful experimentation and more intuitive expressions of intent.

4 IMPLEMENTATION

As shown in Figure 4, Shape n' Swarm's implementation leverages a multi-agent LLM architecture, custom APIs, toio robots, and a frontend display to allow users to author SUIs through shape-based tangible manipulation and speech. This section details the technical implementation of Shape n' Swarm, including hardware components, software architecture, and user interface design.

4.1 Backend System

As shown in our system diagram (Figure 4 C), we developed a system featuring five LLM agents built on GPT-4o[45], a Robot Control API, and an Input API to handle each user request. We used prompt engineering strategies to maximize each LLM agent's accuracy, chosen based on their demonstrated effectiveness in improving LLM tasks:

(1) Rule-based Prompting: Each LLM agent is instructed with a clear input and output format and custom rules for generating its output. The movement LLM Agents are prompted with a clear description of the Robot Control API and Input API.
(2) Guided Reasoning: Each LLM agent is provided with detailed step-by-step instructions to reason through each user prompt.
(3) Example-driven Prompting: Each LLM agent is provided a diverse range of examples (>10) featuring the initial information and final output.
(4) LLM-Chaining: Each speech instruction is filtered through the Prompt Helper Agent to clarify and format the instructions into sub-prompts. Further, generated movement scripts are built upon the manipulation interpretation for 'shape-aware' actuation.

4.1.1 LLM Agents. When designing our system, we identified sub-challenges based on the authoring workflow established in Section 3. For each sub-challenge, we architected dedicated LLM agents connected via LLM-chaining [48]. These agents are prompted such that generated motion or interaction reflects a thematic understanding of the user-manipulated shapes and the user's intent conveyed through speech. Additionally, each agent incorporates the history of prior user inputs and system outputs to allow users to make quick adjustments through speech. This approach allows users to make quick, follow-up alteration prompts to iterate upon specific animations or interactions. Below, we outline the role, implementation, input, and output of each LLM agent.

(1) Prompt Helper Agent. To be friendly to novice users, the Prompt Helper Agent processes diverse, conversational speech instructions and determines the intent behind the instructions (interpret shape, author animation, author button-trigger interaction, author input-mapping interaction, or follow-up to previous instruction). In our system, the user speaks into a microphone to give speech instructions, which are transcribed using the open-source OpenAI Whisper API (whisper-1 model [50]). The Prompt Helper Agent serves as a prompt preprocessor, filtering speech prompts to ensure grammatical clarity.

- **Input:** Transcribed speech instructions from the user.
- **Output:** Formatted string instructing the system on which LLM agents to activate (one or multiple in sequence), containing sub-prompts for each activated LLM agent based on the initial instructions.

(2) Manipulation Interpretatation Agent. The Manipulation Interpretation Agent builds an understanding of the user-manipulated shape formed by the robots, handling both brief descriptions (one to two-word names) and long descriptions identifying specific parts. This agent identifies key groups composing the overall shape (e.g. the body parts of a stick figure) and sorts the robots into these groups. Manipulation interpretation always precedes any animation or interaction.

- **Input:** Formatted prompt from the Prompt Helper Agent containing (1) the robot positions and (2) the user's description of the manipulated shape.
- **Output:** A labeled thematic grouping of the robots (Figure 4 B), which we define as a *manipulation interpretation*.
- **Post-processing:** The thematic grouping is displayed through each robot's LED-indicator and the front-end display, for user review and adjustment.

(3) Animation Agent. The Animation Agent enables users to author motion and animation, handling both simple movement requests (e.g. move in a straight line) and abstract, complex requests (e.g. dance happily). This agent determines if, when, and how to actuate the robots given their status defined by the manipulation interpretation, writing a Python script that calls the Robot Control API 4.1.2.

- **Input:** Formatted prompt from Prompt Helper Agent containing (1) the manipulation interpretation, (2) robot positions, and (3) user instructions for an animation.
- **Output:** Python animation script that coordinates the motion of robots using the Robot Control API.
- **Post-processing:** The system executes the Python script as a separate thread. Each function call made by the Python script outputs a set of target positions for the robots, which is sent to the robots via Bluetooth in real-time.

(4) Button-trigger Interaction Agent. The Button-trigger Interaction Agent enables users to create tangible interactions by linking simple or complex movements to button presses of one or more robots. Based on the manipulation interpretation, this agent identifies which robots should act as inputs and which robots to actuate and scripts the motion of the output robots.

- **Input:** Formatted prompt from the Prompt Helper Agent containing (1) the manipulation interpretation, (2) user instructions for the button-trigger interaction, (3) robot positions, and (4) button-selected robots.
- **Output:** Python script that defines the output animation or motion, and an input-to-output relationship.
- **Post-processing:** The system calls the Input API to map the selected robot(s) as button trigger(s) for the generated script, such that whenever the input robot is pressed, a new thread runs the movement script.

(5) Input-mapping Interaction Agent. The Input-mapping Interaction Agent allows users to define the movement of robots correlated to granular updates to an input robot's position and orientation, which gives users real-time fine motion control over the robots. In contrast to script generation, this agent interprets the manipulation interpretation and user prompt to determine input robots, output robots, and the nature of the I/O relationship.

- **Input:** Formatted prompt from the Prompt Helper Agent featuring (1) the manipulation interpretation, (2) robot positions, and (3) user instructions for the input-mapping interaction, and (4) button-selected robots.
- **Output:** Formatted string featuring (1) input type (joystick, slider, or knob), (2) output motion (translate, rescale, or rotate), (3) the parameters scaling the input-to-output relationship from the prompt, and (4) output robots.
- **Post-processing:** The Input API detects live position updates from the input robot. When the Input API detects an update, the system automatically calculates target positions for the output robots in real-time.

4.1.2 Robot Control API. We developed a Robot Control API to facilitate LLM agent control over actuation, defining movement functions that output real-time robot target positions (Figure 4 D). Four primitive movement functions control movement: *translate, rotate, rescale*, and *reposition* (Figure 4 D). The first three move a group of robots based on parameters (*translate:* x, y; *rotate:* angle; *rescale:* scale). Multiple group-based function calls can be made concurrently, allowing for complex movements. In contrast, *reposition* generates a set of target positions for every robot, enabling more detailed control over movement. Every movement call includes a movement speed parameter, set by the LLM agent based on user instructions. We prompt each movement LLM agent with detailed instructions on each movement function and diverse examples. To generate actuation, movement LLM agents generate scripts that make function calls to the Robot Control API.

4.1.3 Input API. We also developed an Input API that uses individual robots as one of four input types: button, knob, slider, or joystick (Figure 4 D). The Interaction Agents call the Input API to initialize the input robot. The Interaction Agents set parameters for the input type, the output robot(s), and the output movement, generating an input instance, which will continuously detect the specified tangible input until it is reset.

4.2 Toio Robots

We use toio robots, $2.85 \times 2.85cm$ cube-shaped units that navigate independently across gridded mats. Each robot is equipped with a

downward-facing scanner for real-time position tracking, a built-in button for selecting robots and triggering interactions, a multi-color LED indicator for displaying group status, and a built-in speaker for notifying the user of group and selection status changes.

As shown in Figure 4 A, each robot is connected to the system via Bluetooth. While up to 12 toio robots can directly connect to a computer via Bluetooth, we utilize multiple Adafruit Feather Bluetooth bridges to establish a Bluetooth connection with up to 50 robots. The Processing program tracks each robot's status and position and relays motion commands to the robots. The Processing program is connected via an Open Sound Control (OSC) server to the main Python program. The main Python program exchanges live status, positions, and target positions with the Processing program.

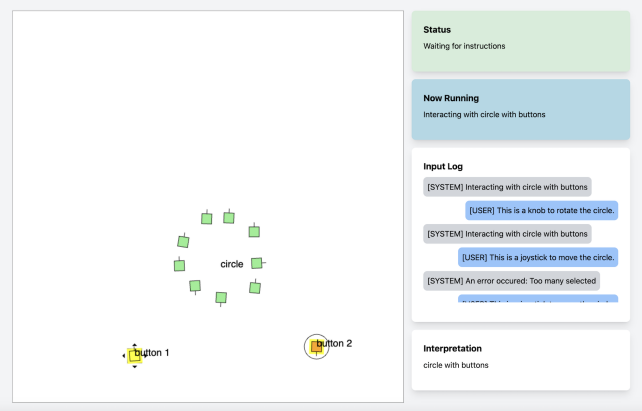

Figure 5: Frontend Display: showing a manipulation instance of a circle and two buttons. The left button is mapped to be a joystick to control the circle, and the right button is mapped to be a slider to control the circle.

4.3 Frontend System

As shown in Figure 4 E, we use a React web application to display 1) the robot state (positions, grouping, interaction mapping), 2) a log of user inputs and system responses, and 3) the system status (waiting for instructions or currently processing request). The React web application communicates with the main Python program via a Flask server. Optionally, the user can interact with a simulated version of Shape n' Swarm directly through the frontend display through mouse drag and click inputs.

5 TECHNICAL EVALUATION

We performed a technical evaluation to determine the best LLM model and assess the performance and scalability of our system.

5.1 Test Cases

. To gather test cases, we simulated the system in a virtual environment and recorded every request made to the system throughout our user study.

Success Criteria for Test Cases. Our technical evaluation checks for significant errors rather than the subjective quality of each output. While the user study goes further into the quality of outputs, our test cases provide a baseline understanding of how often the system returns a valid, compilable output.

- Prompt Helper: The test case is deemed a success if the Prompt Helper Agent returns a string that matches the requested format and each action call in the string is valid.
- Manipulation Interpretation: The test case is successful if (1) all robots are assigned to a group, (2) no robots are assigned to multiple groups, (3) no non-existent robot is assigned to a group, and (4) the output follows correct formatting.
- Animation: The test case attempts to compile and run the entire output script, checking whether each command is a valid Robot Control API function call.
- Button-trigger Interaction: The test case follows the same approach as the Animation test cases. Additionally, the test case only passes if the mapped input and output robots exist and the string describing the mapping is properly formatted.
- Input Mapping Interaction: The test cases check whether the input and output robots exist and form disjoint sets, and that the I/O relationship is valid (joystick, slider, or knob, as in Section 4.1.3).

5.2 Model Comparison and Evaluation

To determine the best LLM model for Shape n' Swarm, we evaluated LLM models for commercial and research applications (GPT-4o, Claude-3.7-Sonnet, and Llama-3.3-70b) on both speed and success rate. We the Prompt Helper Agent, Manipulation Interpretation Agent, Animation Agent, Button-trigger Interaction Agent, and Input-mapping Interaction Agent with each of the three candidate LLM models. For each agent, we randomly selected 20 participant requests from the user study, resulting in a total of 100 test cases.

As shown in Figure 11, GPT-4o demonstrated the fastest mean load time across all tasks. Further, GPT-4o had the strongest performance with passing test cases, illustrated in Appendix A. Based on this comparison, GPT-4o offered the best trade-off between speed and success rate.

5.3 System Scalability

To evaluate our system's scalability, we measured how the success rate and load time were affected as the number of robots increased. We hand-created 10 arrangements with 10, 20, 30, and 40 robots, mirroring the prompts and designs found in the user study, forming 40 unique arrangements. We ran sample prompts corresponding to each arrangement through the Manipulation Interpretation Agent and Animation Agent, measuring performance with the test cases. Example arrangements and output animations with 30 and 40 robots are included in Appendix B.

As Figure 6 shows, the system performs best with 10 robots, with a mean success rate of 97.5%. With 20 robots, the system maintains a mean accuracy of 87. 5% with a mean load time of 1.71 seconds. However, large swarms require further optimization to maintain a high success rate and low response times, as both metrics show noticeable degradation. Such limitations are further addressed in Section 7.2.2.

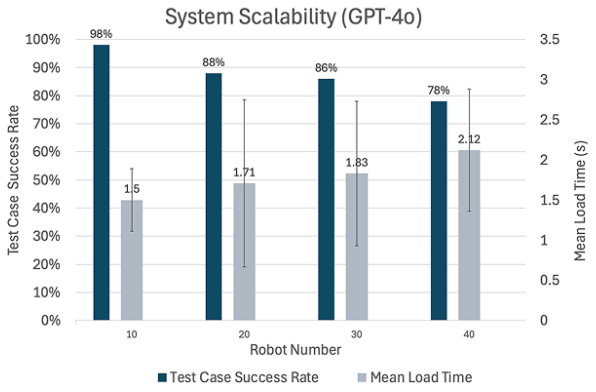

Figure 6: System Scalability: Test case success rate and mean load time at varying robot counts.

6 USER STUDY

Apart from gauging the overall user experience with our authoring method, our user study focused on two core objectives: (1) evaluating the quality and characteristics of system-generated responses to freeform user inputs, and (2) understanding how participants engaged with the shape-aware generative authoring process, including the creations they attempted. Centered on these objectives, we designed two open-ended tasks: (Task 1) create any animated outcome, and (Task 2) move the target object by authoring an interaction, allowing us to observe a wide range of interactions and behaviors across participants. These findings surface key opportunities and limitations of the authoring method, which we discuss in detail in a later section.

6.1 Procedure

Participants and General Procedure: We recruited 11 participants through university social media channels and screened participants to ensure diverse self-rated coding experience levels (6 high, 2 medium, 3 low). Our study was approved by our university's Institutional Review Board (IRB) (IRB24-1325). Each study session took approximately 30 minutes to complete. The user evaluation involved a preliminary onboarding task (Task 0) and two primary tasks (Tasks 1 & 2). In Tasks 1 and 2, we asked participants to "think out loud" by communicating their ideas for creation and reactions as they interacted with the system.

Task 0: This onboarding task familiarized participants with the system. Using a user manual, participants followed step-by-step instructions to build and animate a stick figure and create interactions by linking it to a button. They then constructed a rectangle and experimented with different controls (knob, slider, joystick). Finally, participants reset the system and had 3 minutes for free exploration.

Task 1: Participants were asked to design an expressive, practical, or educational application using the system. The application had to meet three requirements: 1) use all 11 robots, 2) include one animation, and 3) feature two interactions with the swarm. The user manual was available for reference during the task. The primary goal of this task is to allow users to create open-ended applications using shape-aware generative authoring. We aim to extract the nature of the applications (Section 6.2) that participants attempt, learning how the system supports or undermines their intent. The three requirements are only intended to encourage the participants to attempt complex applications that justify the use of our system.

Task 2: This task required participants to create a system that rotates a provided L-shaped block without directly touching the block or placing robots directly adjacent to the block at the onset. The primary goal of this task is to assess the system's flexibility in supporting diverse approaches for problem solving. Shape-aware generative authoring enables participants to employ creative, non-prescribed methods – using direct manipulation and voice commands – to rotate an L-shaped block in various ways, tailored to their unique strategies and context. We discuss these diverse approaches in Section 6.2 and the system's strengths and weaknesses in supporting them.

Post-study: After the tasks were completed, participants filled out a Likert-scale questionnaire (Figure 9) and participated in a semi-structured interview to describe their reactions and impressions of the system. The interview allows us to capture a qualitative understanding of how participants experienced our shape-aware generative authoring method.

6.2 Results - Overall Task Outcomes

In this section, we describe (1) the types of creations in Tasks 1 and 2, illuminating potential applications of our authoring method, and (2) the primary challenges users encountered. Table 1 lists all creation attempts and our analysis of the system output results, which are detailed below.

6.2.1 Approaches to Task 1. In Figure 7, we detail several creations that the participants made during Task 1 of the user study.

Task 1 demonstrates the expansive potential for actuation and interaction with SUIs made possible through shape-aware generative authoring. In under 15 minutes, participants authored diverse creations, as shown in Figure 7. Many participants tended towards creations that could be considered artistic or playful, focusing on entertainment or expressive value. For instance, P7's giraffe was authored to nod its head and walk, while P2's skater and half-pipe (Figure 7) showed a skater riding the half-pipe. Educational elements emerged in P4's two lines, which performed addition and merged the two lines, serving as an interactive tool for understanding mathematical concepts. P9's number display aimed to allow users to increment and decrement numbers. Practical elements were evident in designs such as P5's excavator, which attempted to showcase the functionality of a real-world machine and allowed users to explore mechanical principles through interactive control. Overall, participants rapidly learned the authoring process, producing diverse creations highlighting the flexibility of our authoring method.

6.2.2 Approaches to Task 2. Figure 8 illustrates several participant approaches to Task 2 of the user study.

Participants took diverse approaches to achieve Task 2, where they were asked to rotate an L-shaped block without directly touching the block or placing robots next to it. In under ten minutes, participants authored varied designs to accomplish this moderately

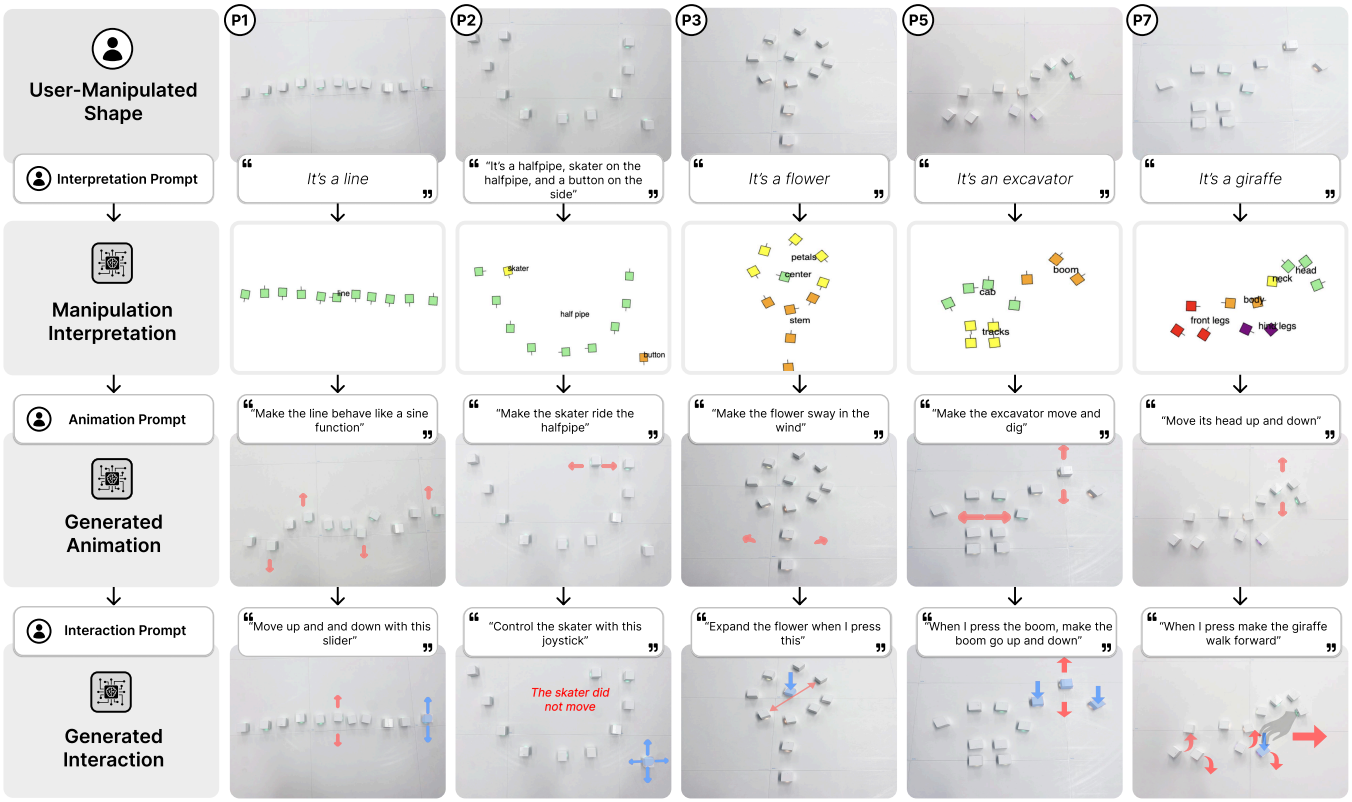

Figure 7: Examples of Task 1 creations (left to right): Line to demonstrate sine curve (P1), skater and half-pipe (P2), flower (P3), excavator (P5), and giraffe (P7)

complex task. P3 and P5 created a grabbing mechanism to simulate a claw-like structure to move the block. P6 relied on two pushers to turn the block and also apply force to move it. P9 introduced a more complex design with a "pivot point pusher", where one robot served as a stationary pivot while another pushed the block, leveraging rotational movement. These varied strategies reflect the potential of our authoring method to enable problem-solving.

6.2.3 Authoring Challenges. Based on user feedback and our observations of the system, we subjectively identified three patterns of challenges that users encountered in the authoring process, as color-coded in Table 1. These three challenges included: *grouping errors*, *movement inaccuracies*, and *understanding gaps*. These three errors impacted participants' ability to successfully engage with the system and complete their tasks, as detailed in the following sections.

Grouping Errors. Participants faced challenges due to grouping errors that impacted their ability to perform tasks effectively. For example, P2 struggled when the joystick button was incorrectly grouped as part of the half-pipe. This misinterpretation resulted in no response when the joystick was moved, discouraging P2 from further interactions. Similarly, P10 attempted to create a box and L-shaped pusher but faced errors when the system grouped inputs as part of the shapes, causing a malfunction and leading to abandonment of the task. These grouping errors highlight the need for the system to interpret user input more accurately and provide more robust options to adjust groupings when necessary, which is further discussed in Section 7.2.1.

Movement Inaccuracies. Participants encountered issues with precision, unintended movements, and incomplete executions in Task 1. P8 attempted to flank an opposing army line, but the robots only moved partway to position themselves behind the opposing army, capturing the general intent, but rendering the interaction incomplete. P9 experienced problems with shape transformation; their effort to reconfigure a "1" into a "2" resulted in vaguely recognizable, but distorted and collision-prone animations. These movement inaccuracies further underscore the issues of LLM reliability and readability, discussed further in Section 7.2.1.

'Understanding' Gap. During task 2, participants encountered a few challenges in controlling the mechanism they designed to manipulate the L-shaped block (Figure 8). Although the participants were technically successful, there was a gap between the participants' understanding of the input controls (knobs/sliders/joysticks). For example, P2 used an L-shaped pusher that could only nudge the block slightly because they mapped a discrete button input to the movement instead of a continuous knob, resulting in choppy movements. While some of these issues can be solved by improving the GUI and providing guidance to users, these challenges across participants echo the question – how do we communicate input

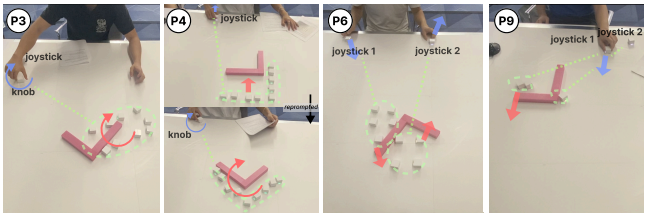

Figure 8: Examples of Task 2 strategies (left to right): grabber design (P3), L-shaped pusher (P4), two pushers (P6), and pusher and pivot (P9)

affordances of interactions authored by the user through our approach? This is discussed further in Section 7.3.3.

6.3 Participant Reactions to the Authoring Experience

We aimed to identify recurring patterns in (1) participant engagement with our authoring process and (2) system output quality through collecting post-study surveys and interviews. The following summarizes the key themes from participant feedback.

6.3.1 Survey Results. The survey reveals that participants felt that both modes of input, tangible shaping and speech, contributed to their control, though their impressions varied across different steps of the authoring process. Figure 9 illustrates participants' Likert scale responses following the study. Users reported that both hand-arranging the robots and using speech contributed to their sense of control, as reflected in the average ratings (AR) for Q1 and Q2: 6.1 and 5.8 out of 7, respectively. While these inputs were generally perceived as empowering, users expressed mixed reactions to different stages of the authoring process. Participants had mixed opinions about the system's ability to interpret and group their manipulated shapes (Q3 AR: 4.2). In contrast, users generally felt more positive about authoring animations and interactions (Q4 AR: 5.3 and Q5 AR: 5.5). Further, users generally felt that the system understood their intent (Q6 AR: 5.4). Finally, users showed positive to neutral results about whether the system enables easy transference of ideas (Q7 AR: 4.8). To further unpack these survey findings, we identify key trends in interview responses.

6.3.2 Shape Implies Functionality. Participants expressed that the ability to manipulate the arrangement of robots enhanced their control over robot behavior. Interestingly, the survey showed that users considered arranging the robots to be the more important control method compared to speech (Figure 9). Users observed that forming the shape of the robots helped implicitly define their functionality. P5 notes, "It felt like there was a connection between what I placed and what kind of functionality would be expected from it," while P6 observed that hand-arranging "seemed to be the most intuitive way to control the robots." Participants noted that the authoring tool was user-friendly to learn, echoed by P10, who found the tangible manipulation "very easy to learn,' and P6, who described tangible manipulation as a "direct transference of ideas." These comments help confirm our initial hypothesis that tangible shaping can be used to convey intent for actuation, as outlined in Section 1.

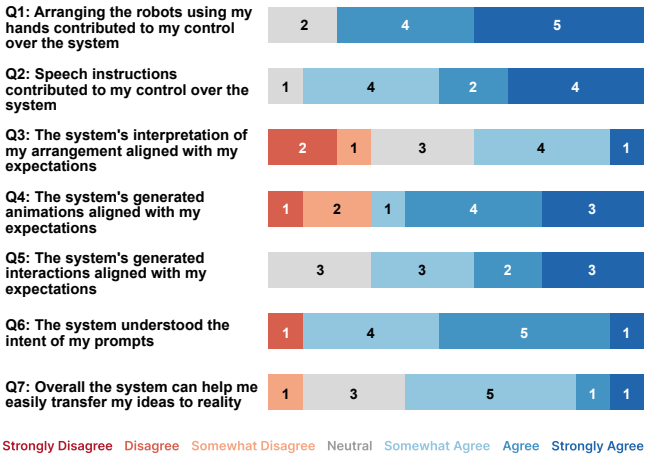

Figure 9: Post user evaluation questionnaire results.

6.3.3 Impact of LLMs: Magical When it Works, Frustrating When it Doesn't. The interplay between the speculative potential and the inherent opaqueness of LLM-based systems created a duality of experiences for participants, oscillating between moments of awe and frustration. Many participants marveled at the system's ability to intuitively interpret and execute complex commands. For example, P2 was "blown away" by the system's flawless execution of a challenging task on the first attempt, describing the system as "close to magic." However, this magic came with its own set of challenges, as discussed in Section 6.2.3. These missteps were exacerbated by the lack of transparency in black-box LLMs, making debugging more difficult. Particularly, our post-study survey illuminated the need for further transparency and user agency for the manipulation interpretation step. While our system's nature often delighted users, its opaqueness posed barriers to consistent and controllable interaction, a tension that will be explored further in Section 7.2.1.

6.3.4 Thinking through Shaping. The open-ended nature of generative authoring and the absence of clear constraints can make it challenging to settle on a specific direction. In Tasks 1 and 2, we often observed participants idly re-arranging the robots at random until arriving at an idea to pursue, as if brainstorming through manipulations to spark inspiration (which could be referred to as hand-storming). Hand-shaping the robots helped participants think through tasks and refine their thoughts into specific ideas. P3 observed, "once you put it onto a physical medium, then it actually makes more sense." Similarly, P11 mentioned that the hand manipulation "helped me visualize a lot better." The ability to quickly hand-adjust designs helped fuel ideation and iteration, as P11 noted, "I could easily change my designs as soon as I had moved the robots" and P7 reflected, "You could let your imagination go and rearrange it however you want to a very high capability." These observations provide unique insight into how tangible interaction can assist user ideation when working with LLMs. Recent research in Human-AI Interaction has explored LLM-based systems that nudge users to

Table 1: A summary of results from both tasks of the user evaluation. Black indicates task completion without encountering major challenges (defined in 6.2). Light Blue indicates a movement inaccuracy (defined in 6.2.3). Purple indicates a grouping error (defined in 6.2.3). Orange indicates an understanding gap (defined in 6.2.3). Yellow indicates that the participant did not attempt the task.

	Task 1				Task 2
	Creation	Animation	Interaction 1	Interaction 2	Strategy
P1	Line	Sine wave	Standing wave	Move up/down	U-shaped pusher
P2	Skater, Half-Pipe, & Button	Ride half-pipe	Move w/ button	No Attempt	L-shaped pusher
	Smiley face	Change expression			
P3	Flower	Swaying	Expand flower	Contract flower	Grabber
P4	Two lines	Addition	Move left	Move right	L-shaped pusher
P5	Excavator	Move & dig	Move boom up & down	Drive forwards	Tongs
P6	Tree	Leaves shaking	Cut tree in half	Move up	Two pushers
P7	Giraffe	Move its head	Walk forwards	Walk backwards	Reverse L-shaped pusher
P8	Two lines of soldiers	Attack each other	Flank line	No Attempt	Two-line pusher
P9	Number display w/ buttons	Change 1 to 2	Increment	No Attempt	Pivot point & pusher
P10	Square, triangle & 2 buttons	Rotate square	Rescale square	Rotate triangle w/ knob	Box & L-shaped pusher
P11	Stick figure & basketball	Jump for ball	Jumping jack	No Attempt	Two-line pusher
	Two racers & a finish line	Race to finish			

think critically [8, 66]. We speculate that the intersection of tangible shaping and AI-augmented ideation may have potential in this realm and should be explored in future work.

7 DISCUSSION, LIMITATIONS, AND FUTURE WORK

In this paper, we introduced shape-aware generative authoring, showcasing the method's potential through the proof-of-concept Shape n' Swarm authoring tool. The user study showcased pathways to exciting future applications, but also the need to address LLM and hardware-related limitations through further work.

7.1 Potential Application Spaces

The user study highlighted the unique strengths of our authoring method, including rapid customizability and embodied interaction, enabling both creative and functional applications with SUIs. Although this section focuses on tabletop robots, we believe the underlying concepts are broadly applicable to other A-TUI hardware platforms.

7.1.1 Education. As a flexible, embodied learning tool, our approach merges generative authoring with building blocks, a familiar mode of play for children (Figure 10 A). For example, P10 in our user study defined a triangle, square, and L-shape, showing geometric and alphabet learning opportunities. Children can develop their understanding of the world by building recently learned concepts and asking their creation to behave as it would in real life. Previous researchers [77] have observed that the tangibility of tabletop modular robots and the flexibility of LLMs pair well for interaction with children.

7.1.2 Environment Manipulation. With shaping, users can form geometries to fit unique tasks, then use speech to rapidly actuate the arrangement, building on previous research focused on manipulating the surrounding environment with SUIs [68]. As demonstrated in Task 2 of the user study (Figure 8), users leveraged the flexibility to define custom shapes to rotate a moderately complex shape within minutes of being introduced to the system. As shown in Figure 10 B, we believe that the ability to manipulate the robots into complex shapes can provide additional flexibility and motor control when manipulating the surrounding environment. We envision various use cases, particularly in accessibility contexts, where the system could enable individuals to control out-of-reach objects in a highly customizable way.

7.1.3 Interactive Storytelling. Shaping materials into characters and telling stories with speech is a natural form of creative play. Our approach leverages generative actuation to animate these user-crafted forms, bringing them to life through the user's voice. As shown in Figure 10 C, when the user creates multiple figures or objects simultaneously, they can begin to tell a story about their own creations [10]. For example, P11 created two racers racing towards a finish line, and a stick figure playing basketball. P8 created two lines of soldiers fighting each other, describing their experience as "lots of fun." The user can build multiple characters and environments for those characters, where every component within the story has its unique behaviors. The user can continuously redefine interactions

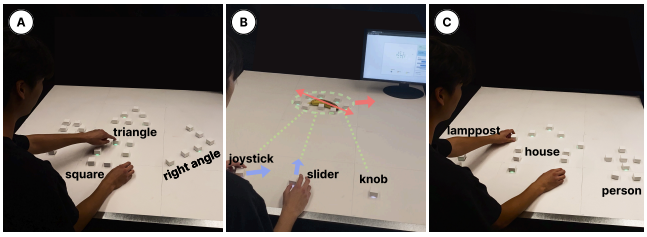

Figure 10: Application space examples (A) Educational geometric building blocks (B) Grabbing a banana with remote object moving (C) Interactive storytelling with person, house, and streetlamp.

and generate new animations, allowing the user to tell an evolving, interactive story.

7.2 Proof-of-Concept Limitations and Design Recommendations

Our technical evaluation and user study also revealed limitations in our proof-of-concept authoring tool, which we discuss below. Further, we identify design recommendations to address these limitations.

7.2.1 Transparency and Determinism of LLMs. Participants in the study expressed the need for greater LLM reliability and transparency when authoring manipulation interpretations, movement, and interactions. P9 emphasized the importance of making the system "more deterministic [and] reliable in understanding what it's doing." Further, the system often struggled to incorporate follow-up feedback after its initial response, limiting the specificity of user control. P11 was "negatively surprised with its adjustments to feedback," noting that the system struggled to incorporate corrections after initial misinterpretations. Overall, LLM unpredictability and lack of transparency generated issues in control and reduced user agency.

Incorporating manual control options for LLM-based systems remains a promising solution, as echoed by Ben Shneiderman [58]. P10 suggested a manual regrouping feature as a faster and more detailed method to correct the system's manipulation interpretation, compared to follow-up speech instructions. Further, the frontend display could be adjusted to show sliders to control key parameters of generated animations and interactions, such as movement speed and size of the movements. Additionally, prompting LLM agents to include an explanation of their reasoning along with each output could assist users in determining their follow-up instructions.

7.2.2 Robot-related Scalability and Reliability. Our technical evaluation in Section 5 showed that our LLM agents encountered challenges when scaling beyond 20 robots. One potential solution is to adaptively swap our example prompts based on the current number of robots, to optimize response quality when the robot count increases. Further, working with higher numbers of robots may significantly change human affordance and general interaction outcomes, which should be further explored through research. Although our user study supported diverse creations using 11 robots, optimizing the system when handling larger numbers could unlock even more complex and varied creations.

We also encountered issues with the robots becoming stuck on each other when their paths converged. In our user study, collisions mainly surfaced when users attempted to pack robots together in their arrangement tightly. As a next step, we hope to further incorporate robot collision prevention techniques [33], integrating automatic path planning with every target position. Further optimization to prevent collisions could yield more reliable animations and interactions. Finally, while our system uses toio hardware, our approach can be applied to other tabletop multi-robot systems, as our architecture uses generic X-Y coordinate inputs and outputs.

7.3 Future Work and Implications for the General Concept

Shape-aware generative authoring opens up a range of opportunities for future research. We outlines key directions for expanding and optimizing this novel authoring approach for A-TUIs.

7.3.1 Shape-Aware Generative Authoring for New Platforms. We believe our approach of shape-aware generative authoring should be explored with diverse SUI and A-TUI hardware platforms. We developed Shape n' Swarm with the vision of a clay-like, moldable material responsive to tangible manipulation [17, 23]. Future research should pursue these visions by applying shape-aware generative authoring to new, tangibly manipulable platforms. Increasing the granularity and scale to 100 or 1000 robots [56] would improve shape resolution and enable more complex creations. Additionally, an implementation with non-tabletop robots capable of navigating varied surfaces would broaden its applicability. Incorporating our approach with a swarm of drones [36] would allow for manipulation and actuation in 3D space. Similarly, a 3D constructive block hardware [53], folding plane-based systems [46, 55], and pneumatic shape-changing systems [18] would enable flexible manipulation of three-dimensional forms, moving closer to future visions. While Shape n' Swarm demonstrates shape-aware generative authoring in a tabletop swarm user interface setting, our authoring approach has much broader applicability to new platforms.

7.3.2 Optimizing Shape Awareness. One key step in the shape-aware generative authoring process is the system's *manipulation interpretation*, the semantic breakdown of user-manipulated shapes based on speech instructions. While our proof-of-concept's usage of an LLM agent enabled flexible interpretation of user-manipulated shapes, further optimization of this step could improve the reliability of the authoring process. One pathway is to integrate our LLM agents with knowledge maps [11], which can provide structured representations of the user's robot arrangements. Incorporating computer vision [45, 49] would enable the LLM to make visual inferences rather than text inferences, potentially improving the system's understanding of user-manipulated shapes. Further, robotics and AI research has explored non-LLM techniques to semantically interpret images through semantic segmentation [6] and the interpretation of natural language with images for robotic motion planning [44]. Adapting such research streams to interpret the shape of A-TUIs could improve the accuracy of this step. We

believe the potential demonstrated by our proof-of-concept justifies further research into refining the shape-aware authoring approach.

7.3.3 Feedback and Interaction Design for 'Shape-Aware Generative Authoring'. Shape-aware generative authoring requires information to be communicated to the user throughout the authoring process. For example, our proof-of-concept communicated the grouping of robots and the user-authored interactions through a screen display. The best interaction design practices for this authoring method have yet to be identified. Such practices should maximize the user agency and the readability of LLM outputs. Future systems could go screen-less, further emphasizing a non-digital approach to authoring. For example, groupings and interactions could be communicated through projected overlays [34], while robots could be strategically actuated to give haptic feedback. Our user study identified a few next steps towards these goals, including manual control options and better signifiers for the affordance of generated interactions. These techniques would be particularly useful for editing system outputs, providing intuitive alternatives to simple follow-up speech instructions. Future work should further investigate the open research questions of what information to convey and the most effective communication methods, in the context of shape-aware generative authoring.

7.3.4 Expanding Multi-modal Interaction to 'Motion-Aware' Authoring. Shape-aware generative authoring has the potential for input modalities beyond interpreting static tangible manipulation and speech. For example, beyond 'shape-aware', such a system would allow for 'motion-aware' actuation generation, enabling the interpretation of dynamic changes to the shape to generate actuated behavior. This approach could be adapted to thematically interpret hand-recorded, shape-changing motions, on top of static shapes.

7.3.5 User Study with Children. While our user study with adults revealed the open-ended potential of our approach for creating animations and interactions, we are particularly interested in extending this work to children, inspired by our initial vision and scenario (Figure 3). We have previously exhibited Shape n' Swarm in public science fair exhibitions, where many children enjoyed interacting with the system. Much like constructive assembly toys such as LEGO, we observed some children getting obsessed with the system during the exhibition, highlighting the strong potential for exploring how our approach might support and expand children's creative expression. Deploying a long-term qualitative study of our approach with children would investigate this question.

8 CONCLUSION

This paper presents shape-aware generative authoring, a novel approach to authoring swarm user interfaces that combines hands-on shape manipulation and speech to communicate intent for generative actuation. We created the Shape n' Swarm authoring tool as a proof-of-concept for this approach, allowing users to author animations and interactions with tabletop robots through tangible manipulation and speech. Our user study showcased this workflow's flexible, intuitive nature and demonstrated the diverse range of user creations and applications made possible. This paper opens the novel HCI research space of "shape-aware generative authoring," a broadly applicable approach for actuated tangible user interfaces with exciting opportunities for future exploration.

ACKNOWLEDGMENTS

We acknowledge the Quad Undergraduate Research Scholars Program at the University of Chicago for its support. From the Actuated Experience Lab, we deeply thank Ramarko Bhattacharya, Harrison Dong, and Raymond Qian for their technical assistance working with the toio hardware. Further, we thank lab members Tucker Rae-Grant, Aditya Retnanto, and David Yuan for their help with proofreading. Lastly, we want to thank everyone else who helped with proofreading, including Matthew's mother and father.

REFERENCES

[1] Tyler Angert, Miroslav Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. 2023. Spellburst: A node-based interface for exploratory creative coding with natural language prompts. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–22.

[2] Matthew Blackshaw, Anthony DeVincenzi, David Lakatos, Daniel Leithinger, and Hiroshi Ishii. 2011. Recompose: direct and gestural interaction with an actuated surface. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. 1237–1242.

[3] Richard A Bolt. 1980. "Put-that-there" Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 262–270.

[4] John Joon Young Chung and Eytan Adar. 2023. Promptpaint: Steering text-to-image generation through paint medium-like interactions. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–17.

[5] John Joon Young Chung, Wooseok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: visual sketching of story generation with pretrained language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–4.

[6] Gabriela Csurka, Riccardo Volpi, and Boris Chidlovskii. 2023. Semantic Image Segmentation: Two Decades of Research. arXiv:2302.06378 [cs.CV] https://arxiv.org/abs/2302.06378

[7] Sida Dai, Brygg Ullmer, and Winifred Elysse Newman. 2024. MorphMatrix: A Toolkit Facilitating Shape-Changing Interface Design. In *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 1–12.

[8] Valdemar Danry, Pat Pataranutaporn, Yaoli Mao, and Pattie Maes. 2023. Don't Just Tell Me, Ask Me: AI Systems that Intelligently Frame Explanations as Questions Improve Human Logical Discernment Accuracy over Causal AI explanations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 352, 13 pages. https://doi.org/10.1145/3544548.3580672

[9] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–22.

[10] Tejaswi Digumarti, Javier Alonso-Mora, Roland Siegwart, and Paul Beardsley. 2016. Pixelbots 2014. In *ACM SIGGRAPH 2016 Art Gallery*. 366–367.

[11] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).

[12] Severin Engert, Konstantin Klamka, Andreas Peetz, and Raimund Dachselt. 2022. STRAIDE: a research platform for shape-changing spatial displays based on actuated strings. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[13] Aluna Everitt, Faisal Taher, and Jason Alexander. 2016. ShapeCanvas: an exploration of shape-changing content generation by members of the public. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2778–2782.

[14] Hao Fei, Meng Luo, Jundong Xu, Shengqiong Wu, Wei Ji, Mong-Li Lee, and Wynne Hsu. 2024. Fine-grained Structural Hallucination Detection for Unified Visual Comprehension and Generation in Multimodal LLM. In *Proceedings of the 1st ACM Multimedia Workshop on Multi-Modal Misinformation Governance in the Era of Foundation Models* (Melbourne VIC, Australia) *(MIS '24)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/3689090.3689388

[15] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. 2013. inFORM: dynamic physical affordances and constraints through shape and

object actuation.. In *Uist*, Vol. 13. 2501–988.

[16] Phil Frei, Victor Su, Bakhtiar Mikhak, and Hiroshi Ishii. 2000. Curlybot: designing a new class of computational toys. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 129–136.

[17] Seth Copen Goldstein, Jason D Campbell, and Todd C Mowry. 2005. Programmable matter. *Computer* 38, 6 (2005), 99–101.

[18] Jianzhe Gu, Yuyu Lin, Qiang Cui, Xiaoqian Li, Jiaji Li, Lingyun Sun, Cheng Yao, Fangtian Ying, Guanyun Wang, and Lining Yao. 2022. PneuMesh: Pneumatic-driven Truss-based Shape Changing System. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 260, 12 pages. https://doi.org/10.1145/3491102.3502099

[19] Yijie Guo, Zhenhan Huang, Ruhan Wang, Zhihao Yao, Tianyu Yu, Zhiling Xu, Xinyu Zhao, Xueqing Li, and Haipeng Mi. 2024. AI-Gadget Kit: Integrating Swarm User Interfaces with LLM-driven Agents for Rich Tabletop Game Applications. *arXiv preprint arXiv:2407.17086* (2024).

[20] John Hardy, Christian Weichel, Faisal Taher, John Vidler, and Jason Alexander. 2015. Shapeclip: towards rapid prototyping with shape-changing displays for designers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 19–28.

[21] Fengming He, Xiyun Hu, Xun Qian, Zhengzhe Zhu, and Karthik Ramani. 2024. AdapTUI: Adaptation of Geometric-Feature-Based Tangible User Interfaces in Augmented Reality. *Proc. ACM Hum.-Comput. Interact.* 8, ISS, Article 527 (Oct. 2024), 26 pages. https://doi.org/10.1145/3698127

[22] Meng-Ju Hsieh, Rong-Hao Liang, Da-Yuan Huang, Jheng-You Ke, and Bing-Yu Chen. 2018. RFIBricks: Interactive building blocks based on RFID. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–10.

[23] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. 2012. Radical atoms: beyond tangible bits, toward transformable materials. *interactions* 19, 1 (2012), 38–51.

[24] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 234–241.

[25] Seungwoo Je, Hyunseung Lim, Kongpyung Moon, Shan-Yuan Teng, Jas Brooks, Pedro Lopes, and Andrea Bianchi. 2021. Elevate: A walkable pin-array for large shape-changing terrains. In *Proceedings of the 2021 CHI Conference on human Factors in Computing Systems*. 1–11.

[26] Aoran Jiao, Tanmay P. Patel, Sanjmi Khurana, Anna-Mariya Korol, Lukas Brunke, Vivek K. Adajania, Utku Culha, Siqi Zhou, and Angela P. Schoellig. 2023. Swarm-GPT: Combining Large Language Models with Safe Motion Planning for Robot Choreography Design. arXiv:2312.01059 [cs.RO] https://arxiv.org/abs/2312.01059

[27] Hiroki Kaimoto, Kyzyl Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched reality: Sketching bi-directional interactions between virtual and physical worlds with ar and actuated tangible ui. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–12.

[28] Lawrence H Kim, Daniel S Drew, Veronika Domova, and Sean Follmer. 2020. User-defined swarm robot control. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[29] Lawrence H Kim and Sean Follmer. 2019. Swarmhaptics: Haptic display with swarm robots. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–13.

[30] Gierad P Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. Pixeltone: A multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2185–2194.

[31] Mathieu Le Goc, Lawrence H Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. 2016. Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th annual symposium on user interface software and technology*. 97–109.

[32] Joanne Leong, Florian Perteneder, Hans-Christian Jetter, and Michael Haller. 2017. What a life! Building a framework for constructive assemblies. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. 57–66.

[33] Jiaoyang Li, Wheeler Ruml, and Sven Koenig. 2020. EECBS: A Bounded-Suboptimal Search for Multi-Agent Path Finding. *CoRR* abs/2010.01367 (2020). arXiv:2010.01367 https://arxiv.org/abs/2010.01367

[34] Jiatong Li, Ryo Suzuki, and Ken Nakagaki. 2023. Physica: Interactive Tangible Physics Simulation based on Tabletop Mobile Robots Towards Explorable Physics Education. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (Pittsburgh, PA, USA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA, 1485–1499. https://doi.org/10.1145/3563657.3596037

[35] Rong-Hao Liang, Liwei Chan, Hung-Yu Tseng, Han-Chih Kuo, Da-Yuan Huang, De-Nian Yang, and Bing-Yu Chen. 2014. GaussBricks: magnetic building blocks for constructive tangible interactions on portable displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3153–3162.

[36] Artem Lykov, Sausar Karaf, Mikhail Martynov, Valerii Serpiva, Aleksey Fedoseev, Mikhail Konenkov, and Dzmitry Tsetserukou. 2024. FlockGPT: Guiding UAV Flocking with Linguistic Orchestration. *arXiv preprint arXiv:2405.05872* (2024).

[37] Damien Masson, Young-Ho Kim, and Fanny Chevalier. 2024. Textoshop: Interactions Inspired by Drawing Software to Facilitate Text Editing. *arXiv preprint arXiv:2409.17088* (2024).

[38] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2024. Direct-gpt: A direct manipulation interface to interact with large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–16.

[39] Ken Nakagaki, Sean Follmer, Artem Dementyev, Joseph A Paradiso, and Hiroshi Ishii. 2017. Designing line-based shape-changing interfaces. *IEEE Pervasive Computing* 16, 4 (2017), 36–46.

[40] Ken Nakagaki, Sean Follmer, and Hiroshi Ishii. 2015. Lineform: Actuated curve interfaces for display, interaction, and constraint. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 333–339.

[41] Ken Nakagaki, Joanne Leong, Jordan L Tappa, João Wilbert, and Hiroshi Ishii. 2020. Hermits: Dynamically reconfiguring the interactivity of self-propelled tuis with mechanical shell add-ons. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 882–896.

[42] Ken Nakagaki, Jordan L Tappa, Yi Zheng, Jack Forman, Joanne Leong, Sven Koenig, and Hiroshi Ishii. 2022. (Dis)Appearables: A Concept and Method for Actuated Tangible UIs to Appear and Disappear based on Stages. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3491102.3501906

[43] Ken Nakagaki, Udayan Umapathi, Daniel Leithinger, and Hiroshi Ishii. 2017. AnimaStage: Hands-on Animated Craft on Pin-Based Shape Displays. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (Edinburgh, United Kingdom) *(DIS '17)*. Association for Computing Machinery, New York, NY, USA, 1093–1097. https://doi.org/10.1145/3064663.3064670

[44] Takeru Oba, Matthew R. Walter, and Norimichi Ukita. 2024. READ: Retrieval-Enhanced Asymmetric Diffusion for Motion Planning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.

[45] OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/ Accessed: 2024-09-12.

[46] Jifei Ou, Mélina Skouras, Nikolaos Vlavianos, Felix Heibeck, Chin-Yi Cheng, Jannik Peters, and Hiroshi Ishii. 2016. aeroMorph - Heat-sealing Inflatable Shape-change Materials for Interaction Design. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 121–132. https://doi.org/10.1145/2984511.2984520

[47] Ivan Poupyrev, Tatsushi Nashida, and Makoto Okabe. 2007. Actuation and tangible user interfaces: the Vaucanson duck, robots, and shape displays. In *Proceedings of the 1st international conference on Tangible and embedded interaction*. 205–212.

[48] Wanli Qian, Chenfeng Gao, Anup Sathya, Ryo Suzuki, and Ken Nakagaki. 2024. SHAPE-IT: Exploring Text-to-Shape-Display for Generative Shape-Changing Behaviors with LLMs. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–29.

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[50] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. arXiv:2212.04356 [eess.AS] https://arxiv.org/abs/2212.04356

[51] Hayes Solos Raffle, Amanda J Parkes, and Hiroshi Ishii. 2004. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 647–654.

[52] Majken K Rasmussen, Esben W Pedersen, Marianne G Petersen, and Kasper Hornbæk. 2012. Shape-changing interfaces: a review of the design space and open research questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 735–744.

[53] John W Romanishin, Kyle Gilpin, and Daniela Rus. 2013. M-blocks: Momentum-driven, magnetic modular robots. In *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 4288–4295.

[54] Karl Toby Rosenberg, Rubaiat Habib Kazi, Li-Yi Wei, Haijun Xia, and Ken Perlin. 2024. DrawTalking: Towards Building Interactive Worlds by Sketching and Speaking. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 113, 8 pages. https://doi.org/10.1145/3613905.3651089

[55] Anne Roudaut, Abhijit Karnik, Markus Löchtefeld, and Sriram Subramanian. 2013. Morphees: toward high "shape resolution" in self-actuated flexible mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 593–602. https://doi.org/10.1145/2470654.2470738

[56] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. 2012. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*. 3293–3298. https://doi.org/10.1109/ICRA.2012.6224638

[57] Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. 2024. PMG : Personalized Multimodal Generation with Large Language Models. arXiv:2404.08677 [cs.IR] https://arxiv.org/abs/2404.08677
[58] Ben Shneiderman. 2022. Human-centered AI: ensuring human control while increasing automation. In *Proceedings of the 5th Workshop on Human Factors in Hypertext* (Barcelona, Spain) *(HUMAN '22)*. Association for Computing Machinery, New York, NY, USA, Article 1, 2 pages. https://doi.org/10.1145/3538882.3542790
[59] Alexa F Siu, Eric J Gonzalez, Shenli Yuan, Jason B Ginsberg, and Sean Follmer. 2018. Shapeshift: 2D spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.
[60] Volker Strobel, Marco Dorigo, and Mario Fritz. 2024. LLM2Swarm: Robot Swarms that Responsively Reason, Plan, and Collaborate through LLMs. arXiv:2410.11387 [cs.RO] https://arxiv.org/abs/2410.11387
[61] Yuta Sugiura, Calista Lee, Masayasu Ogata, Anusha Withana, Yasutoshi Makino, Daisuke Sakamoto, Masahiko Inami, and Takeo Igarashi. 2012. PINOKY: a ring that animates your plush toys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 725–734.
[62] Ryo Suzuki, Adnan Karim, Tian Xia, Hooman Hedayati, and Nicolai Marquardt. 2022. Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–33.
[63] Ryo Suzuki, Eyal Ofek, Mike Sinclair, Daniel Leithinger, and Mar Gonzalez-Franco. 2021. Hapticbots: Distributed encountered-type haptics for vr with multiple shape-changing mobile robots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 1269–1281.
[64] Ryo Suzuki, Clement Zheng, Yasuaki Kakehi, Tom Yeh, Ellen Yi-Luen Do, Mark D Gross, and Daniel Leithinger. 2019. Shapebots: Shape-changing swarm robots. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 493–505.
[65] Tiffany Tseng, Ruijia Cheng, and Jeffrey Nichols. 2024. Keyframer: Empowering Animation Design using Large Language Models. *arXiv preprint arXiv:2402.06071* (2024).
[66] Konstantinos Tsiakas and Dave Murray-Rust. 2024. Unpacking Human-AI interactions: From Interaction Primitives to a Design Space. *ACM Trans. Interact. Intell. Syst.* 14, 3, Article 18 (Aug. 2024), 51 pages. https://doi.org/10.1145/3664522
[67] Chao Wang, Stephan Hasler, Daniel Tanneberg, Felix Ocker, Frank Joublin, Antonello Ceravola, Joerg Deigmoeller, and Michael Gienger. 2024. LaMI: Large Language Models for Multi-Modal Human-Robot Interaction. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 1–10. https://doi.org/10.1145/3613905.3651029
[68] Keru Wang, Zhu Wang, Ken Nakagaki, and Ken Perlin. 2024. "Push-That-There": Tabletop Multi-robot Object Manipulation via Multimodal'Object-level Instruction'. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference*. 2497–2513.
[69] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–22.
[70] Cheng Xue, Yijie Guo, Ziyi Wang, Mona Shimizu, Jihong Jeung, and Haipeng Mi. 2024. DishAgent: Enhancing Dining Experiences through LLM-Based Smart Dishes. In *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–4.
[71] Lining Yao, Ryuma Niiyama, Jifei Ou, Sean Follmer, Clark Della Silva, and Hiroshi Ishii. 2013. PneUI: pneumatically actuated soft composite materials for shape changing interfaces. In *Proceedings of the 26th annual ACM symposium on User interface software and Technology*. 13–22.
[72] Lining Yao, Jifei Ou, Chin-Yi Cheng, Helene Steiner, Wen Wang, Guanyun Wang, and Hiroshi Ishii. 2015. BioLogic: natto cells as nanoactuators for shape changing interfaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1–10.
[73] Kentaro Yasu. 2022. MagneShape: A Non-electrical Pin-Based Shape-Changing Display. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–12.
[74] Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S Chirikjian. 2007. Modular self-reconfigurable robot systems [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14, 1 (2007), 43–52.
[75] Lilith Yu, Chenfeng Gao, David Wu, and Ken Nakagaki. 2023. AeroRigUI: Actuated TUIs for Spatial Interaction using Rigging Swarm Robots on Ceilings in Everyday Space. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.
[76] Yiwei Zhao, Lawrence H Kim, Ye Wang, Mathieu Le Goc, and Sean Follmer. 2017. Robotic assembly of haptic proxy objects for tangible interaction and virtual reality. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. 82–91.
[77] Yue Zhu, Zhiyuan Zhou, Jinlin Miao, Haipeng Mi, and Yijie Guo. 2024. TangibleNegotiation: Probing Design Opportunities for Integration of Generative AI and Swarm Robotics for Imagination Cultivation in Child Art Education. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 66–70.

A MODEL COMPARISON

Success Rate. We evaluated the success rate of each LLM model for each LLM agent. Figure 11 presents the success rates for each model and task, measured by the percentage of successes in 20 trials. Averaging across all tasks, we found that GPT-4o slightly leads with a mean success rate of 97% with a standard deviation (SD) of 4.5%, followed by Llama 3.3-70b (95%, SD: 5%) and Claude 3.7 Sonnet (94%, SD: 6.5%).

Load Time. Figure 12 shows the mean and standard deviation for the load time across 20 test cases, for each LLM agent and LLM model. GPT-4o demonstrated the fastest mean load time across all tasks, at 1.14s. Llama-3.1-70b and Claude-3.5-Sonnet trailed further behind with means of 2.32 and 2.34 seconds. [1] The Animation Agent

[1] Note that API request load times vary based on factors like the overall server traffic and proximity to servers, leading to inconsistency in API response time https://platform.

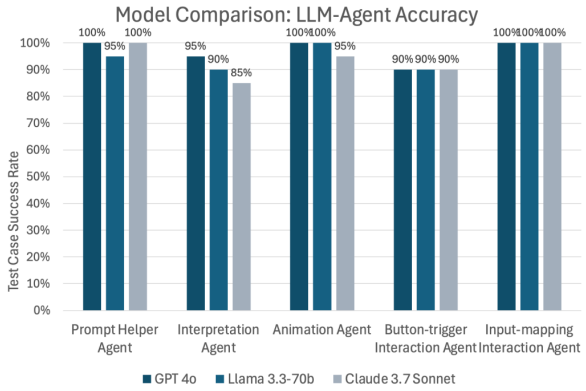

Figure 11: Model comparison by LLM agent test case success rate. For each bar, the success rate is the percentage of passed test cases out of 20 trials.

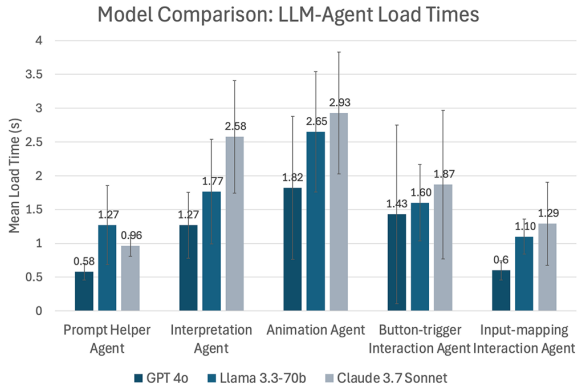

Figure 12: Model comparison by LLM agent mean load time. Each bar shows the mean load time across 20 trials, including the standard deviation.

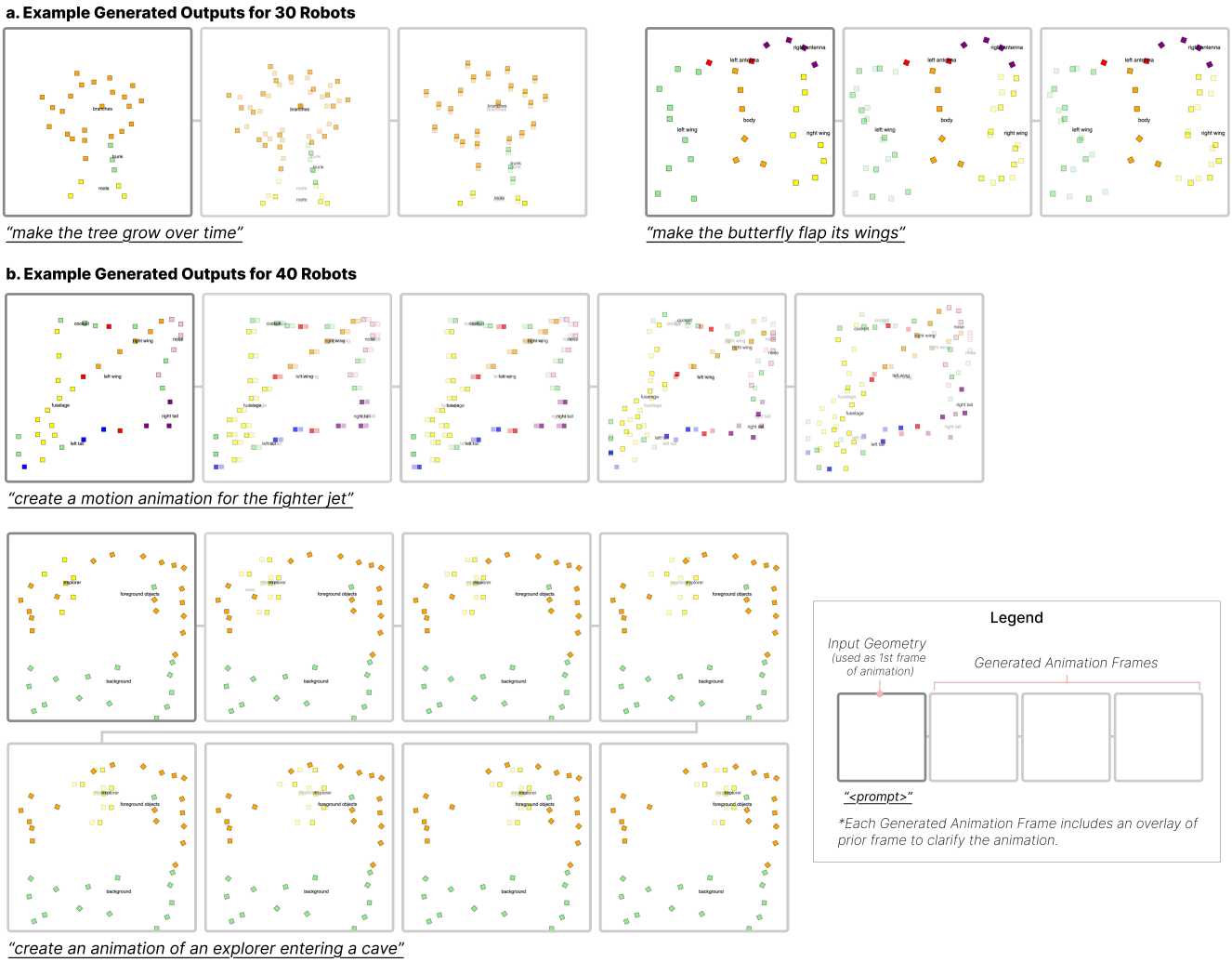

Figure 13: Example animation outputs from the scalability technical evaluation, displayed frame by frame. The top row shows a tree and butterfly animation, each with 30 robots. The middle and bottom rows show a fighter jet and cave explorer animation, each with 40 robots.

took the longest out of the five LLM agents, as it is instructed to generate complex movement scripts. In contrast, the Prompt Helper and Input-Mapping Interaction Agents were the fastest, as their prompts instruct them to output short instruction strings.

B SCALABILITY EXAMPLES

We conducted our technical evaluation with 10, 20, 30, and 40 robots to develop a better understanding of system performance when scaled beyond the number of robots in our user study, which was limited to 11 robots. While we noticed degradation in the success rate (Figure 6), the system succeeded in outputting many animations, which become more detailed as the robot count increases.

openai.com/docs/guides/rate-limits. Our evaluation provides a general estimate for load times and is still a valuable comparison tool between LLM agents and models.

We display several example animations with 30 and 40 robots in Figure 13.