

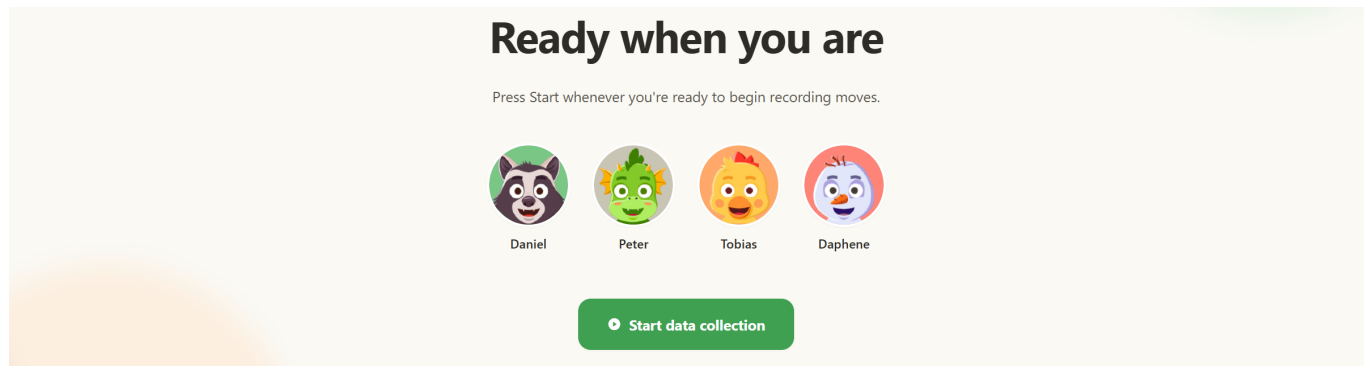
MoveML: A Classroom Platform for Embodied Machine Learning Education

DANIEL ANKER POULSEN, Aarhus University, Denmark

MINDAN CHEN, Aarhus University, Denmark

PETER MÆRSK CLAUSEN, Aarhus University, Denmark

TOBIAS BAK, Aarhus University, Denmark



Abstract

Children today interact with AI-powered systems daily, yet most receive little formal education in how these systems work or why they behave as they do. Existing tools for introducing machine learning to younger learners tend to offer single-user, surface level experiences. That allow students to train and test a model without developing an understanding of the role their data plays in shaping its behavior. This paper presents MoveML, a collaborative, embodied machine-learning platform for micro:bit classroom environments, designed to introduce children aged 8 to 12 to three core ML concepts: variation, consistency, and balance in training data. Using accelerometer-equipped micro:bit wristbands, students contribute movement data to a shared, centrally hosted model and observe in real time how their collective data shapes what the model can and cannot recognize. An evaluation with 14 sixth-grade students that children were able to notice and articulate the consequences of imbalanced and inconsistent training data without direct instruction, and that the embodied, collaborative format drove sustained engagement. We discuss the role of the facilitator, and directions for future development.

ACM Reference Format:

Daniel Anker Poulsen, Mindan Chen, Peter Mærsk Clausen, and Tobias Bak. 2026. MoveML: A Classroom Platform for Embodied Machine Learning

Authors' Contact Information: Daniel Anker Poulsen, au724950@uni.au.dk, Aarhus University, Aarhus, Denmark; Mindan Chen, Aarhus University, Aarhus, Denmark, 202510726@post.au.dk; Peter Mærsk Clausen, Aarhus University, Aarhus, Denmark, 202207956@post.au.dk; Tobias Bak, Aarhus University, Aarhus, Denmark, au720022@uni.au.dk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM XXXX-XXXX/2026/6-ART

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

Education. 1, 1 (June 2026), 21 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 Introduction

Children today are surrounded by AI. They interact with recommendation systems, voice assistants, and content filters from an early age, often before they have any formal understanding of how these systems work or what drives their behaviour. As Druga et al. note, by the time AI is introduced in schools, many children have already formed intuitions about these technologies without the critical tools to question them [9]. This gap between everyday use and conceptual understanding is increasingly recognised as a challenge for K-12 education, yet the tools available to address it are limited in important ways.

Platforms such as Teachable Machine [11], LearningML [13], and CreateAI for micro:bit [18] have made machine learning accessible to beginners, lowering the barrier to training and testing a model considerably. However, these tools tend to centre around a single-user experience in which one student collects their own data, trains a model, and briefly tests it. This design, as Tseng et al. observe in their work on Co-ML [23], limits the opportunity to encounter the kinds of data challenges that matter most: what happens when data is sparse, inconsistent, or unevenly distributed across classes. When a single user collects all the data themselves, in the same way, these issues rarely surface in a meaningful way.

We argue that a collaborative, embodied approach can address this limitation. When an entire class contributes data to a shared model, the consequences of variation, consistency, and balance become visible and discussable in a way that a solo experience cannot easily produce. A student who trained the model on three examples of

their own wave gesture will see it fail when a classmate attempts the same movement. A class that collectively trains on the same gesture will see recognition improve immediately. A dataset skewed heavily toward one class will cause the model to predict that class for almost everything, something that students can observe, identify, and correct themselves.

This paper presents MoveML, a platform designed around these principles. Built on top of the open-source ml-trainer project [17] and extended with a custom server layer, role-based WebSocket communication, and teacher, student interfaces, MoveML enables a class of students to collaboratively train a shared gesture recognition model using micro:bit wristbands. The activity is structured around three phases, each designed to surface one of the core ML concepts through direct embodied experience rather than instruction. A student-facing shared screen makes the state of the model visible to the whole class at all times, supporting collective observation and facilitating reflection between phases.

We evaluated the system with 14 students aged approximately 12 across three classroom sessions, combining semi-structured interviews, observation and user experience questionnaires. The evaluation found that children were able to notice and express the effects of imbalanced and inconsistent training data without these concepts being named in advance, and that the physical, movement-based format drove strong engagement.

2 Related Work

Recent developments in AI education highlight a growing tension between increasing accessibility to machine learning tools and the depth of understanding they provide. While AI and ML are becoming more common in K-12 classrooms, scholars such as Casal-Otero et al. and Sanusi et al. [6, 21, 24] identify an ongoing challenge in supporting children to move beyond surface level interaction toward a deeper understanding of how these systems are constructed, trained and shaped by data.

2.1 Embodied Learning

Research in embodied AI learning shows that physical interaction can help make abstract AI concepts more concrete and accessible. Long and Magerko [15] argue that when learners can move, act or physically engage with a system, ideas like training and prediction become easier to grasp. More recent work on embodied computing supports this view. Dybbøe et al.'s Bit:sort [10] demonstrates how abstract computer science concepts can be made tangible through physical group activities, where students represent data or algorithms with their own bodies, making invisible computational processes visible. By generating data through their own movements, which is immediately used to train a model, students engage with machine learning not only as observers, but as active participants in creating the data.

2.2 Collaborative Learning

Collaboration is also a well established factor in computing education. Preston's work on pair programming shows that structured collaboration, where students work in pairs with clear roles, can

improve learning outcomes, as students support each other, share ideas, and stay more focused [20].

Tseng et al.'s research on collaborative machine learning (Co-ML) [23] extends this to AI education specifically, showing that working together to build datasets improves understanding of how machine learning works. When people contribute data as a group, they can directly observe how differences in the dataset affect model performance, including problems such as class imbalance. The goal is for students to see how their collective data shapes model behaviour. This makes collaboration not just a social arrangement, but a core part of the learning mechanism.

2.3 Socially Meaningful Learning

Several research traditions emphasise that computing education is most effective when it is connected to social interaction and real world meaning. Bjørn and Petersen's FemTech research [2] argues that people are more likely to engage with computing when it is tied to social relationships and meaningful contexts, rather than treated as a purely technical activity. Work on computational crafts and e-textiles by Buechley and Eisenberg similarly shows that students are more engaged when they can design and build something that feels personal or creative. In these settings, children are often more motivated because they can express identity through what they make [5].

2.4 Earlier AI Education

Most existing research in AI and ML education targets older students, typically at secondary school level or above [6, 21]. This leaves a significant gap in our understanding of how younger children engage with and make sense of these technologies. Our project attempts to address this by targeting children aged 8-12, introducing AI and ML concepts earlier in their formal education.

Druga et al.'s research on family-based AI learning [9] shows that children are already active users of AI-powered technologies in their daily lives, interacting with recommendation systems, voice assistants, and personalised content platforms long before they receive any formal instruction in how these systems work. This means that by the time AI is introduced in school, many children have already formed intuitions about these systems, without the critical tools to question them.

By engaging children with the underlying mechanisms of machine learning and how data is collected, how models are trained and how predictions can fail. It gives the children a foundation that most current education delay until secondary school [6, 21].

2.5 AI Literacy and BlackBox

A central concern in current AI education research is that children are often able to train and use models without understanding why those models behave as they do. As a result, AI systems can appear as a "black box," where inputs and outputs are visible but the underlying role of data, bias and model training remains hidden [6, 24]. Casal-Otero et al. and Sanusi et al. [6, 21] argue that AI

literacy must go beyond technical skills to include an understanding on how data reflects human choices, how models generalise from limited examples and how algorithmic systems can reproduce bias and misclassification .

This limitation is also visible in existing tools. Platforms such as Teachable Machine [11], LearningML [13], CreateAI for micro:bit [18], and Machine Learning for Kids [12] make AI accessible to beginners, but tend to offer single use experiences, where students train a model and briefly test it without developing an understanding of why it behaves as it does.

Druga et al.'s research on family based AI learning offers a productive alternative, showing that when children and parents attempt to "break" or confuse AI systems, they often gain important insights into how these systems fail, including issues of bias and fragility [9].

2.6 Overall Contribution

Combined, the reviewed literature suggests that accessible AI education for children requires more than simplified tools for individual model training. It requires designs that make data variation, inconsistency and imbalance visible and discussable. MoveML addresses this gap by turning model training into a shared, embodied classroom activitysee Figure 1.



Fig. 1. Overview of related work, current gaps, and MoveML contribution

3 Research Question

"How can a collaborative, embodied machine-learning activity with micro:bit support 8-12-years old in noticing how variation, consistency, and balance in training data affect machine-learning model behavior?"

4 Prototype Concept

MoveML is designed for a single class session of approximately 50 minutes, structured so that children encounter three core ML concepts (variation, consistency, and balance in training data) through direct, physical experience rather than traditional teaching. These three concepts form the core of the activity and map directly onto the research question driving this project (See section: 3).

4.1 System Overview

The platform is built around Casal-Otero et al. and Sanusi et al. [6, 21] premise: Rather than being told how machine learning works, children become the dataset. They generate movement data with their own bodies, contribute to a shared model hosted centrally on the platform, and observe in real time how the data they produce shapes what the model can and cannot recognize.

All students are equipped with a micro:bit around their wrist and are placed in pairs. The onboard accelerometer captures movement data and streams it to the platform. On the student side, each student selects an avatar to identify themselves within the session. The teacher operates a separate screen that displays students' avatars, collected gestures, training progress and model predictions in real time, functioning as a shared surface for the whole class (Teacher pages can be found in Appendix C).

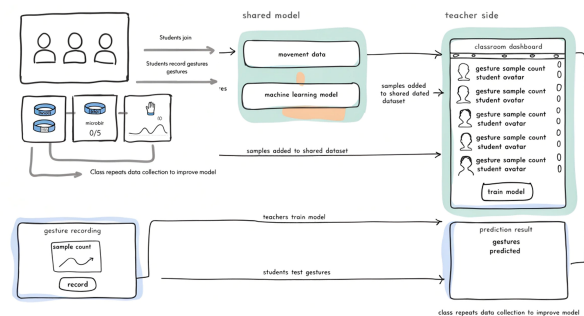


Fig. 2. Sketch showing the different parts of the prototype, and how they work together

The teacher's screen is the primary site of collective observation: it makes the model's behavior visible and discussable at all times, positioning the teacher as a facilitator guiding the class through what they themselves are producing.

See Figure 2 for an overview of the system.

4.2 Activity Structure

The session moves through three activity phases, each designed to surface one of the core ML concepts. The teacher controls the pacing and transitions between phases, and leads brief reflective discussions between them to help children move from concrete observation toward conceptual understanding.

4.3 The Teacher's Role

The teacher's role throughout the session is facilitative rather than instructional. They control when data collection starts and stops, when the model is trained, and which gestures students are asked to test. Between each phase, the teacher leads a short class reflection, posing open questions: *Why do you think the model got that wrong? What is different this time? What do you notice about the data?*

These pauses are important. The embodied activity produces visible model changes in a new way. The value depends on children being supported to articulate what they have noticed and connect it to the underlying concept. The teacher's questions scaffold this transition from experience to understanding (See Figure 3).

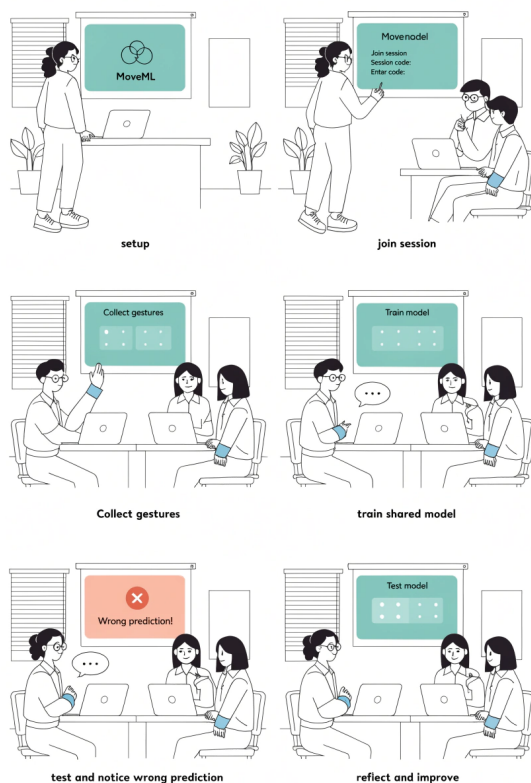


Fig. 3. Storyboard of the MoveML classroom activity based on the evaluation setup. Students work in pairs with laptops and micro:bit wristbands while a facilitator uses the student-facing projector screen to coordinate data collection, model training, testing, and reflection.

4.4 Use-Case Scenario

The following scenario illustrates how the session is intended to unfold, following a class of 24 ten year olds and their teacher, Ms. Andersen.

Setup. Ms. Andersen has arranged the classroom into twelve pairs. Each student gets a micro:bit wristband. She opens the teacher dashboard on the classroom projector and introduces the session: today, she tells them, they are going to teach a computer to recognize movement using their own bodies. All students pairs their micro:bit and selects an avatar on the student platform. Within a few minutes, all of the students appear on the shared screen.

Phase 1: Variation. Ms. Andersen starts data collection and instructs everyone to invent their own movement and record three samples of it. The pairs spend a few minutes experimenting. Anders settles on a arm wave. Brian chooses a wrist flick. Caroline does a small jumping motion. The movements appear on the shared screen as they are recorded, and named accordingly.

Ms. Andersen then stops collection and trains the model. She asks the class to perform Anders' wave. The model does not recognize it, when the student tries to replicate it. She pauses the class: *"Why do you think it's struggling?"* A student suggests that maybe they're all doing the wave slightly differently from how Anders does it. Another notes that some students are left-handed and their movements come from a different direction. Ms. Andersen asks the class to consider what that means for the model. The idea surfaces: the model only ever saw the wave performed by Anders, three times, in one particular way. It has never seen anyone else do it.

Phase 2: Consistency. Ms. Andersen starts collection again. This time, every person records ten samples of Anders' wave move. The shared screen shows the amount of recorded samples growing rapidly.

Ms. Andersen trains the model and asks students to test it. The wave is recognized reliably, by students from every pair. The class notices the difference immediately. *"What changed?"* she asks. *"Everyone did it,"* one student says. *"So the computer saw loads of different versions of it,"*. Ms. Andersen confirms this and names the concept: when more people contribute data for the same movement, the model learns to recognize a wider range of performances.

Phase 3: Balance. The wave now has far more samples than any of the gestures from Phase 1. She says nothing about why this might matter and instead asks the students to test Brian's wrist flick. The model predicts *wave*. The class reacts with surprise.

"Why did it say wave?" Ms. Andersen asks. After a brief discussion, a student points out: *"There is way more waves than anything else"* The class agrees: the model has seen the wave move so many times that it has started predicting it for almost everything. Ms. Andersen asks what they think should be done. The students add more samples to the underrepresented gestures, the distribution evens out, and the model is retrained. When the wrist flick is tested again, it is recognized correctly.

Closing Reflection. Ms. Andersen brings the class together. She asks what surprised them and what they now think they understand about how AI learns. The students talk about needing lots of examples, about the model making mistakes when data is uneven, and about how the same movement can look different depending on who performs it. Ms. Andersen closes by telling them that what

they did today, collecting data, training a model, and noticing how their choices shaped its behavior, is the same process behind the AI systems they encounter every day.

5 Design Process

Our design process evolved through several iterations, shaped by literature, feedback and ongoing reflection. And our research question evolved alongside it. The story of how MoveML came to life is not one of a clear vision from the start, but of following a path, questioning ideas along the way, and eventually arriving at something more interesting and novel than what we originally had in mind.

Throughout the process we found the value in externalizing ideas through discussions, sketches, observations and prototypes. Making ideas visible allowed them to be reflected upon, challenged and refined as a group. This aligns with Dix and Gongora's [8] description of externalization as a central activity in design.

5.1 Understanding the Domain

The process began with gaining an understanding of both the domain and the micro:bit ecosystem itself. Through meetings and presentations with Johnny from the micro:bit organization, the group was introduced to the values, vision and educational ambitions behind the platform. These conversations provided insight into how micro:bit aims to support playful learning, creativity and accessibility within technology education.

At the same time, the group explored the broader research field concerning children, technology education, and emerging technologies in schools. Research by Van Mechelen et al. [24] argues that emerging technologies in K-12 education should support creativity, collaboration, and embodied interaction rather than only focusing on technical skills. This became an important inspiration for the project direction.

To further understand the domain, our group conducted observations and interviews at Dokk1, where we attended a "digital play" workshop (See Figure 4). Here, we observed children and their parents interacting with different technologies, especially technologies designed to teach children about IT, computational thinking, and digital creativity. Observing how the children engaged with these technologies gave us valuable insights into what captures their attention and what makes learning feel playful and engaging. During the workshop, we talked with "Ole Caprani", professor at Aarhus University and an experienced educator in the field of digital tools for children. He provided us with many ideas, reflections and perspectives that helped shape our own concepts and discussions throughout the project. Most importantly for our project, he mentioned the importance of personalization and engagement when working with children.

We also spoke with "Michael Hjardgaard Andresen", an IT teacher at an elementary school, who gave us feedback on our initial ideas



Fig. 4. Observations and discussion during the "digital play" with Ole Caprani

and shared insights into how technology is currently taught in elementary schools. He emphasized the importance of an "engage first" approach, where the main focus should be on capturing the children's interest before introducing technical concepts. He highlighted the value of creating concepts that feel personal and relatable to the children. Furthermore, he shared reflections on the current situation regarding boys and girls in IT education, including why more boys than girls tend to participate in these activities, as well

as some of the challenges teachers experience when teaching technology in classrooms.

These activities helped ground the project in authentic educational settings and revealed both opportunities and challenges when designing technology for children.

5.2 Initial Ideation and Concept Development

Following the domain research, we began brainstorming possible concepts. A central ambition from the start was to create a product capable of engaging girls in IT education through the use of micro:bit and Jaccad technologies.

Several initial concepts were generated and later presented to the micro:bit organization for feedback. From these, one concept stood out as both promising and technically feasible: DanceCode. The original idea behind DanceCode was to combine coding and dance, allowing children to program dance routines through movement and interaction (See Figure 5).

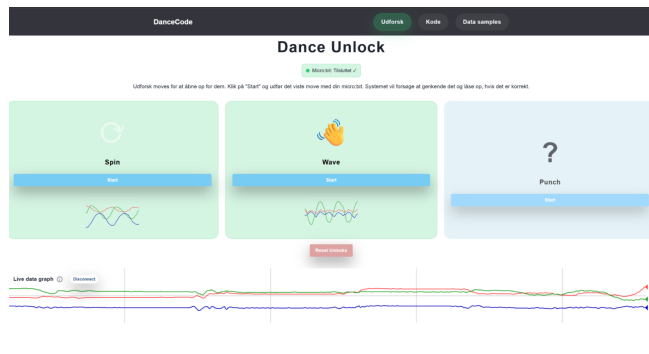


Fig. 5. DanceCode interface

We then entered a long iterative phase working on DanceCode. During this period, brainstorming sessions and internal discussions were continuously used to evaluate the concept and align the group's understanding. We also engaged with other stakeholders throughout this process, including teachers and the micro:bit foundation, which shaped many of our design decisions. This reflects aspects of a user-centered and collaborative design approach as described by Beyer [1], where iterative reflection and continuous alignment between stakeholders are essential parts of agile development.

5.3 Killing Our Darling

As development progressed, we found ourselves repeatedly caught between three competing constraints: our research question focused on engaging girls in IT, the technical limitations of the micro:bit and a concept that was increasingly resembling existing product like CreateAI [18] rather than developing into something novel. Every time we tried to move forward, we had to compromise one of the three in order to proceed. This created a cycle that was difficult to break out of.

Ideas that were too ambitious for the hardware had to be scaled back. Features that felt creative and engaging were removed to stay within the scope of the research question. And the closer the concept slowly moved toward existing systems, the harder it became to justify what we were actually adding.

Motivation within the group was also lacking due to this. We did not feel strongly about the direction we were heading, which made it harder to push through the compromises. Through discussions with teachers, supervisors, and within the group itself, it became clear that we were forcing a concept onto a platform it was not well suited for, and that continuing down this path would not lead anywhere meaningful. We decided it was better to stop and rethink the direction entirely.

This realization became a critical turning point in the design process. The project no longer felt meaningful or motivating. We therefore decided to “kill our darling” and restart the conceptual process despite having already invested significant time into the existing solution. This shift became an important learning experience.

It highlighted the importance of remaining reflective throughout the process and ensuring that ideas were continuously questioned rather than treated as fixed solutions. Drawing on Dix and Gongora's notion that external representations serve as tools for collective reflection and critical discussion [8], we adopted a more iterative and exploratory approach where sketches, concepts and discussions were revisited repeatedly instead of only in the early phases of the project. This approach reduced the chances of us spending too much time on a wrong direction again.

5.4 Reframing the Project

Although the first iteration was abandoned, it still provided valuable insights and research foundations. Through this earlier work, we had gathered substantial knowledge about what makes computing education engaging, particularly for girls. Much of the literature suggested that approaches emphasising collaboration, creativity, personalisation and physical engagement can positively influence participation and motivation, as described in Section 2.6.

These insights directly informed our new direction. Rather than focusing solely on engaging girls in IT, we reframed the project toward creating a system that could engage children more broadly with AI and ML concepts through embodied and collaborative interaction. This shift was also informed by the view, expressed by our teacher collaborator "Michael Hjaardgaard Andresen", that a design succeeding in engaging girls would naturally appeal to boys and most of the other underrepresented groups in IT as well.

The earlier research therefore remained central to the new direction rather than being discarded. The focus on engagement, creativity and collaboration carried forward, whilst shifting the focus to teaching children about ML.

This reframing also helped us clarify the design commitments that would guide the next iteration. First, we wanted the learning experience to begin from action, so that students could encounter machine-learning behaviour through movement before being introduced to abstract concepts. Second, we wanted collaboration to be more than a social feature. We chose to make the classroom itself become the unit of learning, with students contributing to a shared dataset and observing how collective data affects model behavior. Third, we wanted the system to make failure visible and useful. Misrecognition was treated as a learning opportunity through which students could notice the effects of variation, inconsistency, and imbalance in the training data. These commitments shaped the transition from DanceCode to MoveML and informed the later decision to structure the activity around a shared model, a student-facing display, and three phases of embodied data collection and testing.

5.5 MoveML

Following the change in direction, we conducted brainstorming sessions and held discussions with teachers and teaching assistants, whose feedback had also been part of the reason for the shift in the first place. Through these conversations, we developed a clear sense of what we want to build: a more classroom focused system.

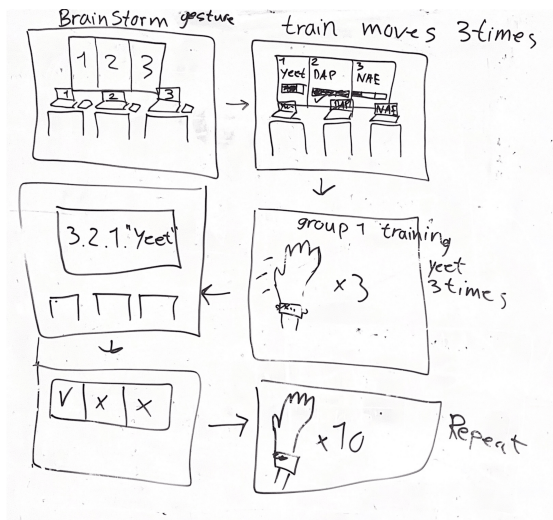


Fig. 6. Rough initial sketch showing the flow of the system, as well as the teacher and student screens.

Because our initial concept was closely related to existing tools like CreateAI [18], we wanted to develop something novel rather than simply replicating what already existed. This led us to MoveML, a system that builds on the values of micro:bit-based AI training while introducing a key distinction: instead of students training individual models, the whole class contributes to one shared model together (See Figure 6 for a sketch of the idea). Collaboration was therefore not just a feature but a core principle of the system. This was an important design shift.

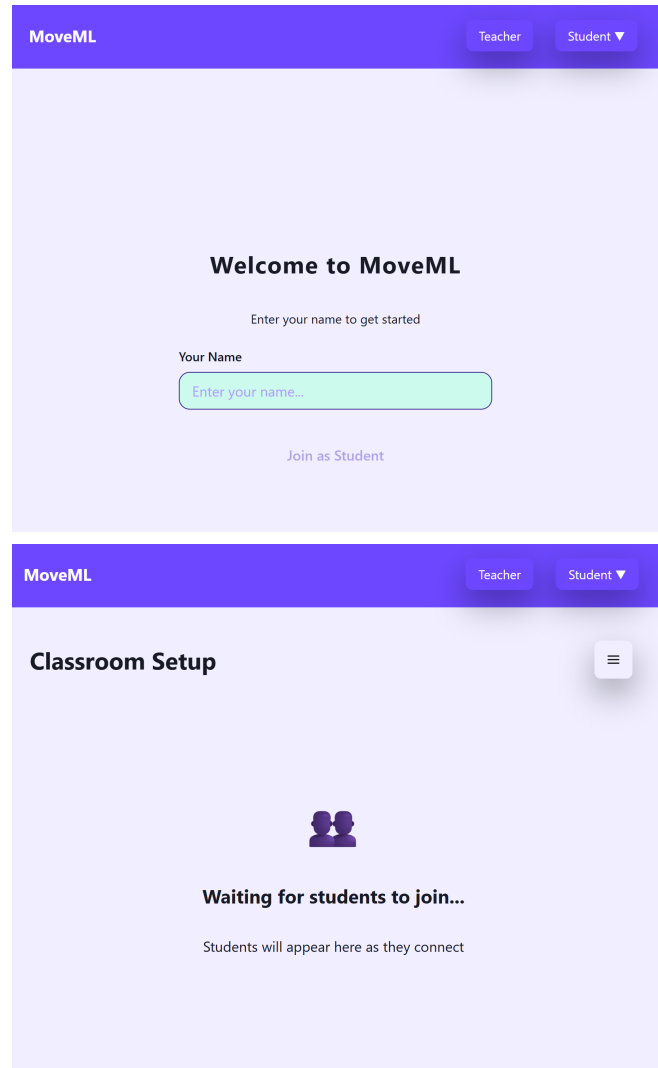


Fig. 7. One of the first prototypes of MoveML, showing the starting screen and the teacher/whiteboard screen.

In DanceCode, movement was treated mainly as an engaging input method. In MoveML, movement became the material through which students produced training data, compared different performances and observed how the model responded to the collective dataset.

We already had a system hooked up to a website, so we had the whole technical concept of connecting multiple micro:bits to a shared system. In addition, we now had a shared vision of what we wanted to make, so it was just about tweaking our new research question, and then developing the concept.

With the concept in place, the focus shifted toward testing and refinement. We tried the new system in different settings, stress-tested its functionality and worked through iterations and the bugs and flaws they introduced (See Figure 7 for early iteration of MoveML).

Feedback from teachers and internal testing helped us improve the layout and usability over time. Through this process we gradually arrived at a prototype that was clean, accessible and easy to understand.

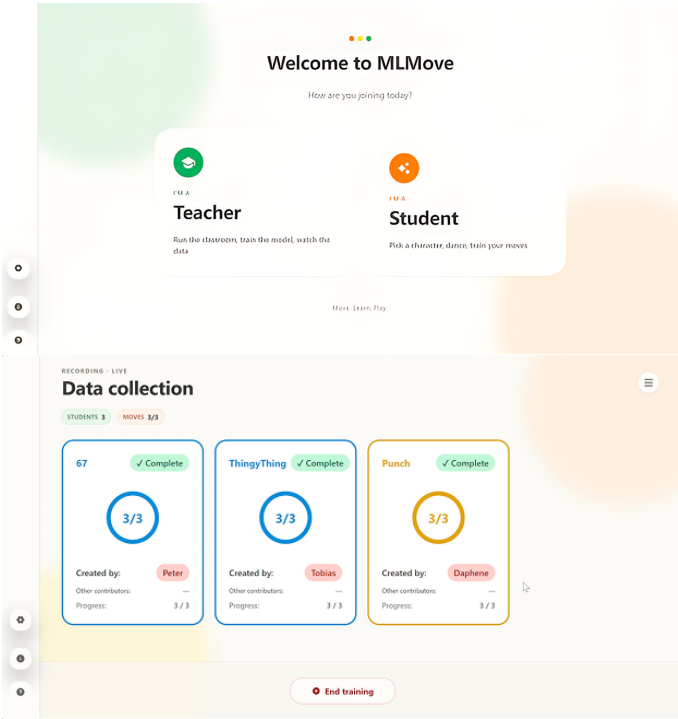


Fig. 8. One of the latest prototypes of MoveML, showing the welcome screen and teacher data collection screen

MoveML was shaped by everything we had gathered along the way, including the related work on embodied and collaborative learning, feedback from teachers and teaching assistants, and the lessons learned from earlier iterations. Rather than starting from scratch, we combined these insights into a single system that was both educationally grounded and genuinely novel.

The many directions we explored and eventually abandoned were not wasted effort. Each dead end taught us something, and it was through that process of failing and redirecting that we gradually found both the project and the approach that made it work. Final interface for the MoveML prototype can be seen in Figure 8.

5.6 Wristband Refinement

Although MoveML was mainly developed as a web-based classroom platform, the evaluation made it clear that the physical wristband also mattered. The first wristband prototype exposed the micro:bit and used a simple strap. This worked well enough for collecting movement data, but it still looked more like a testing device than something designed for children to wear. During the evaluation, some students commented on the appearance of the wristbands. At the same time, students spent a noticeable amount of time choosing

avatars, which showed that personalisation was important to them.

This led us to reconsider the wristband as part of the overall experience. The aim was not only to make it look nicer, but to think about how the wearable could better support the activity. It still needed to function as a motion-sensing device, but it also had to feel more robust, more comfortable, and more appropriate for a classroom with young learners. We explored different strap structures, ways of placing the micro:bit, visibility of the display, access to buttons, and existing micro:bit wearable cases. These explorations helped us define what a more integrated version of the wristband might need.

The resulting concept was called Animal Buddies (See Figure 9). Instead of treating the wristband as a neutral technical strap, this concept turns it into a set of animal-based wearable characters. Each version gives the device a different identity, while still keeping the micro:bit visible as the interactive part of the product. This connects the physical wristband with the avatar idea in the interface. Rather than having the avatar only appear during setup, the character could become part of the whole experience, from wearing the device to contributing movement data to the shared model.



Fig. 9. ML Wristband.

6 Technical Implementation

6.1 Project Overview

The application is an interactive machine-learning platform for micro:bit classroom environments, enabling students to collectively collect accelerometer data and collaboratively train a shared gesture recognition model in real-time. The system is derived from the open-source `ml-trainer` [17] (the same as `CreateAI` [18]) project, which provided a single-user, client-side foundation for gesture data collection and model training. Our contribution transforms this into a multi-user classroom tool by introducing a server layer, a role-based communication architecture and a set of teacher and student interfaces built on top of the existing frontend stack.

Concretely, the original `ml-trainer` had no server component. All data collection and model training happened locally in a single browser. We added a Node.js server, moved the ML model to run server-side so that all students contribute to one shared dataset, and built the teacher and student pages that controls the classroom session. The technology choices for the frontend (React 18 with TypeScript, Chakra UI, and Vite) were kept consistent with the existing codebase rather than introduced from scratch.

6.2 Architecture and Real-Time Communication

The system follows a client-server architecture where a teacher client and a set of student clients all connect to a central Node.js server over WebSockets. The most significant architectural decision was how to implement real-time synchronization between clients.

We considered a database approach in which clients would write state to a cloud database and poll for changes via API calls. For a classroom of roughly 20 simultaneous users this would likely have been functional, but it would have required a constant polling loop on each client to detect state changes, introducing unnecessary latency and network traffic. WebSockets, by contrast, keep a persistent bidirectional connection open, so the server can push updates to all clients immediately when something changes. This fit our use case well: when a teacher starts data collection or triggers model training, all student interfaces need to react within moments. We therefore chose `Socket.IO` [22], which provides a `WebSocket` abstraction.

The server maintains a single session state, shown below, which is broadcast to all connected clients whenever it changes:

```

1 {
2   isCollecting: boolean,    // Training data
                             // collection active
3   dataset: Array,          // All collected
                             // training samples
4   model: MLModel | null,   // Trained
                             // gesture model
5   trainingInProgress: boolean, // Model
                             // currently fitting
6   currentStudentPage: string, // Synced
                             // navigation state
7   gestures: Array,         // Available
                             // gesture definitions

```

```

8   studentConnections: Map  // Active student
                             // connections
9 }

```

The communication between clients and server is divided into four sequential phases, illustrated in Figures 10–13.

The full set of `Socket.IO` events is summarised in Table 1.

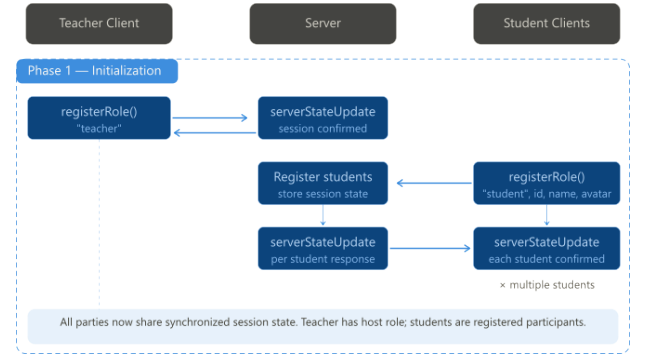


Fig. 10. Phase 1: Initialization. The teacher registers with the server and receives session confirmation. Students then register with their ID, name, and avatar, and all parties arrive at a shared synchronized session state.

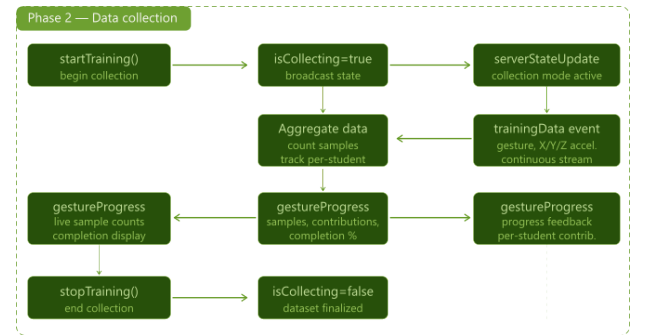


Fig. 11. Phase 2: Data collection. The teacher triggers collection via `startTraining()`. Students stream accelerometer samples to the server, which aggregates them and broadcasts progress back to all clients.

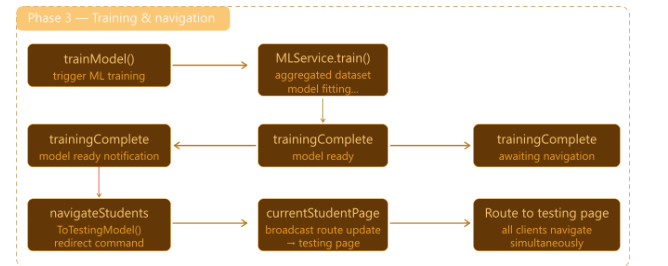


Fig. 12. Phase 3: Training and navigation. The teacher triggers `MLService.train()` on the aggregated dataset. Once complete, a `trainingComplete` event is broadcast and the teacher navigates all student clients simultaneously to the testing page.

Event	Direction	Purpose
registerRole	Student → Server	Initial connection identification
trainingData	Student → Server	Submit accelerometer + gesture label
testData	Student → Server	Submit data for model inference
createGesture	Student → Server	Define new gesture class
startTraining	Teacher → Server	Enable data collection
trainModel	Teacher → Server	Trigger ML model training
navigateStudentsTo*	Teacher → Server	Broadcast page navigation command
serverStateUpdate	Server → All	Broadcast updated session state
gestureCreated	Server → All	Notify all clients of new gesture

Table 1. Socket.IO events and their directions.

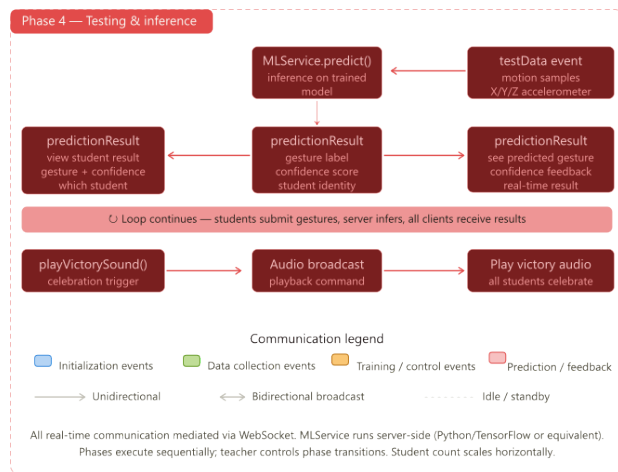


Fig. 13. Phase 4: Testing and inference. Students submit motion samples as `testData` events. The server runs inference via `MLService.predict()` and broadcasts `predictionResult` events with gesture label and confidence score to all clients.

6.3 Server-Side ML and Role Separation

A core design decision was moving the `MLService` from the client to the server. In the original `ml-trainer`, the model was trained and ran entirely in the user's browser, meaning each user had their own isolated model. To support a collaborative classroom scenario, where every student's gesture samples contribute to a single shared model, the service needed to live centrally. We therefore ported `MLService` to the server unchanged in terms of its internal logic, but now it operates on the collected dataset from all students rather than data from a single user.

Role separation between teacher and student is a concept we introduced entirely. The teacher acts as session host: they control when collection starts and stops, when the model is trained, and when students are navigated to the next phase. Students are participants who contribute data and receive feedback. This is implemented through the `SocketHandler` class on the server, which routes incoming events differently depending on the role declared at registration time.

6.4 Student and Teacher Interfaces

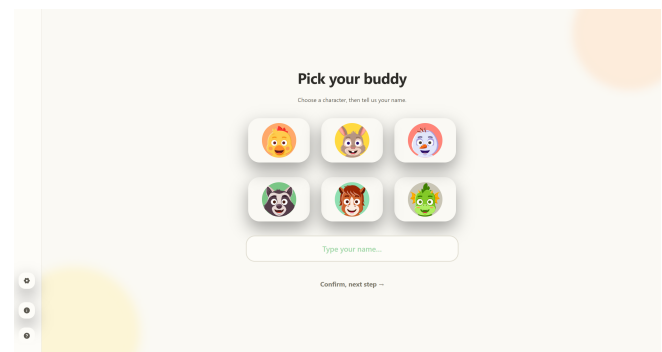


Fig. 14. Student setup screen



Fig. 15. Student testing screen

The student-facing interface consists of five React components. On arrival, students complete a short setup step in `StudentSetupPage` where they enter their name and select an avatar (See Figure 14). The main interaction happens in `StudentTraining` (Can be seen in Figure 8), which displays a live graph of accelerometer readings from the connected micro:bit and allows students to record gesture samples individually or in batches of ten. During the testing phase, students submit gestures and receive immediate visual feedback showing the predicted label and confidence score as can be seen in Figure 15.

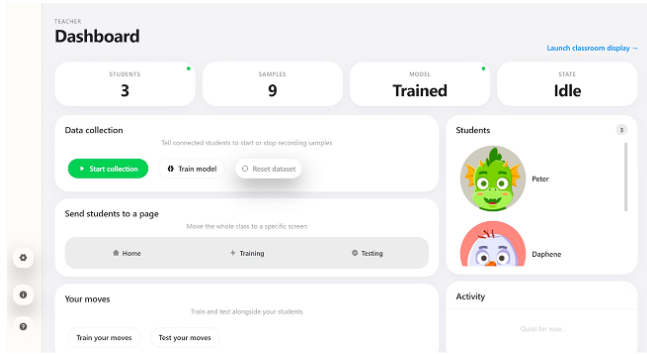


Fig. 16. Teacher dashboard screen

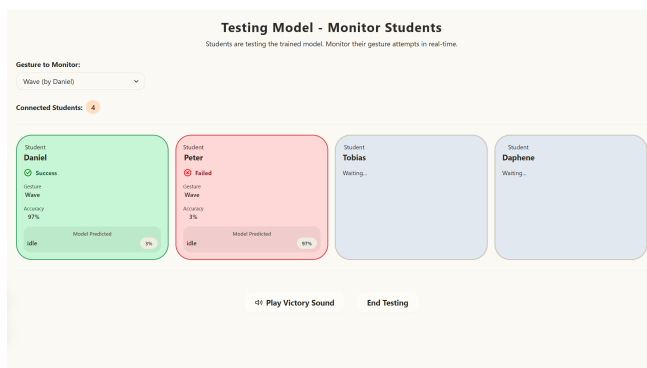


Fig. 17. Teacher testing screen

The teacher interface consists of three components controlling the session from a single screen. `TeacherClassroomScreen` provides a multi-step workflow that moves the session from joining through data collection to training and testing, displaying live progress such as sample counts per gesture and the number of contributing students (The teacher pages can be found in Appendix C). `TeacherDashboard` (See Figure 16) gives an overview of session state, and `TeacherTestingPage` (See Figure 17) allows the teacher to monitor prediction results and trigger a victory sound broadcast when a gesture is correctly recognised.

6.5 Deployment

The system is containerized using Docker Compose, with separate containers for the server and client. This decision was due to the circumstances: the evaluation required the application to run on several machines simultaneously, and Docker meant that nothing needed to be installed on those machines. We also considered deploying the application to a hosted cloud service, but given that it only needed to function on the evaluation day within a local network, that would have been unnecessary. Instead, the teacher's machine runs both containers, and student devices connect over the local network via HTTPS.

The server container exposes port 5000 for WebSocket connections, and the client container serves the built React application over port 443. The server Dockerfile is shown below as an example of the build setup:

```
1 FROM node:18-alpine
2 WORKDIR /app/server
3 COPY server/package*.json ./
4 RUN npm install
5 COPY server/ ./
6 EXPOSE 5000
7 CMD ["npm", "start"]
```

6.6 Summary of Contributions

All server-side code is new work. The frontend builds on the React and Chakra UI foundation of `ml-trainer` but all teacher and student pages were written from scratch. The `MLService` was ported from client to server without modification to its core logic. In summary:

- **Student pages:** 5 custom React components for student onboarding, data collection, and testing
- **Teacher pages:** 3 custom React components for session orchestration and result monitoring
- **WebSocket server:** approximately 300 lines of event handling, session state management, and role-based routing
- **Server-side ML:** `MLService` migrated from client-side to a central server, enabling collaborative model training
- **Docker infrastructure:** multi-stage builds, HTTPS support, and environment-based configuration for local network deployment
- **Custom UI:** custom UI components build using existing the existing UI framework Chakra UI

7 Evaluation

Before the formal evaluation, we tested the system informally with classmates and among ourselves. The aim was to catch obvious bugs, confirm the system ran end to end, and get a rough sense of its stability and engagement. The formal evaluation came next: held in a classroom with students on a high-fidelity prototype.

The formal evaluation combined quantitative and qualitative data collection. Quantitative data were gathered using the User Experience Questionnaire (UEQ), with the results presented in Section 7.7. Qualitative data were collected through semi structured interviews and observations conducted in a school setting. The observational approach was informed by Blomberg's ethnographic framework [3], which emphasizes understanding participants within their own context and from their own perspective, while the interviews followed the guidelines proposed by Leech [14]. Combining these methods allowed us to develop a more holistic understanding of how the system was used in its real-world context. The gathered data was analysed using thematic analysis following Braun and Clarke's methodology [4] to identify recurring patterns and themes.

7.1 Evaluation Goals

Three goals shaped the evaluation. First, we wanted to see whether students could notice and put into words how the training data

shaped the model's predictions, particularly its quantity, variation, and inconsistency. Second, we wanted to understand how the embodied, collaborative format affected engagement and participation. Third, we looked for usability and facilitation problems that would matter for classroom deployment: setup, timing, feedback, and what the facilitator has to manage.

7.2 Participants and Setting

We ran three sessions, one with six students and two with four students each, for 14 participants in total. Every participant was a sixth-grader, and they worked in pairs. Each pair shared a computer, with the system open across two browser tabs in a split-screen layout. That gave both students the activity interface and the relevant system view at once, though only one of them could control the computer at a time.

Sessions took place in a classroom at Aarhus University. A facilitator operated the teacher dashboard, projected at the front of the room, so the class could follow the activity as it unfolded: who was participating, what data was coming in, how the model trained and what it predicted.

7.3 Procedure

Each session ran the same way and lasted about 50 minutes. We opened with a 10 minute warm up session on what students already knew about AI and ML. Had they heard the terms before, and where? Did they use AI at school? Could AI get things wrong, and what did they think it was good or bad at?

The main phase was three rounds with the system, roughly ten minutes each. In each round, students recorded movement samples, trained the shared model, tested its predictions, and discussed how it behaved, following the activity flow described in Chapter 4.4.

A 10 minute reflection closed each session. We asked what had surprised them, what they had learned, what they still wanted to know, and whether their view of AI or ML had shifted. They then completed the UEQ as a descriptive and honest record of their experience with the prototype.

7.4 Data Collection and Analysis

While students worked, the team watched how they used the prototype, how they divided the work in pairs, where they got stuck, and how they accounted for the model's behaviour during reflection. We paid particular attention to the moments when a student noticed an error, argued about why a prediction had failed, or changed how they collected data in response to what the model had just done.

The warm-up and closing discussions let us compare what students understood before and after the activity. We listened for how they described AI and ML, whether they accepted that such systems can be wrong and whether they could explain how a model learns from data. A set of prepared questions kept the same themes in view across sessions, and follow-up questions let facilitators pursue

explanations as they came up.

Observation notes and discussion notes then went through thematic analysis, looking for recurring patterns in learning, engagement, collaboration, and interface breakdowns. Three questions guided us: did students connect the model's performance to the data they had produced, did the physical interaction help them take part, and what broke when the system met something close to a real classroom?

The UEQ responses were used as a descriptive supplement to the qualitative findings. The questionnaire was included to capture students' subjective experience of the prototype across dimensions such as attractiveness, efficiency, perspicuity, dependability, stimulation, and novelty.

7.5 Results

7.5.1 Prior Knowledge and Conceptual Understanding. At the start of the sessions, AI was already familiar to most participants. Students referred to popular media, consumer products, schoolwork, and the fact that their school supports the use of "skoleGPT" [7]. This familiarity helped underline the relevance of the project: children were already encountering AI in everyday and educational contexts, but did not necessarily have a clear understanding of how such systems learn or why they make mistakes.

Students grasped the training process early. They understood that they had to train the model themselves and that more samples generally meant better performance. One student drew on films to explain ML. This suggests that popular culture can give children a useful starting point for understanding unfamiliar concepts, even if it is not always accurate.

By the end, the concept students stated most confidently was the link between how much data the model had and how well it performed. They did not get there from a lecture. They watched the model fail, added or compared samples, and saw the predictions change. This is ML literacy of a practical kind: students learned about training data by producing it, testing it, and watching the model react.

Once every group had trained the same shared movement with ten samples each, that movement had around 40 samples behind it while the earlier individual gestures had only three. The model then predicted the dominant class regardless of what movement someone actually performed. Students noticed this themselves and explained it: the model had leaned toward whatever it had seen most. The activity made class imbalance visible to the students.

Some failures they explained through variation in the data itself. When a gesture went unrecognized, a student would point out that one person was left-handed, or that someone had performed it "his own way." They were no longer scoring the model as simply right or wrong. They were tracing errors back to how the data had been produced, which is exactly what MoveML is meant to support, since

it treats training data as something the class makes with their own bodies.

7.5.2 Engagement and Learning Behaviour. Engagement held up across all three sessions, and the movement was a large part of why. Students did not just sit by their laptops. Several stood up, used larger body movements, and treated the activity as play. The teachers in the room noticed it too and remarked that the students seemed to be enjoying themselves.

The more outgoing students tended to run the discussions and steer their pairs, while quieter ones sometimes slipped into a watching role. Even then they stayed within the activity. They could observe the model, add to the shared dataset, and discuss within the pair without participating in class reflections.

7.5.3 Usability Issues. Observation turned up several usability problems. Many students needed help getting set up and connected, which tells us the onboarding is not yet simple enough for this age group to manage on their own. The activity worked well with two evaluators present, but it could be challenging for one teacher to manage an entire class alone. Future versions need a more simple setup.

Timing also confused some students. They often could not tell when to perform a gesture or when the system was actually recording. Someone suggested a countdown sound during the sessions.

7.5.4 Design Feedback and Suggestions. Several concrete design suggestions emerged from participant feedback and observation. Multiple avatar options were requested by several participants, suggesting that greater personalisation could increase engagement and a sense of ownership.

The wristbands used during the evaluation also received negative feedback from students. Several participants found them difficult to put on independently, and while working in pairs helped somewhat, the setup process still caused unnecessary friction. Some students also commented that the wristbands looked unappealing. This was not entirely surprising, as improving the physical wristband design was something the group had discussed early on but ultimately set aside in favour of other priorities. The evaluation confirmed that it remains a worthwhile direction, and further design explorations are presented in Section 5.6.

One participant also suggested that the wristband should deliver an electric shock when an incorrect prediction was made. This is not a literal design recommendation, but it reflects a genuine desire for stronger, more tangible feedback from the system, and should be taken as a signal that the current feedback mechanisms felt insufficient.

7.6 Limitations

The total sample of 14 participants across three sessions is small, which limits the generalisability of the quantitative results. The questionnaire received notable criticism from participants, suggesting it was not sufficiently tailored to the target age group. This

should be taken into account when interpreting the data, and the question framing should be revised for future evaluations.

Finally, our framing as facilitators was found to have a significant effect on session dynamics. The first session in particular was shaped by a competitive framing introduced by the evaluators. This is both a limitation of the evaluation and a finding in its own right: it underscores how sensitive the activity is to facilitation, and how important it will be to provide clear guidance to teachers in future deployments.

7.7 Questionnaire Results

The UEQ results provide a descriptive indication of how students experienced the prototype. Overall, responses suggest that the system was perceived as friendly, easy to learn, and easy to use (See Figure 18).

The system received lower ratings on novelty and leading edge qualities than expected. This was somewhat surprising, as we consider both the system and the topics it addresses to be relatively novel. While the activity was perceived as enjoyable and easy to understand, students did not necessarily view the interface itself as particularly innovative. This distinction is important, as the novelty of MoveML may lie less in its visual interface and more in its educational design, where students collectively generate training data and observe how a shared machine learning model responds.

These aspects may not have been fully reflected in the students' responses. Furthermore, terms such as *konservativ* (conservative) and *innovativ* (innovative), which were used in the questionnaire, may be difficult for younger students to interpret.

8 Future Work

While the evaluation of MoveML was broadly positive, aspects of the design were adjusted and reconsidered following the study, as also mentioned in the previous section of the design process 5. Beyond these refinements, some areas remain open for future iterations of the platform.

Personalisation and Avatar Integration. The current implementation offers limited personalisation: students select from eight avatars during setup, but these avatars play no further role in the experience. A natural extension would be to use avatars more meaningfully throughout the session, for instance by displaying them on the teacher's shared screen as prediction results come in, so the class can see who produced which outcome. We have also explored the idea of physically printed wristbands matching each pair's avatar as described further in Section 5.6, giving students a tangible connection between their identity in the system and the device on their wrist.

Integration with micro:bit Classroom. At the request of micro:bit, a worthwhile direction would be to integrate MoveML into the existing micro:bit Classroom platform [16]. This would allow teachers to run MoveML alongside other micro:bit assignments within a single

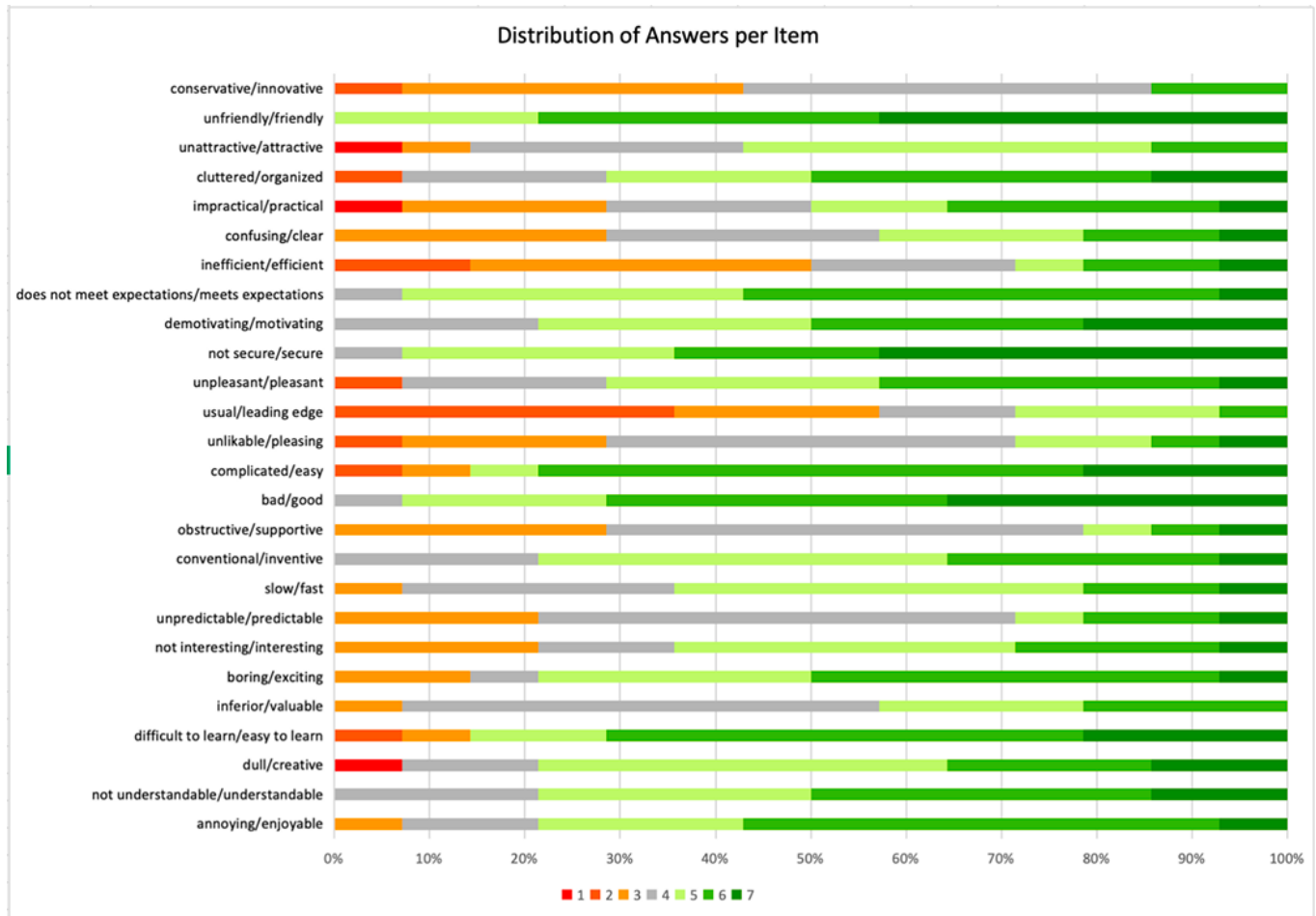


Fig. 18. User Experience Questionnaire Results

session.

Such an integration would also require aligning the visual design of the interface with the existing micro:bit style, making it easier to switch between tools and ensuring a more coherent experience within the broader micro:bit ecosystem. This would allow students to work within a platform they are already familiar with, not only when learning about ML but also across other subjects. In this way, progress and personalization could carry over between activities, enabling students to seamlessly continue where they previously left off.

Block Programming and Computational Thinking. Early in the project we explored a block-programming extension that would let students combine trained gestures into simple programs. We set this aside as it felt outside the core scope, but it remains a natural next step, especially if MoveML is integrated into the micro:bit ecosystem. Tying gesture recognition to MakeCode style block programming [19] would connect MoveML to computational thinking in a way that feels familiar and engaging for this age group.

Dataset Inspection. A dedicated dataset inspection interface would allow both teacher and students to explore the data collected during the session in greater depth. This is directly inspired by the Training Data Dashboard in Co-ML [23], a collaborative tablet-based ML tool in which all training samples are displayed in a shared view grouped by label, with per-class sample counts visible. In Co-ML, this dashboard served as a focal point for collective discussion: participants used it to identify mislabelled samples, notice gaps in data representation and recognise class imbalance before it affected model performance. A similar interface in MoveML could give students a concrete view of the dataset they have collectively built, making it easier for the teacher to explain why the model behaves as it does. Co-ML also found that when students could see and edit the dataset, they would naturally start checking it themselves, spotting and removing bad samples without being asked. Whether this level of depth is appropriate for the 8 to 12 age group is an open question.

Multi limb Data Collection. Finally, the current platform captures accelerometer data from a single micro:bit worn on one wrist. A richer data collection setup capturing movement from multiple limbs

simultaneously would encourage more movement and thereby engagement. One approach would be to use Jacdac to connect several micro:bits worn on different limbs, though this introduces wiring complexity that could be disruptive in a classroom setting. This remains an open hardware challenge, but addressing it would improve the consistency and fairness of the training data across all students, as well as expand the range of possible moves. The current inconsistency and unfairness in the data is however something that helps get the ML concepts across to the students, so that is something to keep in mind.

9 Discussion

9.1 Focus on Students, Not Teachers

The design and evaluation of MoveML centered almost entirely on the student experience. This was a deliberate choice given the scope of the project, but it leaves an important gap. ML is unpredictable: a model can behave unexpectedly, classifications can fail for reasons that are not immediately obvious, and the relationship between data and model behaviour is not always easy to read in the moment. In a classroom deployment, the teacher is the person who needs to respond to these situations, field questions from students, and guide the class toward a meaningful understanding of what just happened.

This requires at least a foundational understanding of how machine learning works, which cannot be assumed of every teacher. Our evaluation was facilitated by the researchers who built the system, which meant that ML expertise was always present in the room. In a real deployment, this would not be the case. Supporting teachers, through documentation, a facilitation guide, or a brief onboarding experience, could be a next step for any future version of the platform.

9.2 Competition

The activity was designed as a collaborative experience, where all students worked toward a shared goal of training a model that recognises everyone's movements. However, the first evaluation session showed that the framing of the activity strongly influenced how students engaged with it. In that session, students were more competitive. This was not a property of the system itself but of how it was introduced. When the framing was adjusted in later sessions, less competitive behavior emerged.

This raises a design question: what would a deliberately competitive version of the activity look like, and what could it add? While it is unclear whether this would support deeper understanding or distract from it, the observed dynamics suggest that competition is a meaningful alternative framing worth exploring. Competition is not necessarily negative; it can also increase engagement and motivate students to persist and improve, which may in turn support learning.

9.3 Pairs or Individuals?

We initially envisioned that each student would work individually on their own computer. A strong requirement was that all students should wear a micro:bit. In the evaluation, every student wore a micro:bit, but because we did not have enough computers, they shared

them in pairs. This turned out to work well in practice, as students collaborated, helped each other in small teams, and then shared their results with the whole class. This aligns with Preston's work on pair programming [20], which shows that structured collaboration, where students work in pairs with clear roles can improve learning outcomes, as students support each other, share ideas, and remain more focused.

Though this pair approach worked well, using this exact method with multiple taps of course isn't a reliable or consistent solution. We need to change the system so it's built to be worked on in pairs in future iterations of our systems. We should also revisit the original approach of having each student work on an individual computer in practice, in order to evaluate its value and determine which of the two directions is the most viable going forward.

10 Conclusion

This project set out to answer the research question in section 3. To address this, we designed and built MoveML, a classroom platform in which groups of students contribute accelerometer data from micro:bit wristbands to a shared, centrally hosted model. Rather than instructing children about ML directly, the system positions them as active participants in the data collection and training process, making the consequences of their choices visible in real time through a student-facing shared screen. The platform was built around three core ML concepts, each through a distinct phase of the activity: variation in the first phase, where students discover that a model trained on few examples from one student fails to generalise to others; consistency in the second, where the whole class trains the same gesture together and recognition improves immediately; and balance in the third, where the uneven distribution of samples causes the model to systematically predict the dominant class.

The formal evaluation provided positive evidence that the activity can support the intended learning. Children developed a working understanding of the training process quickly, and the ML concepts emerged naturally rather than through instruction, with students noticing and articulating the problem themselves. The embodied, movement-based format drove strong engagement, with participants standing up, performing gestures with their whole bodies, and remaining involved throughout. The collaborative structure contributed to a sense of shared purpose.

At the same time, the evaluation revealed meaningful limitations. The onboarding process was not sufficiently intuitive for this age group without facilitator assistance. The questionnaire instruments were not well suited to the target age group and should be revised in future work. Furthermore, the facilitators framing turned out to be crucial for how the session turns out. If the facilitators had a competitive framing of the system, the sessions turned out more competitive. This further emphasizes the importance and power of the facilitator.

The decision to abandon the original DanceCode concept midway through the project, despite the investment already made, was difficult but ultimately necessary. The concept had drifted away from the research question and toward territory already occupied by existing tools. The experience reinforced the importance of maintaining critical distance from one's own ideas throughout a design process, and of treating prototypes and sketches as tools for questioning rather than for confirming.

In answering the research question, the project contributes a concrete demonstration that children in this age group can, through direct embodied experience, notice and articulate how variation, consistency, and balance in training data affect model behaviour, without these concepts being named or explained in advance. The activity structure, the shared teacher screen, and the role of facilitated reflection between phases all appear to be important contributors to this outcome. The results suggest that collaborative, embodied ML activities represent a promising direction for AI education at the elementary school level.

Division of labour

Although we all contributed to all aspects of the project, we all had our individual focuses, on which we spend more time. Daniel and Tobias were mostly involved in the technical implementations and the decisions regarding that. Peter focused on coordination, concept development, and user experience design. Mindan "Daphene" focused on the visuals of the project. This includes the frontend of the system, and wristband product design, storyboard sketch as well as various presentations throughout the project.

References

- [1] Hugh Beyer. 2010. *Structure of a User-Centered Agile Process*. Springer International Publishing, Cham, 37–68. doi:10.1007/978-3-031-02193-0_5
- [2] Pernille Bjørn and Maria Menendez-Blanco. 2019. FemTech: Broadening participation to digital technology development. In *Proceedings of the 27th ACM international conference on multimedia*. 510–511.
- [3] Jeanette Blomberg, Jean Giacomini, Andrea Mosher, and Pat Swenton-Wall. 2017. Ethnographic field methods and their relation to design. In *Participatory design*. CRC Press, 123–155.
- [4] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- [5] Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 423–432.
- [6] Lorena Casal-Otero, Alejandro Catala, Carmen Fernández-Morante, Maria Taboada, Beatriz Cebreiro, and Senén Barro. 2023. AI literacy in K-12: a systematic literature review. *International Journal of STEM Education* 10, 1 (2023), 29.
- [7] Center for Undervisningsmidler (CFU). [n. d.]. *SkoleGPT er en open source, gratis og GDPR sikker generativ AI, der kan bruges af skolens lærere i undervisningen*. <https://skolegpt.dk/>
- [8] Alan Dix and Layda Gongora. 2011. Externalisation and Design. In *Proceedings of the Second Conference on Creativity and Innovation in Design (DESIRE '11)*. Association for Computing Machinery, New York, NY, USA, 31–42. doi:10.1145/2079216.2079220
- [9] Stefania Druga, Fee Lia Christoph, and Amy J Ko. 2022. Family as a Third Space for AI Literacies: How do children and parents learn about AI together?. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–17.
- [10] Maja Dybbøe, Magnus Høholt Kaspersen, Johanne Birkkjær Bjerrum, and Marianne Graves Petersen. 2025. Bit: sort: Bringing Tangible Computing to Computer Science Unplugged to Support Children's Algorithmic Explorations. In *Proceedings of the 24th Interaction Design and Children*. 429–443.
- [11] Google. 2020. Teachable Machine. (2020). <https://teachablemachine.withgoogle.com/>
- [12] IBM. 2020. Machine Learning for Kids. (2020). <https://machinelearningforkids.co.uk/>
- [13] LearningML. 2021. LearningML. (2021). <https://learningml.org/en/home/>
- [14] Beth L Leech. 2002. Asking questions: Techniques for semistructured interviews. *PS: Political Science & Politics* 35, 4 (2002), 665–668.
- [15] Duri Long, Aadarsh Padiyath, Anthony Teachey, and Brian Magerko. 2021. The role of collaboration, creativity, and embodiment in AI learning experiences. In *Proceedings of the 13th Conference on Creativity and Cognition*. 1–10.
- [16] microbit-foundation. [n. d.]. *Run whole class sessions, easily share code with students and save progress*. <https://classroom.microbit.org/>
- [17] microbit-foundation. [n. d.]. *Train a machine learning model on movement data from the micro:bit's accelerometer*. <https://github.com/microbit-foundation/ml-trainer>
- [18] micro:bit Foundation. 2022. CreateAI for micro:bit. (2022). <https://createai.microbit.org/>
- [19] Microsoft microbit-foundation. [n. d.]. *A Blocks / JavaScript code editor for the micro:bit powered by Microsoft MakeCode*. <https://makecode.microbit.org/>
- [20] David Preston. 2005. Pair programming as a model of collaborative learning: A review of the research. *Journal of Computing Sciences in colleges* 20, 4 (2005), 39–45.
- [21] Ismaila Temitayo Sanusi, Solomon Sunday Oyelere, Henriikka Vartiainen, Jarkko Suhonen, and Markku Tukiainen. 2023. A systematic review of teaching and learning machine learning in K-12 education. *Education and Information Technologies* 28, 5 (2023), 5967–5997.
- [22] Socket.IO. [n. d.]. *Bidirectional and low-latency communication for every platform*. <https://socket.io/>
- [23] Tiffany Tseng, Jennifer King Chen, Mona Abdelrahman, Mary Beth Kery, Fred Hohman, Adriana Hilliard, and R Benjamin Shapiro. 2023. Collaborative machine learning model building with families using Co-ML. In *Proceedings of the 22nd annual ACM interaction design and children conference*. 40–51.
- [24] Maarten Van Mechelen, Rachel Charlotte Smith, Marie-Monique Schaper, Mariana Tamashiro, Karl-Emil Bilstrup, Mille Lundung, Marianne Graves Petersen, and Ole Sejer Iversen. 2023. Emerging technologies in K–12 education: A future HCI research agenda. *ACM Transactions on Computer-Human Interaction* 30, 3 (2023), 1–40.

A Use of GAI-tools

Declaration of use of GAI-tools

☒ **We have used generative artificial intelligence in the creation of this project.**

ChatGPT 5.5

Claude Sonnet 4.6

GitHub CoPilot – Claude Haiku 4.5

We have used generative AI tools in the following way

EXAMPLES OF USE
To get alternative phrasings – We copied some of our own text into it, and it rephrased it better.
Generate Code
Help understand codebase
Help generate figures
Used to bounce ideas off of

B Product Video

<https://youtu.be/JJ6YedQxD3Q>

C Overview of teacher classroom pages

