PlushPal: Storytelling with Interactive Plush Toys and Machine Learning

Tiffany Tseng
tiffany@akg.t.u-tokyo.ac.jp
University of Tokyo
Tokyo, Japan

Yumiko Murai
ymurai@sfu.ca
Simon Fraser University
Vancouver, British Columbia, Canada

Deanna Gelosi
gelosi@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Natalie Freed
natalie.freed@utexas.edu
University of Texas at Austin
Austin, Texas, USA

Tung D. Ta
tung@akg.t.u-tokyo.ac.jp
University of Tokyo
Tokyo, Japan

Yoshihiro Kawahara
kawahara@akg.t.u-tokyo.ac.jp
University of Tokyo
Tokyo, Japan

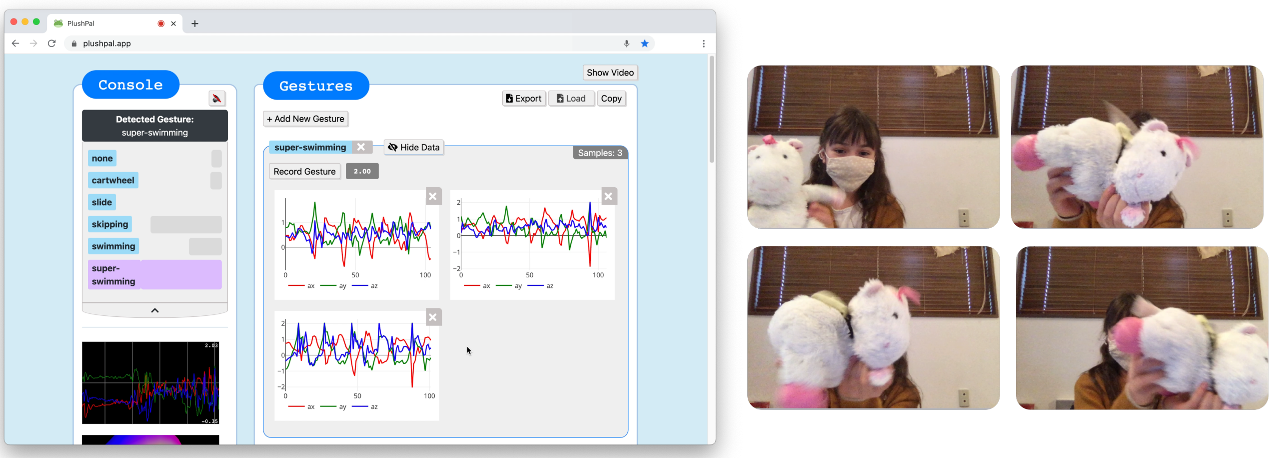

Figure 1: Girl creating *super swimming* gesture with her Unicorn stuffed animal and PlushPal

ABSTRACT

This paper presents PlushPal, a web-based design tool for children to make plush toys interactive with machine learning (ML). With PlushPal, children attach micro:bit hardware to stuffed animals, design custom gestures for their toy, and build gesture-recognition ML models to trigger their own sounds. We describe how, in the context of storytelling, PlushPal introduces core concepts in ML including data sampling and model evaluation. We conducted online workshops and in-person play sessions with 11 children between 8-14 years old building interactive stuffed animals with PlushPal. In these play sessions, we observed how children imagined bringing their toys to life using ML, as well as how children's data literacy changed as a result of experimenting with sensors, data sampling, and building their own ML models. Our work contributes a novel design space for children to express their ideas using gesture, as well as a description of observed debugging practices, building on efforts to support children using ML to enhance creative play.

IDC '21, June 24–30, 2021, Athens, Greece

ACM ISBN 978-1-4503-8452-0/21/06...$15.00
https://doi.org/10.1145/3459990.3460694

CCS CONCEPTS

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

KEYWORDS

machine learning, play, children, physical computing

ACM Reference Format:
Tiffany Tseng, Yumiko Murai, Deanna Gelosi, Natalie Freed, Tung D. Ta, and Yoshihiro Kawahara. 2021. PlushPal: Storytelling with Interactive Plush Toys and Machine Learning. In *Interaction Design and Children (IDC '21), June 24–30, 2021, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459990.3460694

1 INTRODUCTION

Physical computing platforms such as Arduino and micro:bit have created opportunities for beginners to experiment with sensors in building their own creative projects. In the process of programming hardware, makers begin to uncover how computers can understand their environment through a range of sensing including sound, movement, and light. This technical literacy is important not only for understanding how technologies work at a foundational level, but also to empower young people to gain agency in designing technologies themselves [5].

In introductory programming environments that support physical computing (such as MakeCode or Scratch), sensing capabilities can be limited in expressivity. Often, users may only pick from a fixed set of pre-designed defaults; for example, users can only pick from *move*, *shake*, or *jump* gestures in the Scratch micro:bit extension [11]. There are limited opportunities for children to incorporate higher-level sensory input such as speech or gesture, which has the potential to greatly expand the creative potential of making personally meaningful projects.

Speech and gesture recognition are often powered by machine learning (ML), which has generally been inaccessible to novices due to its complexity. Building ML models requires a different mental and design process than traditional programming — rather than supplying instructions via code, users provide examples through data by which a computer constructs a model to interpret and classify sensor input. Research on introducing ML concepts to young people is nascent but promising, with evidence that children as young as 8 years old can begin to understand the modeling process. [6, 13, 14, 35].

We are interested in what creative play opportunities are made possible when ML becomes accessible to novice creators. Additionally, as ML becomes increasingly ubiquitous in daily life, we recognize the importance of helping children understand how ML works and gain experience interacting with it so they can begin to recognize both its potential and limitations.

To explore how children might use ML to support creative play, we designed PlushPal, a web-based design tool for creating interactive stuffed animals using ML. With PlushPal, children attach the micro:bit, a popular microcomputer for beginners, to any stuffed animal and record examples of custom gestures; for example, children can record samples of their stuffed animal flying, running, or swimming and label them as inputs to an ML model. They can then program the computer to trigger sounds in response to gestural input (such as triggering snorkeling noises when a stuffed animal is "swimming.") We designed the interface to build off of children's storytelling and play practices with plush toys, providing a generative design space in which children can imagine their toy coming to life.

To study how children can build interactive toys with ML, we conducted play sessions with children where they designed their own gestures and sounds for a stuffed animal of their choice using PlushPal. We examined how they created their ML models, including how they reasoned about data sampling and debugging. Findings from this study contribute to emerging research on ML tools for children and creative play. Further, by uncovering misconceptions children had about ML, we also contribute to growing efforts around AI education for K-12 [32, 33]. The following research questions guided our study:

RQ1: How do children bring their stuffed animals to life using gestures and sound?

RQ2: How did children engage with data science practices when building their ML models with PlushPal?

In this paper, we first describe the PlushPal application for building ML gesture-recognition models that can trigger custom sounds. We then report findings from our accompanying studies with children designing projects using PlushPal. Finally, we present a series of recommendations to inform future work on interactive play with ML for children.

2 RELATED WORK

PlushPal is informed by existing ML tools for beginners, interactive plush toy toolkits, and research on supporting play using plush toys.

2.1 Machine Learning Tools for Beginners

Curricular standards for artificial intelligence (AI) in K-12 is an actively developing field [12, 32]. In the US, national guidelines for K-12 AI education are currently being developed; an initial set of ideas every student should know includes that "computers perceive the world using sensors" and "computers can learn from data" [32], with suggestions for children as young as kindergarten-aged (5-years-old). A proposed list of AI competencies include understanding how computers "recognize and make decisions" and the role humans play in programming and tuning AI systems [23].

Tools for beginners to design ML classifiers tend to focus on model building, requiring users to build applications *on top of* these models in a separate environment using text-based programming languages. Google's Teachable Machines [3] lets users construct ML models for sound, images, and video without having to write code; models are then exported to build custom web apps using tools like ml5.js [26]. Wekinator [8] is a desktop application for constructing ML models using gesture, audio, and computer vision that can drive custom applications built in other tools like Processing and Open Frameworks. The Example-Based Sensor Prediction (ESP) system [25] is a desktop application for novices to build ML models that handle live sensor input such as accelerometer data; the model can then be used in custom applications written in Arduino or Processing.

ML tools designed for kids is a relatively new domain, with several applications enabling screen-based interactive projects built upon the Scratch visual programming environment [27] including AI blocks for Snap [18], Cognimates [6, 7], ML for Kids [22], and AI with MIT App Inventor [16]. Apps supporting physical hardware include Scratch Nodes [1], which incorporates custom hardware for gesture-based physical play, and AlpacaML, with which children can build ML models for gesture-controlled games [34] or modeling sports moves [35]. We extend this early work by exploring how ML can be used with existing physical toys in a single integrated design tool, where children both construct their ML models and program their toys.

2.2 Interactive Plush Toys

Tools for designing interactive plush toys often involve building toys from scratch, which may require a significant investment of time. Plushbot [15] was a web-based interface for creating design patterns for soft toys that incorporate sewable electronics and conductive thread that makers assemble by hand. Skweezee [9] combined conductive wool, electrodes, and an Arduino for designing custom smart soft objects tested in occupational therapy settings. Cuddly [24] proposed embedding mobile phones into existing plush toys, using the mobile device both for sensing and to provide audio output. However, this method necessitates the destructive cutting of plush toys to embed hardware inside.

Non-destructive toolkits for turning existing plush toys interactive include Pinoky [31] and OnObject [4]. Pinoky is a ring-shaped brace that can be attached around the limbs of plush toys to animate back and forth movement with servo motors; this hardware is not broadly available and only accommodates stuffed animals with limbs of a certain size. OnObject used an RFID reader "ring" and attachable RFID tags to associate gestures like shaking and swinging with programmed behaviors, with a proposed application of supporting interactive toys. Both Pinoky and OnObject use custom hardware and do not allow for an arbitrary set of user-defined gestures. More recently, the use of detachable accelerometer sensors has been explored to support gesture recognition with plush toys [20], but this work also does not offer end users an interface for creating their own gestures.

In summary, the identified limitations of existing work around interactive plush toys is that they require significant effort to build a toy from scratch, require destructively taking apart existing toys, utilize specialized hardware, or are limited in the types of movement they can sense. We address these limitations by offering a tool in which children can attach off-the-shelf hardware (the popular and relatively inexpensive micro:bit device) to their existing plush toys and can design the toy to respond to custom movement of their own design.

2.3 Plush Toys and Play

PlushPal was designed to leverage existing play and storytelling practices with character toys. Pretend play can contribute to a child's cognitive and socioemotional development [2] and progresses from initial simple imitation of gestures in young children to collaborative role play and storytelling activities that can persist through middle childhood (around 11 years old) [10, 30]. Character toys (such as plush toys) are often used beyond the age that play-acting declines [17], perhaps because they provide a tangible anchor for a narrative. By enabling children to non-destructively add capabilities to plush toys they may be emotionally attached to, we hoped that PlushPal could connect to positive memories and prior storytelling activities with those toys. PlushPal also affords opportunities to engage in construction play [21] via the iterative design of a personalized interactive toy. For PlushPal, we chose to work with late-elementary and middle-school-aged youth as a balance between children young enough to enjoy pretend play and old enough to work with the micro:bit.

3 THE PLUSHPAL APPLICATION

PlushPal (https://www.plushpal.app) is a web application that lets users create ML gesture-recognition models using time-series accelerometer data from the micro:bit. It implements a supervised ML algorithm, requiring labelled training data to classify new, live sensor input. Each training example is a 2-second sequence of three-dimensional accelerometer data (x,y,z) from the micro:bit labelled with the name of the gesture. The system utilizes a 1-Nearest-Neighbor algorithm using Dynamic Time Warping (1NN-DTW) to calculate the non-Euclidean distance between live sensor data and each training sample – the label of the training example with the minimum distance is the output of the model, or its prediction of what gesture has been performed. We chose to use 1NN-DTW because it is a popular method that does not require a large quantity of training data and accounts for differences in speed between time series data [19, 29]. The overall interface for PlushPal can be seen in Figure 2, and the general workflow is as follows:

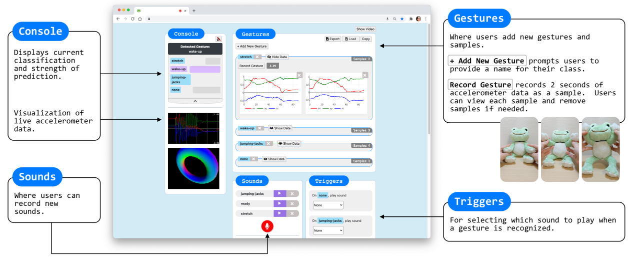

Figure 2: PlushPal Interface

- **Pair**: After downloading a custom hex program to their micro:bit for streaming sensor data over Bluetooth, users pair the micro:bit to the PlushPal app via Web Bluetooth.
- **Attach to Plush Toy**: Users securely attach their micro:bit to their plush toy using a DIY backpack (constructed out of paper or felt using a provided template) or directly with an elastic band.
- **Add Gesture Training Data**: The user adds a new gesture, gives it a name (its class), and records multiple 2 second samples of their stuffed animal performing the gesture.
- **Evaluate Model**: KNN does not require the model to be retrained with new data, allowing users to immediately test how well the model performs. The user can evaluate the model in PlushPal's console, which displays the current detected gesture, as well as the relative distances between the incoming data and each gesture class.
- **Add Sounds**: Users can record sounds using the microphone on their computer.
- **Connect gestures to sounds**: Each gesture can be linked with a sound (or play a random sound) in the Triggers section of the interface. The sound is played through the computer's speaker when the corresponding gesture is detected.

Throughout the process of using PlushPal, users can tweak their model by iteratively adding and removing gestures and samples based on how well it performs.

4 METHODOLOGY

To understand how youth could employ PlushPal to design their own interactive plush toys, we conducted both online workshops and individual in-person play sessions with children. Two different methods were carried out because of restrictions due to the

COVID-19 pandemic. In the United States, where children were not attending school in person, we conducted the PlushPal activity in facilitated online workshops. In Japan, where children were continuing to safely attend school in-person, we ran individual sessions with children at a local school. We obtained IRB approval for both types of sessions from our respective institutions.

For the online workshops, youth participants were recruited by reaching out to teachers and parents through social media, on a maker-educator forum, and through direct email. The remote workshops required that participants have a micro:bit they could use during the workshop. We ran a total of three online workshops, each 90 minutes long, over video conferencing. We iterated with using Jitsi and then Zoom for video conferencing; we had hoped that Jitsi's support for multi-user screensharing would help us better study children's design process, but we ended up switching to Zoom (which does not support multi-user screen sharing) for stability reasons.

After the online workshops, we ran individual in-person play sessions with children at a local school, with each session lasting 60 minutes. Participants were recruited via email through a network of English-speaking parents from international schools. Although the in-person sessions were shorter than the online workshops, the format of the activity stayed the same, and children had an equal amount of time to build their projects. The length of the in-person activity was able to be reduced by effectively eliminating the time that had been dedicated to group introductions and share-out and by bypassing technical instructions and troubleshooting, as a common micro:bit and laptop was used by all children. We ran individual sessions rather than group workshops both as a safety measure against COVID-19 as well as to prevent sound interference from multiple kids recording and playing back audio. The temperature of all youth and their parents were taken before they entered the school, and all materials were disinfected between sessions.

4.1 Workshop Structure

Each workshop began with the facilitator sharing a brief demo of a project built using PlushPal featuring a plush frog. When the facilitator acted out a set of gestures with the stuffed animal (*waking up*, *stretching*, and doing *jumping jacks*), each gesture triggered a different recorded sound ("I'm ready!", "Time to stretch!", and "Jumping Jacks! One! Two!").

After the demo, the participants were asked to introduce themselves and the plush toy they brought with them. Next, participants attached their micro:bit to their stuffed animal (all online participants chose to use hair ties, while a hand-sewn backpack was used in the in-person sessions) and connected their hardware to the PlushPal app. To familiarize children with the accelerometer, we asked participants to observe a graph of live accelerometer data in the PlushPal Console and gave them several minutes to experiment with moving their stuffed animal to observe how the three different colored lines (corresponding to x-,y-, and z-axis measurements) changed in response.

After these experiments with the sensor, the facilitator led a brief presentation about ML, defining ML as "a way for computers to accomplish a task by learning from examples." Children were told that they would capture their own examples of gestures using the micro:bit, and that the computer would use these samples to try to match their movements in real time with the closest example it has already seen.

Next, children were introduced to the PlushPal interface through a guided activity in which everyone created a project using three common gestures: *none*, *jump*, and *running*, gestures chosen because they could be applied to stuffed animals of any size or form factor. The facilitator demonstrated the process live, and participants followed along at each step. First, they recorded 3 *none* samples with the stuffed animal stationary in a neutral position. They then added 3 *jump* samples and 3 *running* samples. In this process, the facilitator described how providing more samples can be helpful for the computer to gain more knowledge of what the gesture looks like. The children were shown how to test their model in the PlushPal Console and how to record sounds and link them to gestures in the interface.

At this point, participants had 10 minutes to brainstorm the project they would build for the remainder of the workshop. Brainstorming was guided by a worksheet template via Google Slides where participants shared the name of their stuffed animal, its favorite place, favorite activities, and types of sound it may make. The template had three fill-in-the-blank sentences following the format, "When [**gesture**], my stuffed animal will respond with [**sound**]" to help participants envision concrete plans for building their PlushPal project.

Children were then given 20 minutes to create their interactive plush toy using PlushPal. Workshop facilitators were available to assist with technical questions. We asked questions along the way, and for the online workshops, we asked participants to show their progress when possible by sharing their screen and holding their stuffed animal up to their web camera. Before ending the workshop, participants were asked to demonstrate their project and asked questions including, "Can you tell me why you chose this set of gestures?" and "Were there any gestures that were hard to make work? Why do you think that is?"

4.2 Data Collection

We collected video recordings from each workshop as well as the children's brainstorming worksheets. The PlushPal app collected anonymous analytics on how users constructed their ML models, including when new samples, gestures, or sounds were added. We also saved JSON data of individual sessions to analyze accelerometer data from their samples.

Pre- and post-workshop surveys were used to understand how participants' knowledge of ML changed as a result of the workshop. Questions included asking what children thought of when they hear the term 'machine learning,' as well as how they thought a voice assistant works. In the post-survey, we asked several questions focused on their workshop experiences; after our initial online workshop pilot with 3 participants, we added two additional questions to the post-survey to evaluate how children would approach debugging an ML model. The questions pose an example scenario specific to PlushPal where the computer has trouble telling the difference between two gestures. We asked children to describe why they think the gestures are hard to tell apart and how they would try to fix it.

4.3 Data Analysis

Four researchers contributed to the analysis of the workshop data, transcribing the workshop recordings and collaboratively reading the transcripts, surveys, data logs, and final ML model data for each participant. We then met and discussed until we reached shared interpretations of how each child used PlushPal to create interactive stories. Visualizations were created for every session showing how models were created and revised throughout the activity. We collaborated on developing a set of codes using inductive coding [28] to identify themes that emerged from these multiple data sources, discussing and iterating over them in weekly meetings.

5 RESULTS

In total, 11 children aged from 8 to 14 (F=6, M=5) completed the PlushPal activity. While 3 mentioned that English was not their primary language spoken at home, all indicated that they were comfortable communicating in English. Four self-identified as Caucasian, five as Asian/Pacific islander, one multiple ethnicity, and one was not sure. Most (n=9) had some experience coding before (such as Scratch), but only 4 had used the micro:bit before. All had interacted with AI technologies like voice assistants or video recommendation systems. Table 1 shows a summary of the participants and the projects they created, where each child is given a pseudonym based on their stuffed animal character.

Children brought a variety of stuffed animals to use, including two fantasy characters (a dragon and unicorn) as well as multiple teddy bears (N=6), a bunny, a dog, and a cat. Eight brought plush toys with poseable arms and legs, and 4 brought stuffed animals they either borrowed or were given to them by a parent to use during the workshop, rather than ones they personally owned.

Children largely developed the set of gestures and sounds for their projects based on the context in which their story took place, with environments such as Disneyland, the playground, the beach, and a cafe. Three of the children described choosing their gestures by drawing from what they personally would do in the same situation: "I thought of what I'd do if I were meeting a new friend that I've never met" (Brian Bear). In one example, the participant Bunny drew from her experience interacting with an actual rabbit, describing her friend's pet rabbit and its activities. The project created by child Soccer Bear was inspired by her unique relationship with her toy; she shared how her bear was purchased at a soccer tournament and then went on to incorporate a *practice soccer* gesture in her project. Performing the gesture caused the mask-wearing bear to say, "Going to practice at home until COVID is over!"

Most participants worked on building a single project throughout the 20 minute design activity; three created multiple projects when they successfully completed their first project under 20 minutes. All but two participants (80%) updated their ML models (by adding additional samples or gestures) after recording their sounds, indicating that children went back and forth between programming and building their ML model.

We begin with a summary of the types of gestures and sounds used across all projects to illustrate the creative space participants were able to engage with through using PlushPal. We then provide descriptions of the children's experiences building ML models, first through two contrasting case studies in data sampling followed by a general summary of misconceptions and debugging strategies. Our goal is to illustrate specific ways these children reasoned about data through play, rather than draw generalizations from our small sample size. In doing so, we aim to highlight how children transformed their existing plush toys using PlushPal, while describing observed challenges children faced when building and understanding ML models.

5.1 How do children imagine bringing their stuffed animals to life using gestures and sounds?

Across all projects, children recorded 42 unique gestures and 45 unique sounds with an average of 4 gesture-sound pairs per project. A classification of their gestures is displayed in Figure 3, with 40% of gestures relating to exercises such as doing a *backflip* or *swimming*, approximately 10% having to do with daily routines like *eating* and *sleeping*, and another 10% involving play and recreation such as *building a sandcastle* and *riding a roller coaster*.

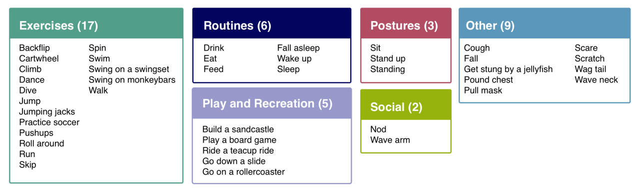

Figure 3: Summary of 42 unique gestures used in PlushPal projects

All but two of the gestures were generic, meaning they could be applied across any plush toy; for example, gestures like *feeding* or *climbing* do not rely on having a specific type of stuffed animal. In one case with a dragon stuffed animal, the unique physical qualities of the toy (its long neck and tail) inspired two form-specific gestures: a neck wave and a tail wag; this participant also expressed wanting to create a *flying* gesture using the physical wings of the dragon. Seven gestures (16%) used poseable arms, such as waving or chest pounding. In our study, 8 of the 11 plush toys (72%) children brought had the standard physical form of a teddy bear with poseable arms and legs – further work would be needed to see if a broader range of stuffed animal shapes might inspire different types of gestures.

Half of the unique gestures were anthropomorphic, with the plush toy mimicking human behavior (such as *doing pushups* or *riding* an amusement park ride), while the other half were movements real animals might make (such as *eating* or *scratching*). Younger children tended to choose diverse sets of gestures whereas older children tended to focus more on anthropomorphic gestures. These results suggest that children mixed animal behaviors with human qualities, which is also supported by the children's frequent use of speech in the sounds they created.

Children added 45 unique sounds that they recorded themselves, with 60% representing speech (such as "Yummy" and "Yahoo!") and the remaining 40% being sound effects (such as munching sounds or footsteps), most of which the children generated themselves through scratching and tapping noises. Six children recorded audio

Table 1: Summary of PlushPal activity participants

Participant	Age / Gender	Project Created [# gestures]	micro:bit experience
Soccer Bear	13 / F	A COVID-conscious soccer-playing bear [3]	yes
Cat	11 / F	A cat that runs around and scares others [6]	no
Dragon	11 / F	A dragon that chirps and wags its tail [6]	no
Panda	14 / M	A sleepy panda that snores when it falls asleep [5]	yes
Dog	12 / M	A dog that does pushups and goes to sleep [3]	yes
Macaron Bear	8 / M	A bear enjoying a day at Disneyland [3]	no
Brian Bear	10 / F	A bear doing activities at the park [3], at home [3], and at a cafe [3]	no
Bunny	10 / F	A bunny playing around in a field [3] and in its cage [3]	no
Jungle Bear	10 / M	A bear that pretends to be a gorilla in the jungle [3]	no
Unicorn	8 / F	A unicorn playing in a playground [5] and at the beach [3]	no
John Bear	9 / M	A bear playing in a park [3]	no

from sound clips found on the Internet, including a pop song and the sound of rustling leaves.

When asked about other ways they would want their toy to come to life, three participants wanted to program their stuffed animal to move on its own, describing having their stuffed animal dance, wave, or move its mouth precisely. One child expressed a wish to program her stuffed animal to communicate with other toys.

5.2 How did children engage with data science practices when building their ML models with PlushPal?

Our goal with PlushPal was to introduce children to the basics of ML and data science, particularly the general idea that ML enables computers to learn by detecting patterns when shown examples through data. Comparing participants' responses in our pre- and post-activity surveys (n=11 and n=10, as one participant did not fill out the post-survey) suggested that most children developed a sense of the term "machine learning" as referring to how computers learn through experience to "recognize things on their own."

In our pre-survey, we assessed children's understanding of ML by asking, "What comes to mind when you hear the term machine learning?" Only one child described computers being able to learn themselves ("robotics learning from each other and using their past commands to guess what is going to happen in the future"), while the remaining responses largely concerned *humans* learning with and about technology. Five of the 11 participants described ML as humans building with technology ("coding," "building with tools," "programming"), while four described humans learning with technology ("using machines to help aide [sic] in the learning process"). In our post-survey, 7 of the 10 respondents correctly described AI, computers, or machines learning or being taught ("teaching machines how to recognize things on their own"). Similarly, when asked how voice assistants like Siri work, five of the 10 respondents described some type of ML process ("listening to your voice and comparing it to the name of the song").

In our PlushPal activity, children engaged with several parts of the ML model building process, including data sampling and evaluating their models. Children added an average of 21 samples over the course of their 20 minute sessions. To illustrate how this looked in the context of play, we begin with two case studies highlighting contrasting approaches to building ML models. We then follow with a summary of practices observed across all participants.

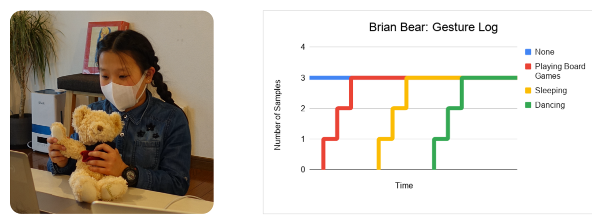

Figure 4: Brian Bear and Gesture Data Sampling Log from play session

5.2.1 Brian Bear: Intentional variation in data sampling. Brian Bear is an example where the child had a distinct approach to data sampling, incorporating samples that were intentionally different from one another. During her session, Brian Bear created a project inspired by activities in her room, incorporating three gestures: *playing board games* (with the bear slapping its hand down as if rolling a die), *sleeping* (the bear lying on its back), and *dancing*. Figure 4, displays how she systematically added three samples for each gesture without removing or editing any samples. She then added her own sound recordings, including a recording of herself saying "I Won!" for the *playing board games* gesture and her favorite pop songs when the bear *dances*.

While creating samples for each of her gestures, Brian Bear intentionally modified the gesture each time. Figure 5 displays an example of these differences for three samples of *sleeping*. In the first, she has the bear lying still on its back; in the second, the bear rolls to one side; in the third, she moves the bear to the other side, as if tossing and turning. This way of recording gestures was seen across multiple instances of her data sampling, suggesting that it was a deliberate strategy rather than mis-remembering how each gesture had been previously carried out. At this point, the researcher inquired about her intentions:

> **Researcher**: "I notice that when you're adding the different samples, you're doing something a little bit

different each time. Can you talk a little bit about that process?"

Brian Bear: "Yeah, I was doing that a little because... when [the bear] was sleeping, I did that [holding the bear still] and maybe make him move side to side [moves bear side to side]. So it sometimes stays [lying still] and sometimes goes [side-to-side]. So it [the computer] would recognize all of it."

Here, Brian Bear reveals that by intentionally recording distinct samples for the same gesture, she intended to cast a wider net of examples for the computer to associate with a given class (*sleeping*). Technically, this approach is valid and is often used for models trained with large training data sets, where capturing variation ensures a broad range of use cases can be correctly classified as the same label. However, this approach may be sub-optimal given the small number of samples (N=3), as she only had a single sample for each distinct sleeping pose.

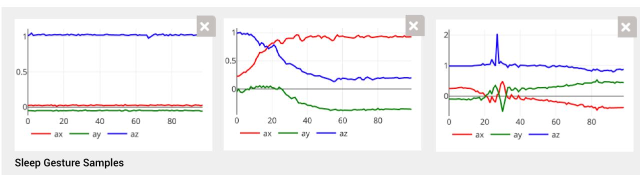

Figure 5: Three intentionally different Sleep gesture samples: lying still, moving to one side, and moving from side to side

Despite her process of creating variation in her data sampling, her project worked well when she tested it, likely because each gesture was quite distinct from the others. However, one issue that arose was that the "I won!" recording would sometimes play even when she wasn't performing the *playing board games* gesture, leading to this exchange:

Brian Bear: "It keeps playing 'I Won!'"

Researcher: "Why do you think it's doing that?"

Brian Bear: "Because I did this for 'I Won!' [slaps bear's hand down], and if you do this [does *dancing* gesture], it does this [shows that the arm movement is similar for both gestures]. I'm going to this [ties bear's hands behind its back and performs *dancing* gesture]."

Researcher: "Do you think the computer can tell when you move its [the bear's] hand then?"

Brian Bear: "Yeah, if you do that [does *playing board games* gesture], it means I won, and if you do that [*dance* gesture], it's the same [hand] movement."

In her response, Brian Bear shares her theory that the movement of the hand affects how the computer understands the gesture, even going so far as to suggest that constraining the bear's arms but performing the same gesture would reduce the chance of incorrect predictions. Here, the child suggests that the sensor is capable of understanding the movement of individual limbs of the stuffed animal, an idea that multiple children had about the micro:bit accelerometer. For example, the child Unicorn believed that her *skipping* and *swimming* gestures were conflicting because of the movement of the toy's arms, leading her to re-record *swimming* by moving the Unicorn's legs instead of its arms. Note that the micro:bit has no awareness of the limbs of the stuffed animal, though the movement of the arms can indirectly affect how much acceleration the micro:bit detects in a given direction due to its weight.

5.3 John Bear: Similar samples and experimenting with the micro:bit accelerometer

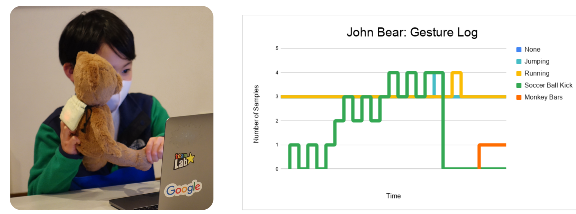

Figure 6: John Bear and Gesture log from play session

The project created by 10-year-old participant John Bear shows a contrasting approach to data sampling, where samples were repeatedly re-recorded in an attempt to make them as similar as possible. This process of removing and re-recording samples was seen in 8 of the 11 participants.

John Bear created a project inspired by activities in a park, reusing the *jumping* and *running* gestures from the introductory activity but adding his own third gesture. At first, he intended to create a gesture called *soccer ball kick* in which the bear kicks its right leg. However, he quickly found himself having to debug why *soccer ball kick* was incorrectly being identified, leading to a realization about how the micro:bit's accelerometer senses movement.

As shown in the gesture log in Figure 6, John Bear re-recorded *soccer ball kick* samples six times while prototyping. Throughout the process, he made comments like, "Kind of very different. I might delete that one," removing any samples that seemed visually dissimilar to the others and showing that he tried to make his kick samples highly correlated.

After adding several samples, the *soccer ball kick* gesture was unexpectedly identified instead of the *jumping* gesture by the ML model. His initial response was, "More data?," at which point he added another sample for both *soccer ball kick* and *jumping*. This revealed that he was aware that the number of samples affects the model. When recording the additional *jumping* sample, he inspected the existing samples, counted the three peaks, and then proceeded to record another sample, counting aloud as he moved the bear up and down three times. This again shows his care to make the samples similar while interpreting the sensor data to match peak count with the number of jumps.

Despite these changes, the ML model continued to confuse *soccer ball kick* with *jumping*. When asked why this might be, John Bear dragged his mouse over the sample graphs of both gestures in the PlushPal interface and stated, "These bumps [the single bump of *soccer ball kick*] are kind of like these bumps [mouses over a single bump in the *jumping* gesture]" (Figure 7). As he rested the bear down on the table, he found that this movement unexpectedly also triggered *soccer ball kick*, leading him to experiment even more; he made the bear fall onto its back and saw that the model falsely predicted *soccer ball kick* again. Then, looking at the live sensor stream in the PlushPal console, he identified that falling or sitting down result in a single-peak in the graph, similar to his *soccer ball kick* gesture.

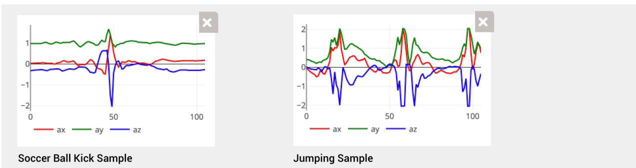

Figure 7: A soccer ball kick gesture sample and jumping gesture sample

In the process of debugging, John Bear started to see that even though the gestures are very different to the human eye (sitting versus kicking versus lying down), to the micro:bit sensor, the accelerometer readings appear quite similar to the point that the ML model incorrectly identifies his gestures. Ultimately, he decided to remove *soccer ball kick* in favor of a more dissimilar gesture of *monkey bars*, where the bear waves its arms as if swinging.

At the end of the session, John Bear shared a suggestion of using multiple sensors: "This could be expensive but maybe putting it [micro:bit] here, on the head or something. Because if it's only on the body, it only records this part [the body]. But if there's any movement here [the arm] or here [the foot] it wouldn't really record stuff." His hypothesis is that additional sensors would be needed to detect more granular movement in the limbs, as the sensor is currently attached to the toy's torso and may not easily detect movement at its extremities.

5.4 Other Model Building Practices

Generally, creating an accurate ML model with PlushPal relies on how distinct each gesture is along with the size of the training data; when users add additional gestures, it's generally a good idea to add additional samples per gesture and to capture samples that are internally consistent. One of the challenges with ML is determining an appropriate sample size to ensure that your model works accurately.

With PlushPal, children's projects ranged from having 3-8 distinct gestures (not including the *none* gesture used for calibration in all projects), with 6 of the 11 projects (54%) having five or more gestures even though the initial brainstorming activity only had them come up with 3 gestures. However, as children added more gestures to their projects, they did not necessarily increase their sample sizes; across all 56 non-unique recorded gestures, 82% had three or fewer samples. This was likely influenced by the introductory activity, where we had children add 3 samples per gesture, although the facilitator did mention that more samples may be needed when additional gestures are added. In fact, 5 of the 11 participants never added more than 3 samples for any gesture in their project. For projects like the one created by Unicorn, where there were 8 unique gestures but no more than 3 samples per gesture, this led to models that were highly variable, often falsely identifying gestures. While it is encouraging to see that the children did not have a shortage of ideas for gestures they wanted to add to their projects, these results suggest that having feedback about considering sample size would be valuable for helping kids debug ML models.

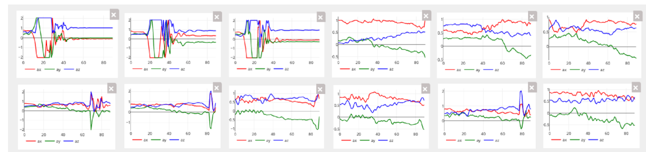

Figure 8: Macaron Bear added 12 samples to her *roller coaster* gesture in an attempt to debug, but the samples were not similar to one another.

Even when more than 3 samples were added, children may not have created similar samples for a given gesture. In one instance, a participant created 12 samples of a *roller coaster* gesture, but because the samples were dissimilar (shown in Figure 8), her gesture was often incorrectly triggered.

In our post-survey, children were asked to debug a hypothetical PlushPal scenario where the computer could not tell the difference between two gestures. Of the 7 (of 8) children that responded to this question, 6 recognized that the gestures appeared to be too similar to one another ("because the [sic] look similar in motion", "because the graphs are similar."). Three suggested increasing sample size, two proposed making the gestures more distinct from one another, two indicated that they would re-record one of the samples, and one said they would use a different gesture altogether. Other than similarity or difference, participants overall included little explicit description of what about the data made them similar or different. These results suggest that children were able to start to understand the role of sample size and similarity between samples but that further work is needed to help children realize the importance of using these strategies in combination, and how to reason about what features of a data sample meaningfully make them "different" or "similar."

6 DISCUSSION

With PlushPal, we were interested in how children might bring their plush toys to life and supported this by providing a tool for children to train an ML model to recognize gestures of their own design. Our analysis indicated that PlushPal flexibly supports a range of gestures, with 42 unique gestures created by the 11 children. From anthropomorphic to realistic movements, children's gestures represented several categories, the largest of which was exercising (like *jumping jacks* or *diving*), followed by play and recreation (such as *going down a slide*) and movements for routines like *sleeping*. Almost all the gestures used were not specific to the form factor of the plush toy, supporting the idea that PlushPal can flexibly support a wide range of play scenarios regardless of stuffed animal form factor.

More than half of the participants generated stories inspired by their own personal experiences, either through their relationship with the toy itself or a personal memory like going to the beach; this suggests that PlushPal is able to support personally meaningful projects. Participants linked their gestures with recordings of their own voices, expressing what their toy would say or mimicking the sound the toy would make. While children are commonly observed using their own voices to animate toys during narrative play, the ability to record and trigger sounds in response to gestures provided them with another creative layer they could customize.

We observed that PlushPal further inspired the imagination of children as they developed their projects. Four of the 11 participants added more gestures than they originally described in the initial brainstorming activity, indicating that they were inspired to try new ideas as they were creating with PlushPal. However, we also saw that when children added more gestures, they did not generally add more samples per gesture, as 82% of all gestures added had three or fewer samples. Having guidance for users to consider sample size as they increase the number of unique gestures they want their ML model to recognize may help support children in creating accurate and successful models.

Another focus of this study was how children engage with ML practices through play with PlushPal. Our post-survey indicated that using PlushPal helped children gain a foundational understanding of ML, transitioning from thinking of the term as a way for humans to learn from computers or programming to considering ML as a practice in which computers can learn on their own. This is a promising result especially considering the short length of the activity. We observed how PlushPal provided a space for children to experiment with understanding accelerometer sensors, engage in data collection, and practice debugging when troubleshooting their ML models. The majority of participants (8 of 11) actively re-recorded samples of their gestures and used terms like 'similar,' 'consistent,' and 'pattern' to verbally describe the process of selecting samples to keep and to remove, indicating that the visualization of accelerometer data encouraged participants to investigate how to better create stronger training data from the computer's point of view. In our post-survey question asking about debugging practices, children identified sample size and similarity between samples as factors for improving an ML model, but in practice, we observed that children did not necessarily apply these strategies to their own projects, especially with regard to increasing sample size with more gestures.

A notable misconception was how aware the micro:bit is of the movement of individual limbs of the toy. In our case studies with Brian Bear and John Bear, we show how in the former, Brian Bear thought that restricting arm movement but performing the same gesture would help the model; in contrast, John Bear saw through experimentation that very different gestures involving limbs (such as sitting and kicking) could trigger similar accelerometer readings. This finding suggests that the practice of building ML models can help provide a playground for children to realize potential limitations of what computers are able to sense more broadly.

7 LIMITATIONS AND FUTURE DIRECTIONS

The pandemic greatly restricted in-person interactions and forced us to host some workshops remotely with participants. While the workshop activity remained the same, the presence of the researchers in the same room for the in-person session might have impacted the way participants developed their ideas and artifacts differently from those who participated remotely. Since parents and teachers signed their children up for the workshop, the children's interest in computing may be generally high; testing PlushPal with a larger number of children with more diverse backgrounds may provide further data on the applicability of this activity for broader audiences.

8 CONCLUSION

With PlushPal, we present a tool that enables children to experiment with accelerometer sensing and ML, all while engaging in a playful activity where they bring their plush toys to life with custom gestures and sounds. Through studying how 11 children between 8-14 years old used PlushPal to power their own creative storytelling, we make several contributions to emerging research on the creative potential of combining play and ML for young people.

First, we provide a design space of the types of gestures children implemented in their designs, highlighting common categories like exercises and play and recreation that can inform the design of interactive toys more generally. Second, we describe different approaches children had for collecting data samples and highlight potential misconceptions, both in how ML models work but also what sensors are able to detect. In particular, while we saw that most participants actively re-recorded samples to improve their ML model, we also observed an instance in which a child intentionally recorded dissimilar samples in an effort to make the computer associate different movements with the same class. Some children also believed the micro:bit sensor was smarter than it actually is and capable of detecting precise movement of individual limbs, suggesting opportunities to help beginners better explore the boundaries of what a computer can and cannot sense. We intentionally did not incorporate specific feedback into the PlushPal interface itself to observe how children would approach debugging ML models on their own. Our evaluation showed that children have many ideas for gestures they want to add but may benefit from support around considering how sample size and similarity across samples affect ML accuracy as they increase the number of classes they want their model to detect.

Overall, we are encouraged that the children who used PlushPal were able to gain direct experience in the basics of data sampling and model evaluation in their short time building projects, and we show the creative potential for ML to support new forms of storytelling and interactive play.

ACKNOWLEDGMENTS

Thank you to Yoshimi Ueda, Junko Takayama, and Sarah Manger for their help with organizing our PlushPal play sessions.

9 SELECTION AND PARTICIPATION OF CHILDREN

Children who participated in this study were recruited through their teachers and parents who saw our invitation to the study posted on social media and online forums for K-12 educators. We asked the teachers or parents to fill out an interest form and selected students who were between 8-14 years old. For the online workshops, students needed to have a micro:bit they could use. For in-person sessions, we recruited from a community of International Schools with students comfortable speaking English, and children did not need to have their own micro:bits. We asked students and their parents to each review and sign an online consent before participating. In those consent forms, we communicated that their participation was voluntary and that their data may be shared with research communities after removing any identifiable information. This study has been approved through the Institutional Review Board of the researchers' respective institutions.

REFERENCES

[1] Adam Agassi, Hadas Erel, Iddo Yehoshua Wald, and Oren Zuckerman. 2019. Scratch nodes ML: A playful system for children to create gesture recognition classifiers. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[2] Doris Bergen. 2002. The role of pretend play in children's cognitive development. *Early Childhood Research & Practice* 4, 1 (2002), n1.

[3] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. 2020. Teachable machine: Approachable Web-based tool for exploring machine learning classification. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–8.

[4] Keywon Chung, Michael Shilman, Chris Merrill, and Hiroshi Ishii. 2010. OnObject: gestural play with tagged everyday objects. In *Adjunct proceedings of the 23nd annual ACM symposium on user interface software and technology*. 379–380.

[5] Edward P Clapp, Jessica Ross, Jennifer O Ryan, and Shari Tishman. 2016. *Maker-centered learning: Empowering young people to shape their worlds*. John Wiley & Sons.

[6] Stefania Druga. 2018. *Growing up with AI: Cognimates: from coding to teaching machines*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[7] Stefania Druga, Sarah T Vu, Eesh Likhith, and Tammy Qiu. 2019. Inclusive AI literacy for kids around the world. In *Proceedings of FabLearn 2019*. 104–111.

[8] Rebecca Fiebrink and Perry R Cook. 2010. The Wekinator: a system for real-time, interactive machine learning in music. In *Proceedings of The Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)(Utrecht)*, Vol. 3.

[9] Luc Geurts, Jolien Deville, Vero Vanden Abeele, Jelle Saldien, and Karen Vanderloock. 2014. Skweezee Studio: Turn your own plush toys into interactive squeezable objects. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*. 377–380.

[10] Thalia R Goldstein and Ellen Winner. 2011. Engagement in role play, pretense, and acting classes predict advanced theory of mind skill in middle childhood. *Imagination, Cognition and Personality* 30, 3 (2011), 249–258.

[11] Kreg Ryan Hanning. 2018. *Tinkering with ScratchBit: Explorations in blended making*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[12] Birte Heinemann, Simone Opel, Lea Budde, Carsten Schulte, Daniel Frischemeier, Rolf Biehler, Susanne Podworny, and Thomas Wassong. 2018. Drafting a data science curriculum for secondary schools. In *Proceedings of the 18th Koli calling international conference on computing education research*. 1–5.

[13] Tom Hitron, Yoav Orlev, Iddo Wald, Ariel Shamir, Hadas Erel, and Oren Zuckerman. 2019. Can children understand machine learning concepts? the effect of uncovering black boxes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[14] Tom Hitron, Iddo Wald, Hadas Erel, and Oren Zuckerman. 2018. Introducing children to machine learning concepts through hands-on experience. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*. 563–568.

[15] Yingdan Huang and Michael Eisenberg. 2010. Plushbot: an application for the design of programmable, interactive stuffed toys. In *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. 257–260.

[16] MIT App Invetor. [n.d.]. MIT App Inventor. https://appinventor.mit.edu/

[17] James E Johnson, James F Christie, Thomas D Yawkey, and Francis Photo Wardle. 1987. *Play and early childhood development*. Scott, Foresman & Co.

[18] Ken Kahn and Niall Winters. 2017. Child-friendly programming interfaces to AI cloud services. In *European Conference on Technology Enhanced Learning*. Springer, 566–570.

[19] Rohit J Kate. 2016. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery* 30, 2 (2016), 283–312.

[20] Kaho Kato, Naoto Ienaga, and Yuta Sugiura. 2019. Motion Estimation of Plush Toys Through Detachable Acceleration Sensor Module and Machine Learning. In *International Conference on Human-Computer Interaction*. Springer, 279–286.

[21] Barry M Kudrowitz and David R Wallace. 2010. The play pyramid: A play classification and ideation tool for toy design. *International Journal of Arts and Technology* 3, 1 (2010), 36–56.

[22] Dale Lane. 2017. Machine Learning for Kids. https://machinelearningforkids.co.uk/

[23] Duri Long and Brian Magerko. 2020. What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–16.

[24] Suzanne Low, Yuta Sugiura, Kevin Fan, and Masahiko Inami. 2013. Cuddly: enchant your soft objects with a mobile phone. In *International Conference on Advances in Computer Entertainment Technology*. Springer, 138–151.

[25] David A Mellis, Ben Zhang, Audrey Leung, and Björn Hartmann. 2017. Machine learning for makers: Interactive sensor data classification based on augmented code examples. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 1213–1225.

[26] ml5js. 2021. ml5js: Friendly Machine Learning For the Web. https://ml5js.org/

[27] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[28] Johnny Saldaña. 2021. *The coding manual for qualitative researchers*. SAGE Publications Limited.

[29] Samsu Sempena, Nur Ulfa Maulidevi, and Peb Ruswono Aryan. 2011. Human action recognition using dynamic time warping. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE, 1–5.

[30] Eric D Smith and Angeline S Lillard. 2012. Play on: Retrospective reports of the persistence of pretend play into middle childhood. *Journal of Cognition and Development* 13, 4 (2012), 524–549.

[31] 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.

[32] David Touretzky, Christina Gardner-McCune, Fred Martin, and Deborah Seehorn. 2019. Envisioning AI for k-12: What should every child know about AI?. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9795–9799.

[33] Teemu Valtonen, Matti Tedre, Kati Mäkitalo, and Henriikka Vartiainen. 2019. Media literacy education in the age of machine learning. *Journal of Media Literacy Education* 11, 2 (2019), 20–36.

[34] Abigail Zimmermann-Niefield, Shawn Polson, Celeste Moreno, and R Benjamin Shapiro. 2020. Youth making machine learning models for gesture-controlled interactive media. In *Proceedings of the Interaction Design and Children Conference*. 63–74.

[35] Abigail Zimmermann-Niefield, Makenna Turner, Bridget Murphy, Shaun K Kane, and R Benjamin Shapiro. 2019. Youth learning machine learning through building models of athletic moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. 121–132.