How do People Train a Machine? Strategies and (Mis)Understandings

TÉO SANCHEZ, Université Paris-Saclay, Inria, LISN, France
BAPTISTE CARAMIAUX, Université Paris-Saclay, CNRS, Inria, LISN & Sorbonne Université, ISIR, France
JULES FRANÇOISE, Université Paris-Saclay, CNRS, LISN, France
FRÉDÉRIC BEVILACQUA, STMS IRCAM-CNRS-Sorbonne Université, France
WENDY E. MACKAY, Université Paris-Saclay, CNRS, Inria, LISN, France

Machine learning systems became pervasive in modern interactive technology but provide users with little, if any, agency with respect to how their models are trained from data. In this paper, we are interested in the way novices handle learning algorithms, what they understand from their behavior and what strategy they may use to "make it work". We developed a web-based sketch recognition algorithm based on Deep Neural Network (DNN), called *Marcelle-Sketch*, that end-users can train incrementally. We present an experimental study that investigate people's strategies and (mis)understandings in a realistic algorithm-teaching task. Our study involved 12 participants who performed individual teaching sessions using a think-aloud protocol. Our results show that participants adopted heterogeneous strategies in which variability affected the model performances. We highlighted the importance of sketch sequencing, particularly at the early stage of the teaching task. We also found that users' understanding is facilitated by simple operations on drawings, while confusions are caused by certain inherent properties of DNN. From these findings, we propose implications for design of IML systems dedicated to novices and discuss the socio-cultural aspect of this research.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**.

Additional Key Words and Phrases: Interactive Machine Learning; Human-AI Interaction; Sketch recognition; Human-centered analysis

ACM Reference Format:
Téo Sanchez, Baptiste Caramiaux, Jules Françoise, Frédéric Bevilacqua, and Wendy E. Mackay. 2021. How do People Train a Machine? Strategies and (Mis)Understandings. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 162 (April 2021), 26 pages. https://doi.org/10.1145/3449236

1 INTRODUCTION

Machine Learning (ML) focuses on building computer algorithms that learn from data, enabling them to extract patterns and make predictions from unknown input data [25]. Over the past decade, machine learning has been included in an ever-increasing number of complex tasks, including recognising speech, identifying elements in images, or generating realistic music and videos. The increasing expressivity of machine learning based systems, in terms of task complexity and

Authors' addresses: Téo Sanchez, Université Paris-Saclay, Inria, LISN , Gif-sur-Yvette, France, teo.sanchez@inria.fr; Baptiste Caramiaux, Université Paris-Saclay, CNRS, Inria, LISN & Sorbonne Université, ISIR, Paris, France, baptiste.caramiaux@sorbonne-universite.fr; Jules Françoise, Université Paris-Saclay, CNRS, LISN, Gif-sur-Yvette, France, jules.françoise@lisn.fr; Frédéric Bevilacqua, STMS IRCAM-CNRS-Sorbonne Université, Paris, France, frederic.bevilacqua@ircam.fr; Wendy E. Mackay, Université Paris-Saclay, CNRS, Inria, LISN, Gif-sur-Yvette, France, mackay@lri.fr.

2573-0142/2021/4-ART162 $15.00
https://doi.org/10.1145/3449236

diversity, has been widely covered by the media and has rendered them ubiquitous in user-centered applications.

However, most systems are designed to simply display the output of the machine learning model and end users rarely have agency in how such models are trained. When building applications with embedded ML capacities, the typical workflow comprises two separate phases: *training* and *testing*. Conventional machine learning involves a training phase that occurs offline, behind the scenes, and remains opaque to end users. Modern algorithms are trained on large sets of sample data. For example, image recognition tasks use sample data with annotated sets of images, which ensures a textual description of their content, e.g. ImageNet [9]. Once trained, the model is "frozen" and can be deployed in an interactive application where it performs predictions based on user input. In this scenario, end users are never involved in the training phase, which limits their understanding of the system's behavior. This can result in undesirable or even dangerous consequences in safety-critical or sensitive application domains, and also prevents the general public from understanding of such technology.

The fields of Human-Computer Interaction (HCI) and Computer-Supported Collaborative Work (CSCW) have explored alternative ways to re-balance the role of human users in machine learning based systems [2, 18, 47]. In particular, Interactive Machine Learning (IML) [11, 46] focuses on the process by which users interact with the different stages of a machine learning pipeline. The goal is to improve the building of machine learning models, to make them more accurate, transparent, less biased, and easier to understand.

A typical IML workflow is incremental: users iterate between *training* the model by editing the training data, the model architecture and parameters; and *testing* the model by computing various metrics for additional data, e.g. validation datasets, test sets or direct interaction with the trained model. This approach lets domain-expert users personalize existing models with well-curated data, without machine learning expertise. For example, artists can benefit from this approach to train a model with their own data, then explore the model's expressivity, and stop when they 'feel' it reaches their desired level of performance based on their expectations and artistic needs [13, 14, 16]. ML developers also benefit from IML-based approaches, since they can leverage the iterative process of building a ML based system for both debugging and improve its accuracy [11, 26, 43].

The general public is however rarely considered in research in IML and Human-AI interaction. The general public includes novice end users who are not literate in machine learning nor computer science, and who may not be using ML within a specific practice, like music or design. Therefore, we know little about how novice users understand learning algorithms: we do not know how they interpret the system's behavior, nor do we understand which strategies they would use to teach these algorithms.

Exploring how general public interacts with learning algorithms is important: First, it can offer us insights on new guidelines for designing rich interactions with ML based systems, following an important line of previous research in the field [42, 50]. Second, it can bring the technology closer to people such as empowering them in their activities [2, 28], and fostering its democratisation. Third, it can foster ML education, which has been scarcely studied in the field [12]. Finally, gaining insight into how the general public interacts with machine learning systems, and in particular the *learning* part of the process, has the potential to increase our understanding of *machine behaviour* [36], and highlight the contextual and socio-cultural influences of Human-ML (and Human-AI) interaction.

In order to explore how novices can train a machine learning system, we focus on the specific use case of a sketch-based recognition algorithm. In this scenario, the goal is to train a recognition system by drawing sketches associated to a set of categories. The system is incrementally trained and the predictions produced from drawings are used as inputs to monitor its accuracy. We refer to this task as *machine teaching* [40]. This scenario allows us to: (1) **identify novice teaching**

strategies for an image recognition algorithm; (2) **investigate novice understanding** of the machine behavior; and (3) **identify guidelines for designing IML systems dedicated to the general public**.

This paper offers two key contributions. As a technical contribution, we developed a novel web-based IML system called *Marcelle-Sketch*, which is fed by sketches drawn in the interface, and trained incrementally, one drawing at a time. The system uses drawings and associated category labels proposed by the users to make prediction on new sketches (i.e. recognize the sketch's category). The system was designed so the user can easily monitor changes in the model predictions and confidence levels while they are drawing, enabling tight interaction cycles. We used this system to probe how users can explore different strategies to train the system incrementally.

The core contribution stems from an experimental study inspecting the use of *Marcelle-Sketch* by novices. We present a set of quantitative and qualitative findings about users' teaching strategies and users' understanding (or misunderstanding) of the system's behavior, from which we draw a set of implications for the design of IML system dedicated to novices.

Finally, we discuss the concepts of teaching strategies by novices, teaching curriculum, and the socio-cultural implications of this work.

2 BACKGROUND AND RELATED WORK

Our work draws upon previous research in Interactive Machine Learning, in particular previous studies looking at novice users interacting with machine learning. Our work also draws from the machine teaching literature, although we adopt a different perspective. Finally, we review previous research in ML education.

2.1 Novice users interacting with machine learning

In this paper we are interested in end users who are not literate in ML or Computer Science (CS). In the rest of the paper, we will call these end users **novice users**. Building machine learning with a particular focus on this population is at the core of the Human-centred machine learning approach [2, 17, 47].

Stumpf et al. [42] conducted experiments to design rich interactions between end users, novice in ML, and ML based systems. Their working hypothesis is that fostering user understanding and trust of the system would lead to more robust system design. They highlighted the importance of the system's ability to provide suitable explanations, the need to take into account the user's adaptation and correction signals. More recently, Yang et al. [50] also conducted interviews to understand opportunities and pitfalls that novices raise when building ML solutions for themselves in real life, such as relying too much on the accuracy score as a performance measure for the deployment. Oh et al. [34] designed a research probe that predicts aesthetic scores of photographs to investigate how different user communities (AI experts, photographers and general public) reason about AI algorithm feedback in the subjective domain. However, the users had no agency in the training and could only predict on new images.

Novice users can be expert in another domains. They may therefore have alternative ways to assess a model performance, not only focusing on quantitative measures about the model's ability to generalize on new data, but also based on the user perception or expectations. As an example in a musical context, Fiebrink et al. [14] found that non-expert users use a variety of assessment criteria. They may use accuracy and cost the same way as ML practitioners would do, but they may also use qualitative measures such as unexpectedness, and direct evaluation to reflect on the data they provided. Similar research has looked at IML as an opportunity for design. Data-driven design can help designers, who are usually non-experts in CS, building rich interactions and improving

user experience [10, 48]. However, recent works have highlighted the inherent limitations of this approach stemming from the difficulties to understand the scope of ML possibilities [49].

In this work, we target novice users taken from the general public. However, our goal is not to empower novices in their practice, but rather to engage this population in exploring machine learning and reflecting about it.

2.2 Machine teaching

The term Machine Teaching (MT) was first introduced as a theoretical problem related to machine learning where the goal was to find the minimal set of examples to make a ML algorithm reach a pre-defined target state [29, 30, 39, 51]. Simard et al. [41] proposed another view of MT within the field of HCI. MT is defined as a means to "make the 'teacher' more productive at building machine learning models" as opposed to classical machine learning research that aims at "making the 'learner' better by improving ML algorithms". The authors promoted research on making the interactive process of teaching machines "easy, fast and universally accessible".

Research in Machine Teaching involved studies that try to elicit users behavior in a realistic teaching task. Hohman et al. [21] showed that ML practitioners improve model performance mostly by iterating on their data (i.e. collecting new data, adding labels) rather than iterating on the model (i.e. architecture and hyper-parameters). However, teaching strategies used by novices remain poorly understood. In a recent study, Hong et al. [22] investigated how participants trained and deployed an image recognition application using images taken with their mobile phones. They found that participants involve diversity in the set of images used to train. Images varied in terms of size, viewpoint, location, and illumination. They contextualized training set analysis with users background and model performance, and discussed how future teachable interfaces can anticipate users tendencies, misconceptions, and assumptions. Wall et al. [45] investigated how Machine Teaching experts teach an article classifier and designed guidance based on their teaching patters. Although guidance did not affect performances of models trained by novices, the authors found that the teaching task was less demanding for novices (mental load and effort) with guidance (notifications) than without. Our use-case differs since we are interested in a task that involved more idiosyncratic data, and we focus on a think-aloud protocol to elicit user's thought while they are interacting with the system.

More in-depth studies on teaching strategies can be found in the field of Robot-Human Interaction. Cakmak and Thomaz [7] studied how humans spontaneously teach a binary classifier from both a finite example set and by sample generation. They found that humans do not spontaneously generate optimal teaching sequences but this can be leveraged by giving a teaching guidance. Thomaz and Breazeal [44] argued that humans prefer to teach robots as social learners. They conducted a study where participants were asked to teach a virtual reinforcement learning robot to perform a new task. From their observations, they found that participants tend to give more positive than negative rewards. Interestingly, these last two studies involved incremental teaching mechanisms that allowed to inspect changes in behaviors and co-adaptation mechanisms.

In this paper, we look at incremental teaching of a sketch-based recognition system. Therefore, an originality of our work is to consider data that is created by the users (drawn sketches) and sequences of input data curated by users, meaning that users can choose which examples to provide and in which order. In addition, we will consider machine learning models (deep neural networks) that are able to learn rich data representation.

2.3 Transmitting ML to various audiences

Machine teaching by novice users can also be seen as a powerful tool for ML education and democratization. Targeting children, Agassi et al. [1] designed a gesture recognition IML component

in Scratch, a visual programming language dedicated to children [37]. The IML block is associated to a physical device with embedded accelerometers. The authors aimed at encouraging children to include gesture recognition in their Scratch project, allowing them to collect gesture data by themselves and train the model through trials-and-error. The authors argued that fostering an early understanding of ML processes through game and direct manipulation can later help children to understand more complex ML systems. In another work, Hitron et al. [20] showed that teaching a gesture-based recognition system fosters children understanding of machine learning mechanisms and this knowledge can be transferred to applications from everyday life. A similar approach has also been explored in sport [52].

Considering artists and creatives, Morris and Fiebrink [33] argued that ML can play a role for music pedagogy to help students engaging high-level creativity and social interaction without sensorimotor skills e.g. mastering a musical instrument, or academic knowledge e.g. mastering music theory. The importance of research in ML education in creative practice has recently been discussed [12], pointing out that there are very few works on teaching ML to various populations. It is unclear what (non ML/CS) students need to know about ML and what they can already intuit based on their prior knowledge.

Finally, engaging various audiences in ML involves the development of appropriate tools, involving high interactivity and "low-entry fee". As an example, Carney et al. [8] built “Teachable Machine”, an online tool to create and custom personalized machine learning classifier without technical machine learning expertise. Their goal was to help students, teachers or others, to learn, teach and explore ML concepts through interaction. Regarding drawings, few works have proposed drawing-support tools [19, 35], however, as far as we know none allowed users to explicitly teach and personalise the system.

The system we designed can be related to "Teachable Machine" since it is an IML system allowing to train a classifier with user-generated inputs. In image-based "Teachable Machine" scenario, images are collected in batch with a "hold to record". Our system differs since sketches inputs can only be drawn one by one. The dynamic of the interaction with Marcelle-Sketch is more incremental and considerably different from "Teachable Machine". We also provide prediction feedback after each strokes allowing new ways to evaluate the system.

In this paper, we contribute to the democratization of ML towards the large public through the development of a web-based system, teachable by end users. In addition, we will inspect how the use of the system could give insights on human-machine behaviors, and facilitate ML understanding.

3 *MARCELLE-SKETCH*: A TEACHABLE SKETCH-BASED RECOGNITION SYSTEM IN THE BROWSER

This section presents *Marcelle-Sketch*, an online sketch-based recognition system, teachable by end users. The application runs in a web browser and is available online[1].

3.1 Context and design motivations

The creation of *Marcelle-Sketch* originated from a collaboration with the association *Traces*, a think-and-do, nonprofit group interested in science, its communication and its relationship with society[2]. To continue their scientific mediation mission during the pandemic crisis, *Traces* organized weekly virtual sessions addressing a wide range of scientific topics to the general public. We collaborated with them for the first session of the series. The topic was about Artificial Neural Networks and

[1]https://marcelle-sketch.netlify.app/
[2]https://www.groupe-traces.fr/en/traces/

it was organised on the Twitch streaming platform. We specifically designed and implemented *Marcelle-Sketch* for this session (further described in the next Section 4).

The design of the application was steered by three important requirements:

- People should be able to produce their own data to teach the system;
- People should receive immediate feedback about the model's predictions and uncertainty;
- People should be able to use the application anywhere and easily.

With the first requirement, we aim to involve users in the generation and curation of the training examples. We are interested in studying the teaching strategies that emerge when users are free to change the input data in response to the system's outcomes. We use drawn sketches as inputs because they do not require specific expertise, and they are personal.

Second, people need to be able to interpret the model's predictions. Model's predictions always embed a level of uncertainty that is also important to convey to the users. A common feedback strategy consists in displaying likelihoods, i.e. values between 0 and 1 conveying the confidence level that the input instance belongs to each class. In addition to likelihoods, we use another approach that estimates this uncertainty using model ensembles (see details in Section 3.3.2).

Third, our goal was to inspect real-world use of the system by novice users. As such, we brought a particular attention to designing an application that can run online and that is easy to use.

Altogether, *Marcelle-sketch* is thought as a tool to probe novices' teaching strategies and understanding of a sketch recognition system.

3.2 Application overview

Marcelle-sketch is a dashboard composed of two panels, as depicted in Figure 1. The left-side panel is dedicated to inputs. It exposes a white canvas where users can create drawings. It also allows for data management such as dataset download or upload. The right-side panel is dedicated to prediction, training and data visualization.

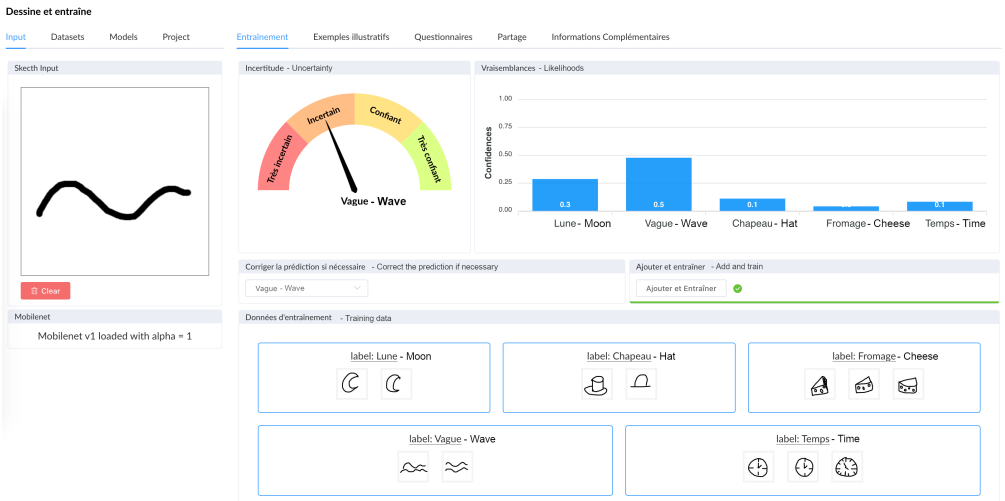


Fig. 1. Application interface used in the think-aloud study. Miniatures of the drawings are displayed on the main screen (bottom right component). We removed the history component and the possibility to change the line width and color of the drawing pen.

The workflow is as follows. The user starts drawing a line ("sketch input") and releases the mouse button. Predictions are automatically updated (chart bars) as well as the prediction uncertainty

(gauge). The user also receives a feedback on the predicted label (drop-down menu below the gauge). If the user wants to correct the prediction, they can click on the drop-down menu and select the correct label then click on the button to update the training set and launch training. Training is fast (few seconds). Once the training is done, both the prediction and uncertainty are automatically updated using the newly trained model. The user could also choose not to add the drawing to the training set and keep adding elements to their drawings, inspecting the changes in predictions and uncertainty.

The application is built using *Vue.js*, a JavaScript frontend framework. Each component displayed on the interface is a Vue component and is reactive, meaning that if a change is made on data (e.g. a new sketch or a new prediction made by the model), the components' display is updated automatically. The server was built with *Node.js* and the data were saved in a MongoDB database.

3.3 Machine learning pipeline and technical features

Marcelle-sketch is designed to allow for online and fast learning. The key technical features of the applications are presented in this section. The machine learning pipeline is divided between an encoder used to extract features from the raw image representing the user's sketch, and a classifier (built on top of the encoder) which is trained interactively by the user.

3.3.1 Machine teaching through transfer learning. Deep Neural Networks are well suited for building interactions involving rich and complex input data such as drawings. The drawback of these methods, however, is that they require a large amount of data to be trained. A way to overcome the data limitation is to use transfer learning [38]. Specifically, one can train a deep neural network on a given annotated image dataset and then use the stacked layers of the model as an encoder, from which it is easy to train a second (small) classifier using much less data. In the application, we used a pre-trained deep neural network model called MobileNet [23], which is suited for image classification. Its architecture uses depth-wise separable convolutions to build light weight embedding. The weights are initialized randomly when the application is loaded, the training is then incremental and only the learning rate is reset at each update. The use of transfer learning through MobileNet allows a simpler classifier to be trained quicker (few seconds), using very limited data (below 100 instances), which is critical in our scenario. Then, the method can be made more robust by assessing the model uncertainty, which is the second technical feature presented in the next section.

3.3.2 Classification with uncertainty estimation. Technically, one originality of our approach is to use an ensemble of models, called *Deep Ensembles* [27], in order to improve model performance under limited data and to allow uncertainty estimation. *Deep Ensembles* involve to train a set of N distinct classifiers (initialised randomly). Each classifier is trained independently and their individual predictions are combined to produce the final prediction. In the proposed system, we built an ensemble model comprised of 5 Multilayer Perceptrons (MLPs), on the top of the MobileNet encoder, initialized with random weights. The ensemble is trained in parallel with the user data: the 5 MLPs are simultaneously learning the mapping between the MobileNet features and the 5 pre-defined classes.

The benefit of having an ensemble of models trained in parallel is the possibility to compute an estimation of the model uncertainty over the predictions. We used variation ratio as estimator of the prediction uncertainty [4], defined as the number of models that agree on the same class divided by the number of models. In other words, the variation ratio can take five values of uncertainty: between $1/5 = 0.2$ (all the model of the ensemble disagree on the prediction) and $5/5 = 1$ (when all the models in the ensemble agree on the prediction). We mapped its values to four categories: "Very uncertain" ($ratio \leq 0.4$), "Uncertain" ($ratio = 0.6$), "Rather confident" ($ratio = 0.8$) and "Confident"

($ratio = 1$). This uncertainty is displayed to the user on the gauge of the component "*incertitude*" on figure 1.

4 PILOT WORKSHOP EXPLORING MACHINE TEACHING WITH THE GENERAL PUBLIC

In this section, we present a pilot workshop we conducted during the live session on the Twitch platform in collaboration with the *Traces* association. The live stream lasted about 90 minutes and was divided into three parts. The association moderators started with an introduction to artificial neural networks that lasted around 20 minutes. Then, we conducted the machine teaching workshop for 40 minutes using the *Marcelle-Sketch* application presented in Section 3. Finally, we answered questions from the audience asked via the chat window during 20 minutes.

4.1 Procedure and participants

After the introduction on neural network, we started the workshop by introducing the interface to the audience. We sent them a link to the application in the chat and participants opened it in a new tab. If they had questions, they could communicate them via the Twitch chat. One of the moderator gave a live demonstration of the application while a researcher was giving the explanations.

We chose a pre-defined set of categories to structure and focus the observations on the teaching strategies. Participants could not train on new custom category. Enabling the creation of new categories would have made difficult the comparison between participants' strategies, both in the pilot workshop and the study (presented in the following section). The number of categories was fixed and their labels pre-defined: "Moon", "Hat", "Wave","Cheese" and "Time". The model was initialized with random parameters (except for the MobileNet embedding) at loading and training set was empty. We asked the participants to train the system until the model was accurate and confident about the predictions for each category. We gave them 20 minutes to perform the task at their own pace. After that, we explained how to share their project publicly. Then, we asked them to load a model from another participant and explore this model with their own drawings. Finally, we invited the audience to answer an online questionnaire about the training process, posted in the text chat.

At the beginning of the workshop, 160 people were connected to the stream live. The number decreased during the session until reaching 84 people at the end of the session. Among these participants, 22 participants made their *Marcelle-Sketch* project public, i.e. we could analyse the data of 22 participants. 7 participants answered the online questionnaire.

4.2 Data collection and analysis

We analysed participants' use of the system afterward, by collecting images after every strokes they did, including all the predictions from the classifier, and the label they chose for the training data. From the whole set of projects, we removed 3 projects that were submitted twice, and kept only the projects with at least one drawing per category. Eventually, we kept 14 projects from 14 participants over the 22 projected submitted. The data included the images (png format), the timestamps, the predicted/trained category and the features from the MobileNet network, after each strokes made on the interface. Our analysis focused on the order in which categories are trained, the proportion of discarded images and the variability of the images.

4.3 Insights and limitations

The analysis highlighted that most participants iterated quickly across categories when training. On average, they did less than 2 consecutive drawings of the same category before moving to another category. We observed that most participants included all their drawings in the training set.

Few participants discarded some of their drawings. Among the discarded drawings, most of them were examples of existing categories that might have helped participants to assess if the model had effectively learned previous representations. The remaining discarded drawings were off-category drawings that may have been occasionally used out of curiosity, as a way to challenge the algorithm without specific expectations on its outcome. Finally, we found that participants used different variations of the drawings for each categories, including variations in representations of the concept (for instance clocks and hourglasses to represent the "time" category), or transformations such as orientations, colors and shapes.

The workshop allowed us to collect rich data in a non-controlled experimental context. However, it also brought limited insights regarding our second research question that focuses on how the participants understand the system during a realistic teaching task. We decided to conduct an experimental study, using a think-aloud protocol in individual sessions, to further investigate the underlying choices and decisions behind the observed behavior and how participants became aware about ML-based systems.

5 USER STUDY: THINK-ALOUD INDIVIDUAL TEACHING SESSIONS

We conducted a remote think-aloud protocol with novices (in ML and CS) with the following objectives: **(1) identify novices' teaching strategies** of a sketch-based recognition algorithm; and **(2) investigate their understanding** of the machine behavior. The teaching task is similar to the pilot workshop, and consists in teaching a classification algorithm from scratch to recognise hand-made drawings in *Marcelle-sketch*.

5.1 Participants

We recruited 12 participants with limited to no knowledge in machine learning or computer science. We recruited participants by email among contacts of the association and from the students of the university, avoiding scientific or technological profiles. Among the 12 participants, 7 are female, and 5 are male. 7 participants are aged between 18 and 29, 1 between 30 and 39, 3 between 40 and 49 and 1 between 50 and 59. About their knowledge, participants graded their prior knowledge about image recognition systems in the pre-questionnaire. 6 participants answered that they are novices, 4 participants are "little informed", 1 participant is "informed" and 1 are knowledgeable.

5.2 Setup

We used an open-source video conferencing platform hosted on a secure server to communicate with the participants. The video conferencing platform can be accessed from the browser. We asked the participants to share their screen at the beginning of the session. We video-recorded their shared screen while they were training the model. We used the computer microphone to record the audio from the video-conference application. The participants performed the task on their own computer, using *Marcelle-sketch* in their browser. The application was linked to a server and a database in order to collect data, such as participant's drawings and models. From the version of *Marcelle-sketch* used in the pilot, we removed the possibility to change the color and the width of the pen. It was not often used by participants and it allowed us to reduce the variability and better compare the teaching strategies. The questionnaires were created with an open-source platform called Framaforms and shared with the participants through a link.

5.3 Procedure

When participants log in the video conferencing platform in the browser, the experimenter starts by explaining the general structure of the experiment. The participants are told that during the session they will have 30 minutes to train an image recognition algorithm to recognize drawings

that they will create, each drawing belonging to one of the predefined categories. Then, a link to the Marcelle-skecth application is sent to the participants. In the application, the third tab of the interface is a page where appears the link to the pre- and post-questionnaires. Participants are asked to fill the pre-questionnaire. The purpose of the first questionnaire is twofold. First, we want to inspect participants' knowledge about image recognition algorithms. Second, it serves as a primer to encourage them to think about how image recognition algorithm works. Once the pre-questionnaire has been filled, participants are asked to share their screen. The main teaching session comprises three steps:

(1) *Explanation of the task and interface.* The task is explained to the participants, which is to teach the algorithm to correctly classify drawings that they make with the mouse, into the pre-defined categories. We use the same categories as in the workshop: "Moon - Lune", "Hat - Chapeau", "Wave - Vague", "Cheese - Fromage" and "Time.- Temps". Then, we explain each component of the interface to the participants and we start recording the session.
(2) *Think-aloud teaching phase.* Participants have 30 minutes to teach the model. During this training phase, we ask them to think aloud. If the participant stops talking for few minutes, the experiment conductor reminds them to comment on what they are thinking about.
(3) *Think-aloud retrospection on the data.* After the teaching phase, there is a 10-minute phase to encourage the participants to reflect and debrief on the algorithm recognition abilities. Participants are asked to describe: 1) which drawings are correctly recognized by the algorithm and 2) which drawings the model is uncertain about. Similarly to the teaching session, participants are asked to comment their choices out-loud. The screen and audio recordings are stopped after this step.

The study ends with a post-questionnaire, which aims to evaluate how participants perceived the system and how participants' prior ideas about the behavior of an image recognition algorithm evolved after the interaction.

5.4 Data collection

We recorded the think-aloud sessions through screen recording. In addition, we collected the datasets made of the intermediate drawings (i.e. drawings after each strokes) and the training set (i.e. drawings used to train the system). We also collected the two datasets built after the training, which contain "recognized drawings" and "drawings the model is uncertain about". For those four datasets, we stored drawing as *png* images together with their creation timestamps, the predicted category when drawn (or assigned category when trained), the computed uncertainty, and the features from the MobileNet network. Finally, we collected the answers to the questionnaires stored on the Framaforms platform.

5.5 Data Analysis

5.5.1 Quantitative analysis of the teaching process. We computed a set of three measures related to the drawings performed by the participants to teach the model. These measures were motivated by our first Research Question on characterizing novices' teaching strategy. The measures are:

- *The amount of drawings trained* i.e. how many drawings were used to train the system. It relates to both the speed at which the participant draw and how often participants want to use a finished drawing to train the model.
- *The variability in the drawings.* We computed a measure of variability within a category using Euclidean distance between pairs of drawings in the feature space, i.e. the output vectors of MobileNet associated to each drawing. We averaged distances between all pairwise combinations of instances within a category (to avoid comparing images from different

category). We then averaged the variability across categories for each participant. Formally:

$$V_{participant} = \frac{1}{5} \sum_{c \in categories} \frac{1}{C^2_{size(c)}} \sum_{X_i, X_j \in c} d(M(X_i), M(X_j)) \quad (1)$$

with $C^2_{size(c)}$ the number of combinations of 2 instances in the category c, d the Euclidean distance, and $M(X)$ the feature vector after passing the input image X into the MobileNet network. To help the reader appreciate the variability across participants, Figure 5 in Appendix depicts the training set of the most variable and least variable participants.

- *The average number of consecutive inputs with the same category.* This measure highlights the sequencing i.e. the order in which participants trained the proposed categories. We display participants sequencing on the upper timelines on figure 6 in appendix 9.

5.5.2 Quantitative analysis of the model performance. We computed the following measures related to the performance of the trained classifier:

- *The generalization performance* measures how each trained model can generalize beyond a participant. We used the final trained model of each participant. We then computed an accuracy score on the test set composed of all the training sets from the 12 participants.
- *The personalization performance* measures how well the model can fit a participant's data provided during the training session. We also used the final trained model of each participant. We then computed an accuracy score on a test set composed of all finished drawings of the participant (that are used to train the model or not). We annotated by hand the finished images (images before the participant "clear"), discarding errors or involuntary strokes.

The performance scores are used as indicators on the model abilities rather than a quantification of the task completion. Participants were not asked to improve the generalization of their model when we introduced the task to them.

5.5.3 Qualitative analysis of the verbalizations. To analyse the verbal elicitation from the participants, we applied thematic analysis [6] to code and categorise the transcribed audio recordings. Two authors first labeled each meaningful verbalization, describing participant's actions or thoughts. From these labels, we created a set of themes that convey the participant's intent and address our research goals: (1) identify novice teaching strategies for an image recognition algorithm and (2) investigate novice understanding of the machine behavior. The theme created during this phase are:

- 5 themes about the participants' **learning behavior understanding**: *"interpretations and beliefs about the learning behavior of the* system", *"asking oneself about the learning behavior of the system"*, *"misunderstanding"*, *"the participant felt the system could learn a drawing successfully"*, *"the participant felt the system could not learn a drawing successfully"*;
- 2 themes about the participants' **teaching decision**: *"justification of an action according to previous ones"*, *"organisation and structure of the overall session"*;
- 2 themes about the participants' **teaching intentions** *"evaluation of previous images learned"* and *"exploration of new drawings"*.

In addition to the thematic analysis, we aligned the verbalisation to the drawings mentioned within them to have a better overview of the course of events and context. Then, the authors coded again the verbalizations according to these themes. The first author of the paper coded all participants' transcriptions and three co-investigators coded 4 different participants each. We gathered the codes and discussed their alignment. We categorized the 710 quotes from the 12 participants over the 9 themes mentioned above.

We finally kept the quotes where a clear agreement could be found between annotators, so approximately 350 quotes. The study was conducted in French, as well as the transcriptions and the analysis. The translation to English was only made to report the results. Note that the neutral pronoun is identical to the masculine pronoun in French. We then decided to keep the neutral pronoun every time the participant refers to the system.

6 RESULTS

In this section we report the findings resulting from (1) the qualitative and quantitative analysis of participant's teaching strategy, and from (2) the qualitative analysis of their understanding of the machine's learning behavior.

6.1 Analyzing teaching strategies

In this section, we present our findings related to the first research question on the identification of teaching strategies by novices and their relationship to model performance. The results in this section are primarily quantitative. They are complemented with quotes from the thematic analysis, which allows us to better describe participants' intentions about their strategy (when verbalized).

6.1.1 Novices adopt contrasting strategies. We analyzed the teaching strategies by looking at three measures informing on the teaching process: the amount of drawings trained; the variability infused in the inputs; and the adopted sequencing (see Section 5.5.1). Figure 2 depicts each participant within this teaching strategy space.

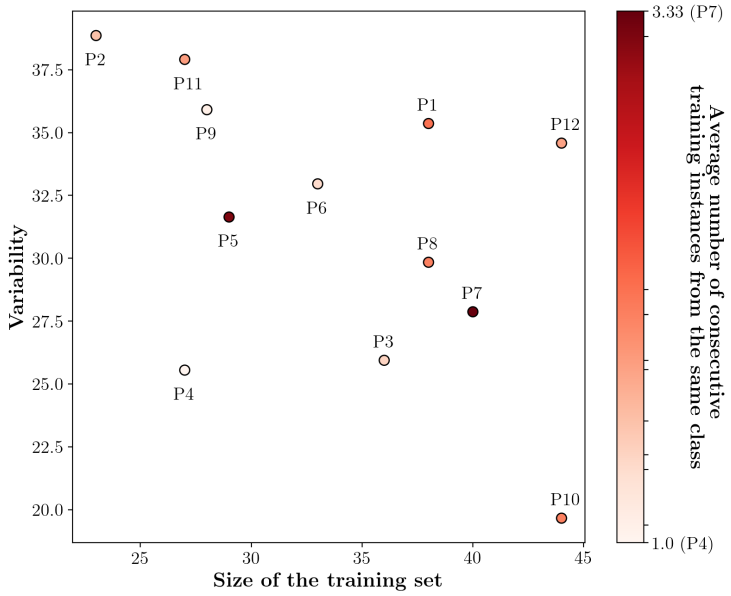


Fig. 2. Teaching strategy space: Variability (y axis) according to training set size (x axis) and sequencing (color map).

We first investigated whether these dimensions provide insights on complementary aspects of teaching strategies. We computed the correlations between these three dimensions and we found that there are not significant, meaning that each measure represents a dimension of the teaching strategies adopted by participants. In addition, we found that participants are well distributed in

the space. Within the 2-dimensional space created by the dimensions *variability* and *size of the training set*, the two extreme cases P2 and P10 suggest that low variability is more often related to simple shapes in the training set.

We also found that the participants adopted different teaching strategies, by analyzing how they sequenced the training instances. We found that the number of consecutive training drawings from the same category spans from 1 to 3.3 (see Figure 2). For instance, P4 never trained the same category with two consecutive drawings (leading to a consecutive rate of 1). On the contrary P5 and P7 have consistently drawn on average more than 3 drawings in a row from the same category. The average number of consecutive drawings from the same category is 1.9. This result highlights the spectrum of strategies from focusing on one category at a time (using several drawings) to constantly changing the training category. Importantly, participants generally did not explicitly state that they used a sequencing strategy.

- **Finding 1** Participants adopted heterogeneous teaching strategies in terms of training size, variability and sequencing, which underline the lack of means of the classifier on the actions to be taken to train it.

6.1.2 Impact of the variability on system performance. We consider two types of performance indicators: generalization performance and personalization performance (as described in Section 5.5.2). In this section, our goal is to link participants' strategies, described in the previous section, to these notions of system performance.

We investigated whether the dimensions of teaching strategies are linked to the system's performance. We found no correlations between these dimensions and the performance measures. However we found that data variability tends to be positively correlated to generalization performance ($R^2 = 0.31$, $p = 0.061$). This means that participants that infuse greater diversity in their drawings train a model that tends to better generalize across other participants' data. To a certain extent, this was expected since in ML, variability is known to be beneficial to generalizability. However, this rule can also be mitigated by the fact that idiosyncratic variability could degrade the performance because fewer correlations within the data can be found.

Interestingly, as a counter-example, P5 created a dataset with low variability and reached higher generalization score than P9 who created a dataset with high variability. Figure 3 depicts examples from the data provided by these two participants. It shows that P5 favoured simplistic, icon-style, representations while P9 opted for more complex and idiosyncratic representations. Therefore, *variability*, as considered in this work, does not systematically imply good generalization score. These results suggest that the nature of this variability is critical.

We found that participant 12 is the participant that obtained the best scores in both performance indicators. P12 obtained the best generalization score (accuracy equals to 0.40) and the second best personalization score (accuracy equals to 0.82). P12 has the largest training set and one of the highest data variability. P12 managed to create well separated categories that may be shared across participants. P12 also gradually increased the difficulty of the inputs curated. As a matter of fact, P12's verbalizations in the theme *"organization and structure of the overall session"* give us information about the dynamic of her teaching strategies. She elicited a precise training policy early in the session to avoid adding similar instances if they are already confidently recognized. She then updated her decision "threshold" based on the subjective quality of the drawing: *«If it's still confident even if I make an ugly drawing, I want to start training it to be very confident with my ugly drawings. That's going to be my new policy because I see that it can be confident with my ugly drawings»*. This process operates as a *curriculum* for the recogniser.

- **Finding 2**: Variability tends to favour the generalisation of the model, while the other dimensions of the teaching strategy do not seem to affect the performance of the system.

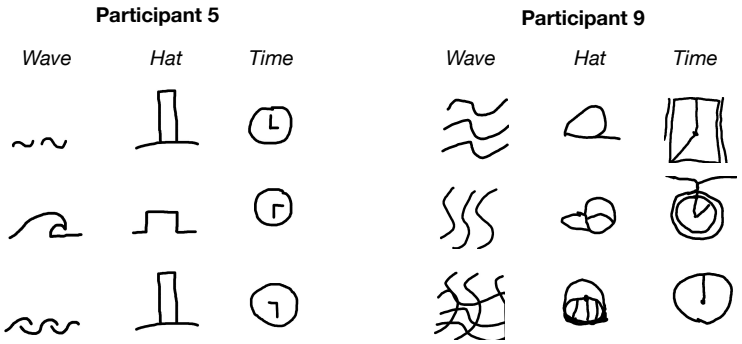


Fig. 3. Samples of the training set of P5 and P9. P5 adopted a more icon-style whereas P9 opted for more idiosyncratic drawings.

The type of the variability, and the fact it might be introduced progressively, plays a role in building an efficient classifier that can handle various representations.

6.1.3 Sequencing affects the model performance and performance perception. The sequencing (i.e. average number of consecutive instance trained with the same category) is not correlated with generalization or personalization performances. However, we found that the first drawings used to train the system are critical to ensure a good performance. Participants who focused on a single category in the beginning of the session created a model that predominantly predicted this category over rest of the session. This phenomenon is due to the incremental nature of the training procedure involved in the system. The model is optimizing its parameters according to limited data that are drawn from a single category. The loss function can then remain locked into a local minimum, blocking the network parameters. The model then require multiple iterations on new instances from other categories to escape from this local minimum and to reach a better optimum.

Figure 4 depicts the training sequencing for participants 7, 1 and 8. For each participant, the top line represents the training sequencing (each instance from the beginning to the end of the training, and its label) while the bottom line represents the predictions. These participants are the ones who trained at least 4 images from the same category at the very beginning of the session. We can see that the consecutive predictions remain the same as the first category. P1 succeeded in cancelling this effect at about 37% of the session, by providing a balanced number of instances to other categories. The effect remains for P8 and progressively disappears between 33 and 60% of the session. The effect seems to persist for P7 until the end of the session. This might be due to the fact that P7 trained the highest number of consecutive instance from the same category at the very beginning of the session (9 consecutive "Moon"). The same visualization for other participants can be seen in figure 6 in Appendix 9. We can see that this effect also affected participant 2.

From the verbalizations related to the themes "misunderstanding" and "the participant felt the system could not learn a drawing successfully", we notice that P7, P1 and P8 perceived this inertia effect, while not necessarily understanding it. Only P1 seems to adopt appropriate actions. Indeed, P7 mentioned two times that "it really likes moon", while P8 and P1 refer to this effect multiple times: *«But why it still thinks it's a wave there, I don't understand.»* (P8), and: *«I change the category because it always refers me to the Moon»* (P1).

- **Finding 3**. The training sequencing (i.e. the order in which examples are given) has an important role in incremental teaching, especially at the very beginning of the teaching. The actions necessary to unlock confusing model behaviors are not transparent.

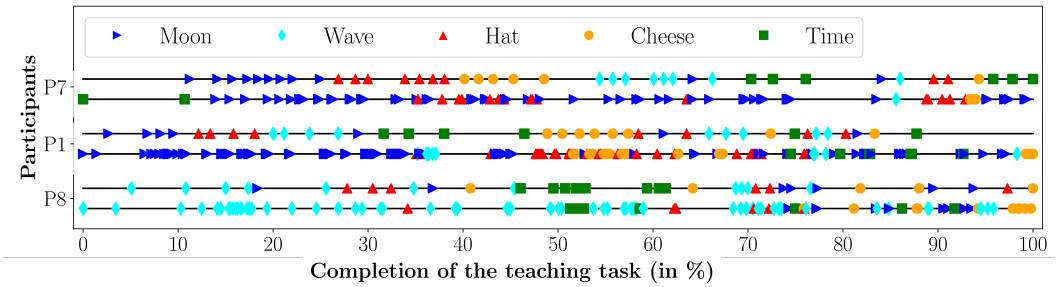


Fig. 4. Categories trained (upper timeline) and predicted (lower timeline) in chronological order for participants 7,1 and 8. These participants trained at least 4 consecutive images with the same class at the beginning and their predictions are affected for the rest of the session. The x axis represent the completion of the task (in %).

6.2 Understanding the machine's learning behavior

We now present the results relating to novices' understanding of the system's learning behavior. This section reports qualitative results drawn from the thematic analysis (findings 4 and 6) and the questionnaire (finding 5). The quantitative data (images drawn) were only used during the analysis to give a broader picture of the context.

6.2.1 Participants investigate and teach input feature variations. We found that some participants became aware of the features that the system takes into account in the recognition process. In a first preliminary analysis, we categorized verbalization in which participants mentioned the use of variability. They are gathered in the theme "exploration of new drawings" mentioned in section 5.5.3. When we categorized the quotes in the theme "exploration of new drawings", we noticed the occurrences of geometric vocabulary (rotations, size changes) and decided to group these explorations as "operations". This group includes the reuse of the same representation for geometrical transformations or duplications. The group "executions" stood out from the rest since two different gestures could lead to the same representations. Finally, the "representations" group encompasses all remaining extracted labels. The drawings in this group are all characterized by changes in the composition of the drawing i.e. drawings made with a different organization of the strokes with respect to each other. From this categorization, we built a taxonomy of the different input features that the participants mentioned when introducing variability. This taxonomy is summarized in Table 1. As we mention above, we identified three groups: 1) the *representation* of an image such as the shape, the infilling, the relief (plane or depth), and context (adding contextual details on the image); then 2) the *execution* of the drawing, such as the gesture used to draw; and finally 3) the *operations* on an image such as translation, rotation, duplication (drawing several representation on one image), or change in size. In Table 1 we also report participants who used these features and an example from their verbalization. Participants in bold in the table intentionally conducted investigations to understand how the system could handle this feature, while underlined participants are the ones who came up with conclusions from their investigations.

From this analysis, we found that participants created new insights on the model mostly when investigating *operations* and *execution*. We assume *representation* features are harder to isolate in order to conduct investigations. For instance, changing the context (adding related representations on the drawing) also affects the general shape of the drawing. Conversely, *execution* and *operations* can easily be isolated and tested on learned representations.

Feature group	Feature	Part. ID	Quote example
Representations	Shape	P6, P7, P9, P10, P11	"Ok so I think it's pretty much all learned now, mostly based on the shapes" (P11)
	Infilling	**P2**, **P6**, P7, **P11**	"I was wondering [...] if it's only the structure that I draw, if it would be detected as a moon even with the color with all the details" (P11)
	Relief	P9	"I think that's a key thing like knowing the difference between 2D ans 3D." (P9)
	Context	**P11**	"Now I'm trying to add more other details rather than just "vague" [...] to see if the machine can still detect the main subject of this painting." (P11)
Execution	Gesture	P6, **P10**	"I thought it was recording the final image, but it's possible that it records every movement I make." (P6)
Operations	Translation	**P8**, **P10**	"Maybe the position didn't change anything. I'm going to put the cheese in a different corner." (P10)
	Rotation	**P7**, **P8**, P12	"First I will try to see if my theory is confirmed, that there is no direction" (P8)
	Duplication	P9	"I tried different methods such as doubling the amount, maybe even tripling, quadrupling, so many many more" (P9)
	Size	**P8**, **P10**	"Does size matter? [...] I do a little clock test depending on the size and it doesn't work at all." (P8)

Table 1. Input features presumed to be considered in the learning process. Participants that investigated their hypothesis with further inputs are indicated in bold.

Intentional investigations on the representations were mainly made with "infilling" i.e. participants investigated what happened when they changed inner details, such as texture and color. P6 and P11 both concluded that the color did not affect the prediction only after drawing one or two new colored images that were correctly recognized. They did not perform extensive analysis of this feature and conclude with a partially false claim about the importance of the infilling regarding the shape.

Regarding *execution* and *operations*, 5 out of the 7 participants that conducted investigations generated insights that were in line with the design of the system. Regarding execution, P10 did two identical images but inverting the direction in which she made the strokes. Based on these tries, she concluded that only the final image is taken into account. P7 investigated rotations and found that the uncertainty decreased when tilting a learned representation: *«When I flipped the hat 90 degrees, it became uncertain. Maybe I didn't noticed that for the moon.»* (P8). Another example is P10 who drew each category with a regular size, and then drew a "cheese" in the bottom right corner (cf figure 5b). P10 trained the model with the transformed representation twice. After placing a cheese in the top left-hand corner, P10 became aware about the translation: *«It still thinks it's a cheese [...] when it's not at all in the same corner of the picture, so it must not be the position in the picture»*. Here participant 10 understood that the model is invariant to translations (i.e. position). Then she did the same operation with other categories but the system kept predicting "cheese". P10 concluded that: *« I first showed the system that cheese could be in different corners, so it understood for the cheese. When I do other things in other corners it still thinks it's a cheese »*. This illustrates how participants who actively investigate operations may build a more precise mental model about the underlying algorithm and the features it takes into account.

- **Finding 4**: Participants verbalized various features that the system might take into account in the learning, and they tend to discover insights about the system inner working when investigating 'execution' and 'operations'.

6.2.2 Participants understood the order in which the examples were given affects the training. In the pre- and post-questionnaires, we asked the following question: "According to you, how important do you think the following criteria are for learning the algorithm?". We provided a list of criteria that participants had to annotate on a 5-point Likert scale (from "not important at all", to "very important"). Using pairwise t-tests, we found that the importance attributed to "the order in which examples are given" significantly increased after the teaching session ($p = 0.011$).

P8 and P12 explicitly expressed doubts about the importance of order during the session: *«Yes, so you'll notice that I didn't take the time to sit down and think [...] without thinking about whether the order in which I draw will have an impact on the algorithm in fine.»* (P8). P5 and P2 became aware about order regarding the wrong predictions following the category they trained: *«If I had started by drawing rectangle-shaped cheeses before the hats, it would have recognized the cheeses well. So it's not that it's badly recognized but it's because I did it in that order!»* (P5). *«Oh yes, so everything is a wave. From now on, everything is a wave».* (P2). It is worth mentioning that P2 and P5 have a high number of consecutive examples from the same class, meaning that they mostly focused on training one class after the other. The design choice (incremental teaching) and the phenomenon described in finding 3 (model locked on certain predictions) are probably responsible for the participants' reconsideration of the order effect after the experiment. Participants did not anticipate this effect, which suggests that they were expecting a more intuitive learning behavior from the machine, possibly closer to human learning. P7 said: *«This is the big difference between the machine and humans because we are intuitive. The machine will never have an intuition».* This result shows the need to help novice users to consider order in the interaction, by helping them building meaningful curriculum in the teaching (discussed in the implications for design section 7).

- **Finding 5**: Participants became aware of the importance of order in which drawings are provided, which may characterize incremental teaching.

6.2.3 Underlying neural network properties are confusing for novices. In this section, we studied all the quotes where participants asked themselves questions about the system's behavior or expressed a lack of understanding. The quotes are gathered in the themes *"asking oneself about the learning behavior of the system"* and *"misunderstanding"* introduced in section 5.5.3. We categorized them according to the source of the confusion. If the majority of the confusions are due to unexpected predictions, 29% of them stemmed from properties of Neural Networks. From these confusions, we built a taxonomy reported in Table 2.

The taxonomy is composed of four properties. *Exclusivity* is the fact that each input is associated with a unique output both during training and prediction. The network cannot predict that a drawing belongs to two different categories at the same time. This property was discussed by P2 and P6 since they drew ambiguous images expecting that the system would predict two categories. *Pre-existence of categories* stems from the initialization of the network with pre-defined output size (number of categories). Thus, P2 and P5 were surprised that the model could predict a category for which no image was yet provided. *Optimization inertia* is the fact that the model is not building immediate rules from participants' demonstrations but it optimizes parameters towards an optimum. Thus, P1, P2 and P3 were surprised that the model could still be wrong on the same image after having being trained on that image. Finally, *Prior knowledge* is the fact that the model embeds prior knowledge or not. P5 wondered if the algorithm was trained with other participants' drawings beforehand: *« It's strange, you still get the impression that others have provided images. I feel like I'm*

System property	Participants	Quote example
Exclusivity	P2, P6	"Is it possible for a drawing to be well recognised in both one category and another, and is it true?" (P6)
Pre-existence of categories	P2, P5	"I am very surprised, because I don't understand why it makes me a proposition when it has never seen a hat or anything else." (P2)
Optimization inertia	P1, P2, P3,	"It predicted a hat with a low confidence, and I told it "yes it is a hat", and it didn't say "ah well ok, I'm confident because you told me.""
Prior knowledge	P5, P8	"It's weird, you still get the impression that others have provided images. I feel like I'm not the first." (P5)

Table 2. Properties of Neural Network perceived as confusing for novices along the teaching session.

not the first. » (P5). P5 then changed her mind when the model failed on categories she had not trained yet.

P8 first believed that the algorithm relied on rules that the system designer chose. The idea of a rule-based system was primed in the questionnaire. P8 stated that *« it would be easier to provide rules rather than drawing over and over. »*. Later, P8 tried to identify the nature of these rules: *« it was part of your rules that if there's some kind of vague line, it's a wave »*. She finally intuited a notion of optimisation with the idea that the rules could be adaptable to the data: *« I think it's the one that may have ... not the fewest rules, but the rules that get the more easily adapted»* (P8).

- **Finding 6**: 29% of the confusions expressed by the participants originate from 4 properties of neural network inherent mechanisms that we identified.

7 IMPLICATIONS FOR DESIGN

We have shown how novices train a supervised machine learning algorithm able to recognise drawings, and how they elicited their understandings about the system's behavior. In this section, we draw implications for designing IML systems intended to neophytes in ML. Such systems may have an important impact on ML education, creative applications or expert domains (outside of ML and CS) involving ML algorithms.

[I1] Provide guidance for building teaching curricula

A teaching curriculum is a strategy to organize the training examples in a meaningful way, that gradually introduces complexity. We showed that curricula have a critical role in interactive machine teaching with incremental inputs, in particular regarding the order in which examples are given. More precisely, we found that NN-based algorithms can be blocked if too many instances from one category are provided at an early stage of the training process, which might confuse the user (finding 3). In addition we saw that the participant reaching the best performance had an explicit curriculum strategy that she elicited and applied during the teaching session (finding 2). This finding is supported by the Machine Learning literature, although rather scarcely explored, which shows that simulating a curriculum improves the performance [5]. Therefore, we propose to:

- Encourage users to diversify the categories used at an early stage of the training process;
- Encourage users to consider a strategy in training that presents a progression in (objective or subjective) difficulty.

[I2] Allow modification of past teaching actions and sequences of actions

Providing feedback on participant's previous teaching actions (e.g. sequence of drawings) and allowing for modification of past actions could further improve the personalization and the model ability to handle variability. For example, we found that novices noticed an order effect without being able to correct it (finding 5): *« I do a little with spontaneity, without thinking if the order in which I do the drawings will have an impact on the algorithm in fine »*. This could be implemented with visualization mechanisms reflecting on the passed teaching curriculum, allowing users to change the sequence of the passed examples and retrain the model with the new curriculum. Dataset history has recently been shown useful for ML developers in real-world scenarios [21]. We suggest that curriculum history should go one step further by actively helping practitioners to build better models incrementally and get a better understanding of these models by structuring the teaching into levels of difficulty. Therefore, complementary to curriculum guidance, we suggest designers to:

- Integrate retrospective feedback on the curriculum and provide interaction techniques allowing users to take actions on it.

[I3] Assist supervised data augmentation in incremental machine teaching

We showed that several participants investigated geometrical operations (rotations, translations, changes in size). Although these inputs served as a way to understand the system's robustness, the resulting transformations were also added to the training set (finding 4). Regarding this strategy, participant 10 said: *«But if you want to vary in size or something like that, if you want to change an attribute, you have to add more data so that it understands that too, but I think it's possible.»*. This result, in addition to the tendency that higher variability results in better generalization performance (finding 2), pushes toward the following design implications:

- Provide tools to easily transform existing inputs;
- Leave the choice of the type and the amplitude of the transformations to the user to foster investigative behaviors.

[I4] Show optimization inertia and model's state changes

Beside exclusivity, most participants were confused about the fact that the system remained wrong right after training the model on a new example (finding 6). This is due to the fact that such a classifier is an optimizer that does not consider the training set as ground truth but as a support to update its parameters. A way to convey the idea that the model relies on states and not rules is to visualize updates. This could be done by showing changes of predictions and uncertainty not only on the current drawing but across all previous inputs from the data set. It could also support users' willingness to observe the evolution of uncertainty as they add similar representations as expressed by participant 11: *«So I cannot compare the clock and the watch, like how it's detected?» (P11).*

We suggest designers to:

- Show optimization updates by showing the evolution of predictions and uncertainties across the dataset.

8 DISCUSSION

The use of *Marcelle-Sketch* in a realistic teaching task gave us insights on novices' teaching strategies, and how these are related to data sequencing and variability. In addition, we learned about novices (mis)understandings regarding the system's underlying learning mechanisms and how they change during the teaching session. In the previous section we proposed four implications for the design of IML systems dedicated to the general public. Here we further discuss the interaction workflow

enabled by the proposed system, the notion of curriculum and the socio-cultural implications of this research.

8.1 Workflow and novice users' teaching strategies

The workflow implemented in our application differs from typical IML interaction scenarios in which users are iterating between testing and model updates [3]. Our scenario involves incremental learning, as opposed to batch-based learning popular in several IML tools such as Teachable Machine. Therefore, model evaluation arises both from the ability to create data and get immediate predictions about them. This process enabled intertwined model testing and updates. In other words, the typical distinction between training and testing is blurred. For instance, we found that participants that investigated geometrical operations also taught the system with the transformations created. The tight coupling between exploration and training allowed participants to explore input variability as a way to both 1) challenge the algorithm with ambiguous examples and 2) extend the generality of the taught concepts. Hong et al. [22] also accounted for the use of variability in training data in their teaching scenario. However, their task involved separate training and testing phases, encouraging fixed teaching strategies (also highlighted in [34]). Surprisingly, the authors noticed that testing examples were less variable than training examples. We can assume that direct feedback and incremental training influences data variability since it arouses the curiosity of the participants to know the evolution of the prediction and uncertainty as they create the drawing (real time updates after each strokes).

8.2 Sequencing and teaching curriculum

Also stemming from the incremental form of training involved in our scenario, our results suggest that the sequencing in the early part of the teaching is crucial to avoid the system to be blocked and untrainable. We proposed in the previous section that designers should include guidance for meaningful curricula. This would improve both the robustness of the system and the understanding of novices. Here we further discuss this idea of curriculum and the type of guidance that a user could received to build it.

Cakmak and Thomaz [7] discussed three types of teaching guidance that apply only in problems with an known optimal curriculum, with instances that can be characterized as a set of categorical features or with a finite set of possible examples. Our scenario does not meet these criteria so the teaching guidance proposed by Cakmak and Thomaz [7] are difficult to apply. However, teaching guidance can be designed from 1) a user-centered approach or 2) techniques borrowed from the machine learning literature. As an example of user-center approach, Wall et al. [45] designed guidance notifications from observations of experts teaching patterns in a task of article classification. The kind of teaching patterns they found significantly differs from ours since the task requires the exploration of a large database of articles (as opposed to a limited number of user-generated inputs). User-centered design of teaching guidance must consider the use-case specificity. Wall et al. [45] also emphasize on the importance of the timing at which notifications are shown. On the other hand, computational methods can also support the design of a meaningful teaching guidance to help sequence the inputs. For instance, a technique called Active Class Selection [31] focuses on calculating the most beneficial class in which the one should add data according to the model parameters. These techniques should be investigated in the context of expressive machine teaching, such as presented in this paper.

8.3 Socio-cultural implications: democratization and appropriation of ML technologies

The application *Marcelle-Sketch* that we developed as a technology probe in the context of this work is related to the well-known *quickdraw*[3] application designed and developed at Google. Quickdraw is also presented as a teaching application: people using quickdraw contribute to building a large dataset of doodling data. In the Quickdraw application, a participant is asked to draw a given category (for instance a *drill*, or a *house*) in under 20 seconds. Once the application has guessed the drawing, it stops the session and goes on to the next category. The two applications are similar, but their differences are insightful. In our application, user agency is key. Giving users agency by actively involving them in the training process highlighted teaching strategies, beliefs and understanding of the technology. In quickdraw, users are not aware about the impact that their drawings will have on the system. In addition, quickdraw's design tends to lead to normative drawing behaviors, which have been highlighted through data analysis and visualization. For instance, averaging the cat-related drawings lead to very similar result across cultures [24]. Averaging the chair-related drawings shows differences across cultures [32], but remains impressively consistent considering the number of input drawings. On the contrary, our application promotes personalization and curation.

In our work, machine teaching takes the strand of training a machine to perform a task that is guided by the end users needs and will. End users were not asked to train a generic sketch recognition model, but a sort of teammate in a drawing recognition game. The quality of the model is assessed through subjective and qualitative criteria, similarly to the way artists assess IML systems in practice [15]. We see in this approach of machine teaching an interesting means for research in ML democratization and education. As a matter of fact, the work presented in this paper has been initiated through a collaboration with the association *Traces*, dedicated to science popularization. Our collaborators from the association saw the idea of teaching a machine as a means to give to people a tool to learn about machine learning, reflect about it and democratize it. This idea is gaining a very recent interest in the field of HCI and CSCW [12, 20, 28, 52]. Our work is in line with this work, promoting learning, appropriation and decentralized governance of technology and extends it by allowing novice users to be engaged with the expressive capacities of modern ML (deep learning), which means the possibility to convey increasingly rich concepts through data.

9 CONCLUSION AND FUTURE WORK

We explored the way people could teach learning algorithms, what strategy they use to "make it work" and what they understand from their behavior. To do so, we studied how novice users use *Marcelle-Sketch*, a sketch recognition application, designed to be incrementally teachable and usable in a web browser. The application has original ML features allowing for rapid and robust training. This application has been used in both a general public online pilot workshop and individual think-aloud sessions with novice users in ML and CS.

We found that participants adopted heterogeneous teaching strategies regarding sequencing and variability. The variability tends to favour the model generalisation abilities but the type of variability, and the fact it might be introduced progressively, plays a role in building an efficient classifier. We also found that a repetitive sequencing at the very beginning of the teaching can be detrimental to future predictions. Regarding users' understanding, we found that participants discovered new insights on the system by investigating transformations on representations. They also became aware about the importance of sequencing. Then, participants' confusions originate

[3]https://quickdraw.withgoogle.com

from 4 inherent properties of Neural Network that we discuss. From these results, we proposed implications for design of Interactive Machine Learning systems and discuss the specificity of our workflow. Finally, we discussed socio-cultural implications of teachable systems on democratization and appropriation of Machine Learning technologies.

Future research will focus on how novice strategies and understandings are developing on the longer term since appropriation and learning are mechanisms that require time. We think that more research needs to be conducted on the role of machine teaching for ML education and democratization. We believe our approach contributes to this endeavour, and more generally promotes a Human-centred approach of Machine Learning and Artificial Intelligence.

ACKNOWLEDGMENTS

This research was supported by the ELEMENT project (ANR-18- CE33-0002) from the French National Research Agency and the CNRS-funded project INTACT under the PEPS programme. We want to acknowledge and thank everyone who was involved in each stage of the research, in particular the anonymous reviewers and the participants of the study. We especially want to express our sincere gratitude to Matteo Merzagora, Aude Ghilbert, Paul Boniface and Arnaud Malher from the association TRACES and the Projet Siscode (Horizon 2020 Research and Innovation programme, grant agreement No 788217), whose collaboration made this study possible. Thanks to Gianni Franchi for his useful thoughts on uncertainty in Deep Neural Networks.

REFERENCES

[1] Adam Agassi, Iddo Yehoshua Wald, Hadas Erel, and Oren Zuckerman. 2019. Scratch nodes ML: A playful system for children to create gesture recognition classifiers. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312894

[2] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (12 2014), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

[3] Saleema Amershi, Maya Cakmak, W. Bradley Knox, and Todd Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* (2014). https://doi.org/10.1609/aimag.v35i4.2513

[4] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. 2018. The Power of Ensembles for Active Learning in Image Classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2018.00976

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. https://doi.org/10.1145/1553374.1553380

[6] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[7] Maya Cakmak and Andrea L. Thomaz. 2014. Eliciting good teaching from humans for machine learners. *Artificial Intelligence* 217 (12 2014), 198–215. https://doi.org/10.1016/j.artint.2014.08.005

[8] Michelle Carney, Barron Webster, and Jonas Jongejan. 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*. Association for Computing Machinery, 9. https://doi.org/10.1145/3334480.3382839

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[10] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. 2017. No Title. https://doi.org/10/cvvd

[11] Jerry Alan Fails and Dan R. Olsen. [n. d.]. Interactive machine learning. ACM Press, New York, New York, USA, 39. https://doi.org/10.1145/604045.604056

[12] Rebecca Fiebrink. 2019. Machine learning education for artists, musicians, and other creative practitioners. *ACM Transactions on Computing Education (TOCE)* 19, 4 (2019), 1–32.

[13] Rebecca Fiebrink and Baptiste Caramiaux. 2018. *The machine learning algorithm as creative musical tool*. Oxford University Press.

[14] Rebecca Fiebrink, Perry R Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 147. https://doi.org/10.1145/1978942.1978965

[15] Rebecca Fiebrink, Perry R Cook, and Dan Trueman. 2011. Human model evaluation in interactive supervised learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 147–156.

[16] Jules Françoise, Jules Françoise, Gesture-sound Mapping, Interactive Music, Systems Proceedings, and Jules Françoise. 2014. Gesture-Sound Mapping by Demonstration in Interactive Music Systems To cite this version : HAL Id : hal-01061221 Gesture – Sound Mapping by Demonstration in Interactive Music Systems. *Proceedings of the 21st ACM international conference on Multimedia* (2014), 1051–1054. https://doi.org/10.1145/2502081.2502214

[17] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, Nicolas D'alessandro, Joëlle Tilmanne, Todd Kulesza, and Baptiste Caramiaux. 2016. Human-centered machine learning. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 07-12-May-2016. Association for Computing Machinery, New York, New York, USA, 3558–3565. https://doi.org/10.1145/2851581.2856492

[18] Marco Gillies, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, Bongshin Lee, and others. 2016. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 3558–3565.

[19] David Ha and Douglas Eck. 2017. A Neural Representation of Sketch Drawings. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (4 2017). http://arxiv.org/abs/1704.03477

[20] 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.

[21] Fred Hohman, Kanit Wongsuphasawat, Mary Beth Kery, and Kayur Patel. 2020. Understanding and Visualizing Data Iteration in Machine Learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376177

[22] Jonggi Hong, Kyungjun Lee, June Xu, and Hernisa Kacorri. 2020. Crowdsourcing the Perception of Machine Teaching. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376428

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets. *arXiv preprint arXiv:1704.04861* (2017). https://doi.org/arXiv:1704.04861

[24] Ian Johnson. Sep 29, 2018. *Machine Learning for Visualization*. https://medium.com/@enjalot/machine-learning-for-visualization-927a9dff1cab

[25] Michael I Jordan and Tom M Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. *Science* 349, 6245 (2015), 255–260.

[26] Todd Kulesza, Margaret Burnett, Weng Keen Wong, and Simone Stumpf. 2015. Principles of Explanatory Debugging to personalize interactive machine learning. In *International Conference on Intelligent User Interfaces, Proceedings IUI*, Vol. 2015-January. Association for Computing Machinery, New York, New York, USA, 126–137. https://doi.org/10.1145/2678025.2701399

[27] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*. 6402–6413.

[28] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. 2019. WeBuildAI: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–35.

[29] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. 2017. *Iterative Machine Teaching*. Technical Report. https://dl.acm.org/citation.cfm?id=3305903

[30] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James M. Rehg, and Le Song. 2017. Towards Black-box Iterative Machine Teaching. *35th International Conference on Machine Learning, ICML 2018* 7 (10 2017), 4911–4928. http://arxiv.org/abs/1710.07742

[31] R. Lomasky, C. E. Brodley, M. Aernecke, D. Walt, and M. Friedl. 2007. Active class selection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4701 LNAI. Springer Verlag, 640–647. https://doi.org/10.1007/978-3-540-74958-5{_}63

[32] Kyle McDonald. Aug 28, 2017. *Looking at quickdraw data*. https://twitter.com/kcimc/status/902229612666658816?lang=en

[33] Dan Morris and Rebecca Fiebrink. 2013. Using machine learning to support pedagogy in the arts. *Personal and Ubiquitous Computing* 17, 8 (12 2013), 1631–1635. https://doi.org/10.1007/s00779-012-0526-1

[34] Changhoon Oh, Seonghyeon Kim, Jinhan Choi, Jinsu Eun, Soomin Kim, Juho Kim, Joonhwan Lee, and Bongwon Suh. 2020. Understanding How People Reason about Aesthetic Evaluations of Artificial Intelligence. *dl.acm.org* (7 2020), 1169–1181. https://doi.org/10.1145/3357236.3395430

[35] Changhoon Oh, Jungwoo Song, Jinhan Choi, Seonghyeon Kim, Sungwoo Lee, and Bongwon Suh. 2018. I lead, you help but only with enough details: Understanding user experience of co-creation with artificial intelligence. In *Proceedings*

of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 649.

[36] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. 2019. Machine behaviour. *Nature* 568, 7753 (2019), 477–486.

[37] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: Programming for all. *Commun. ACM* 52, 11 (11 2009), 60–67. https://doi.org/10.1145/1592761.1592779

[38] Tyler Scott, Karl Ridgeway, and Michael C Mozer. 2018. Adapted deep embeddings: A synthesis of methods for k-shot inductive transfer learning. In *Advances in Neural Information Processing Systems*. 76–85.

[39] Ayumi Shinohara and Satoru Miyano. 1991. Teachability in computational learning. *New Generation Computing* 8, 4 (12 1991), 337–347. https://doi.org/10.1007/BF03037091

[40] Patrice Y. Simard, Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. Machine Teaching: A New Paradigm for Building Machine Learning Systems. *arXiv preprint arXiv:1707.06742* (2017). http://arxiv.org/abs/1707.06742

[41] Patrice Y. Simard, Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. 2017. Machine Teaching: A New Paradigm for Building Machine Learning Systems. (7 2017). http://arxiv.org/abs/1707.06742

[42] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies* 67, 8 (2009), 639–662.

[43] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. 2009. EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1283–1292.

[44] Andrea L. Thomaz and Cynthia Breazeal. 2008. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence* 172, 6-7 (4 2008), 716–737. https://doi.org/10.1016/j.artint.2007.09.009

[45] Emily Wall, Soroush Ghorashi, and Gonzalo Ramos. 2019. Using Expert Patterns in Assisted Interactive Machine Learning: A Study in Machine Teaching. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11748 LNCS. Springer Verlag, 578–599. https://doi.org/10.1007/978-3-030-29387-1{_}34

[46] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. 2001. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies* 55, 3 (2001), 281–292.

[47] Christine T. Wolf, Haiyi Zhu, Julia Bullard, Min Kyung Lee, and Jed R. Brubaker. 2018. The Changing Contours of "Participation" in Data-Driven, Algorithmic Ecosystems: Challenges, Tactics, and an Agenda. In *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '18)*. Association for Computing Machinery, New York, NY, USA, 377–384. https://doi.org/10.1145/3272973.3273005

[48] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating how experienced UX designers effectively work with machine learning. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 585–596.

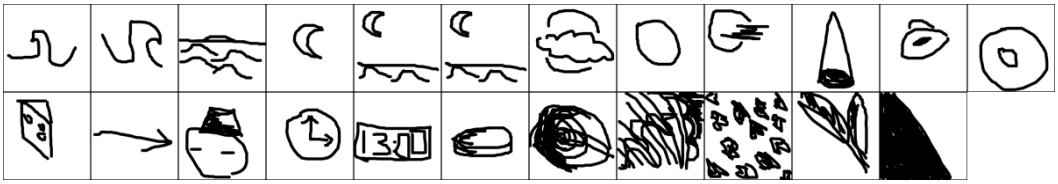
[49] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[50] Qian Yang, Jina Suh, Nan-Chen Chen, and Gonzalo Ramos. 2018. Grounding interactive machine learning tool design in how non-experts actually build models. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 573–584.

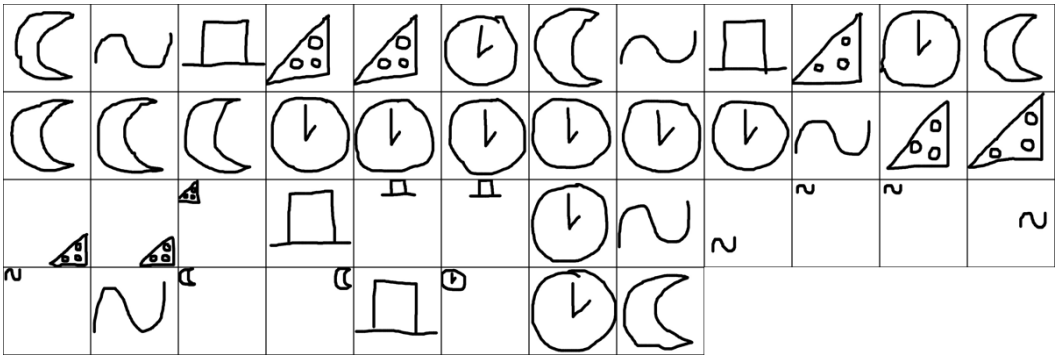
[51] Xiaojin Zhu. 2015. *Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education A(D) θ **. Technical Report. www.aaai.org

[52] 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.

APPENDIX A: EXAMPLE OF CONTRASTING TRAINING SETS



(a) Training set of participant 2



(b) Training set of participant 10

Fig. 5. Most variable (a) and least variable (b) training set among participants.

APPENDIX B: PARTICIPANT SEQUENCING AND PREDICTIONS

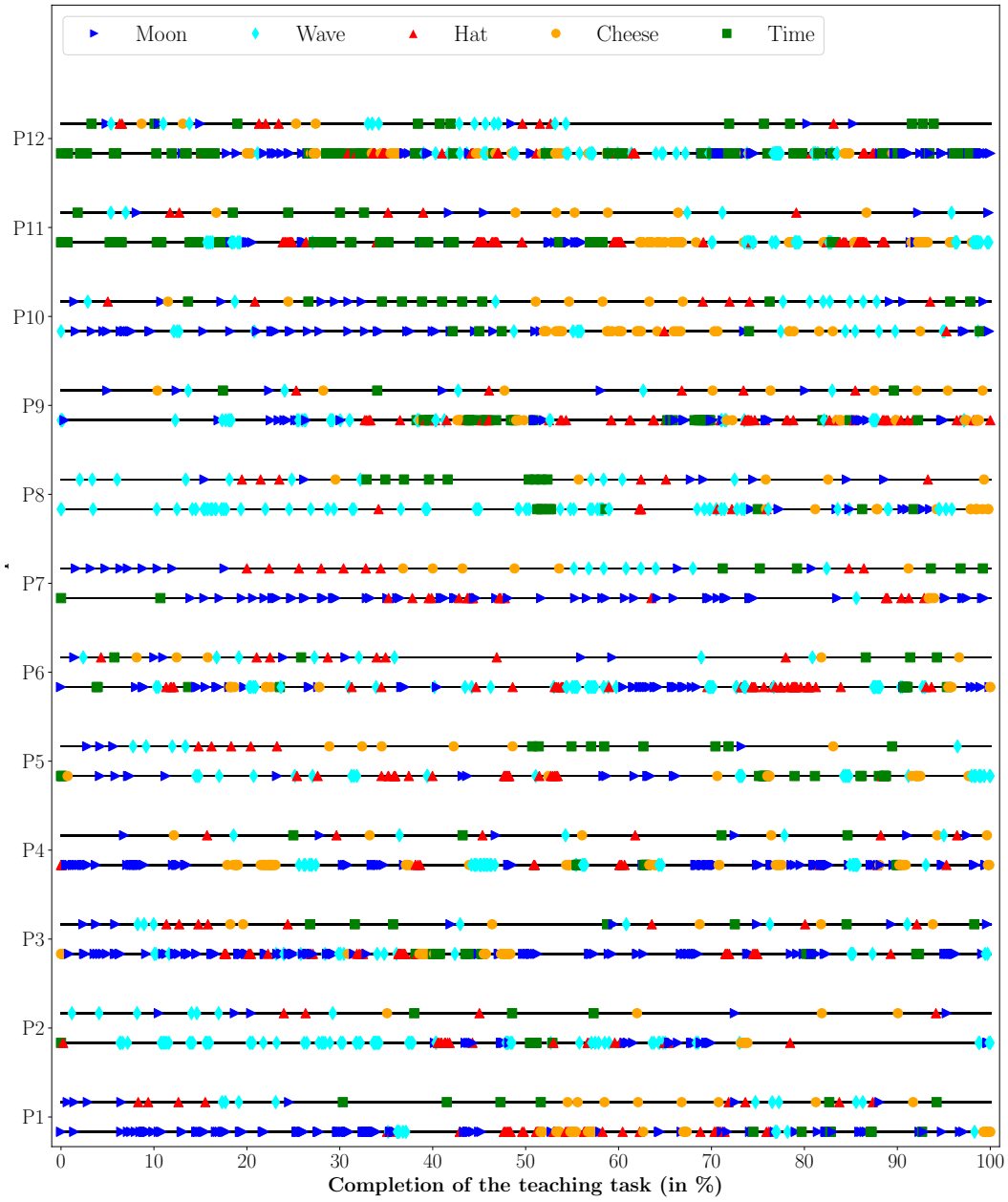


Fig. 6. Categories trained (upper timeline) and predicted (lower timeline) in chronological order for each participants according to the completion of the task (in %).

Received June 2020; revised October 2020; accepted December 2020