6 Intraversions: Human–Technology Relations in Flux

HC systems, just like any sociotechnical system in everyday life, and despite the linear visions behind them and driving them, are constantly in a state of becoming, that is to say, they are "ontogenetic in nature" (Kitchin 2016, 18). Drawing on insights from his collaborative work with human geographer Martin Dodge (2011), geographer Rob Kitchin aptly describes this state of becoming for algorithms that are "teased into being: edited, revised, deleted and restarted, shared with others, passing through multiple iterations stretched out over time and space" (2016, 18). To some extent, they are "always uncertain, provisional and messy fragile accomplishments" (Gillespie 2014; Neyland 2015; both cited in Kitchin 2016, 18). In the previous chapter, I discussed Stall Catchers' (and Foldit's) multiple meanings, focusing on the participant's perspective with respect to the imaginations inscribed or intended by design, which materialize in and are often articulated through different practices and forms of engagement with and along sociotechnical entanglements. These engagements are both constrained and enabled by nonhuman actors, which afford certain practices and open up possibilities for action, thus, shaping the assemblages collectively with human actors. Together, the intentions of various actors, such as developers and participants, and the human–technology relations, which are sometimes aligned, and sometimes in tension, bring HC-based CS systems into being in continuous negotiations, leading to these systems never being closed or completed.

While this observation may apply to many sociotechnical systems, the intriguing point about HC systems is that they are not intended to be complete or finished in the first place. Instead, designers and developers understand them to be transitory. As such, HC systems do not simply implement known concepts and realizations of human–technology relations but experiment with ideas and new ways of combining humans and technology.

While HC researchers seem to agree that human–AI partnerships or other combinations are the future of AI research, the question of *how* humans, human intelligence, or creativity should be included in computational systems remains at the heart of current research. Humans are often viewed as "assistants" for AI systems (Kamar 2016a, 4071), and, in this sense, the role of humans or human intelligence is already defined in advance. Alternatively, in the field of AI in general, computers or software are often considered assistants to humans. Apple Inc.'s virtual "intelligent assistant" Siri (Apple Inc., n.d.) is one

such example. However, as I argue below, this understanding of either the human or the computer as an assistant to the other does not sufficiently take into account how hybrid systems are constantly undergoing intraversions on their trajectories. Such a dynamic understanding of roles and relations was also proposed recently by computer scientist Zeynep Akata and colleagues in their "Research Agenda for Hybrid Intelligence" (2020):

> In HI settings, artificial and human agents work together in complex environments. Such environments are seldom static: team composition and tasks can change, interpersonal relations evolve, preferences can shift, and external conditions (for example, available resources and environment) can vary over time. Thus, competences cannot be fixed before deployment, and agents will have to adapt and learn during operation. As such, the ability of HI systems to adapt or learn is a prerequisite not only to perform well but to function at all. (Akata et al. 2020, 22)

As such hybrid systems tackle currently unsolvable (AI) problems, they cannot rely solely on existing systems but must create new human–technology relations. This means, for example, new combinations and interwoven systems of humans and software in which humans can take over certain computational tasks, ranging from simple classification to complex problem-solving that requires human-centered notions of creativity. From the previous chapter and related work on the construction of user representations by innovators and developers (Akrich 1995) and the co-construction of users and technology (Oudshoorn and Pinch 2005), we know that sociotechnical systems do not rely only on design and implementation. Rather, they rely just as much on practices of adoption and meaning-making by users themselves as they engage in and with the designed environment. Hence, human–technology relations in sociotechnical systems typically evolve continuously based on design choices and implementation details informed by user feedback and adoption practices that may go beyond the intended design.

Both examples, Stall Catchers and Foldit, can be understood as laboratories for human–technology relations. At the Human Computation Institute, HC-based CS projects function as laboratories for new human–technology relations in pursuit of particular visions, as discussed in Chapter 4, of how these relations should be for the "betterment of society" (Human Computation Institute, n.d.) in the future. For Foldit, the primary goal of combining humans and computers is to solve complex biomedical problems related to protein structure prediction and design. While the protein structure prediction problem has seen enormous progress in the last few years, especially with the development of the AlphaFold AI system, which I will return to below, there is currently no canonical solution to the protein design problem. Foldit can, thus, also be understood as a laboratory for finding new ways to solve these problems by combining humans and technology in novel ways. New computational developments are continuously being introduced into Foldit to "help players." In response to my question about the role of AI in Foldit (in early 2020), Foldit team member Gidon explained:

> [W]e always try to see it as humans and computers working together, and so certainly, I mean, if you consider AI to basically be more generally kind of algorithms, they have always been a part of Foldit like from the very beginning. […]. So, I think

> we are always trying to integrate the latest advances that we can get into the game to help players. And so […] it's sort of a collaboration, I think, between the human players and the AI optimization algorithms that are built into the game. And those are always advancing. (Jan. 31, 2020)

Human–technology relations in Stall Catchers and Foldit should, therefore, be seen as always preliminary and in constant flux, in a "dance" with each other (Pickering 2010 cited in Lange, Lenglet, and Seyfert 2019, 600), not only in their everyday appearance but also by design. A genealogical analysis of human–technology relations in Stall Catchers and Foldit reveals how these phenomena are shaped not only by their everyday becoming but also by the continuous pursuit of pushing the systems toward a goal, an abstract idea of ideal human–technology relations that has yet to be materialized. In the examples studied, this pursuit consists, for example, of the introduction of new tools and features, the attunement or restriction of different algorithmic flows (Mousavi Baygi, Introna, and Hultin 2021), and the opening up of new action spaces within the systems.

In this chapter, I apply the concept of intraversions to the analysis of the becoming and continuous changing of selected human–technology relations. The historical and processual study of human–technology relations takes into account instantaneous and gradual temporal developments and focuses on examples of participant–technology relations, followed by researcher–technology relations in the second part. The investigation of researcher–technology relations sheds light on other human–technology relations that are part of HC systems. As described in Chapter 4, these relations, although not often explicitly mentioned by HC advocates and designers, usually remain in the background. Of course, these relations also consist of participant–researcher–technology, developer–technology–participant relations, and other configurations and actors. For this chapter, however, I focus on these selected examples to trace the almost circular forward movements and shifts happening within these relations. Nevertheless, I include other actors who intervene and engage with the relations discussed. I argue that the ongoing changes are not merely incremental improvements to the systems and their (scientific) results but often represent intraversions of human–technology relations within the systems themselves. Inspired by Hutchins' work on distributed cognition, I show how these changes continuously transform and intravert the subject/object positions, the practices, and the nature of the tasks to be performed in these sociotechnical relations. Before concluding the chapter with a summary of key arguments, I use the example of ARTigo to discuss the dynamic interactions between human–technology relations within HC systems and the advances in AI that become visible with the concept of intraversions.

The motivation for this chapter also relates to how STS scholars Wiebe Bijker and John Law have described their interest and concern with technology: technologies or, in my case, human–technology relations, "*might have been otherwise*" (1992, 3, emphasis i.o.). Finding answers to why sociotechnical systems became what they are involves asking questions that concern all actors involved: the design and implementation decisions and their narratives, how human–technology relations actually unfold in the everyday, but also how participants, for example, "reshape their technologies" (Bijker and Law 1992, 3). Answers to such questions help us understand how they continue to evolve, since changes

in human–technology relations always create new potentials and entanglements while, at the same time, constraining what is possible. In this sense, relations partly stabilize over a certain time. However, they always remain open to new intraversions and tweakings according to the tensions between existing human–technology relations, everyday becoming, and future-oriented visions of HI. In the following, I first turn to participant–technology relations in Foldit.

Never Obsolete: Intraversions of Participant–Technology Relations in Foldit

The story of Foldit, like that of Stall Catchers, goes back to a scientific problem that could not be solved satisfactorily according to scientific quality standards. In the late 1990s, the Baker Lab of the Institute for Protein Design at the University of Washington sought an automated solution to get closer to solving the difficulty of protein structure prediction by developing the Rosetta program (Zimmer 2017). Rosetta computes the probability of interactions between segments of amino acid chains. This process is based on energy levels, such that the structure with the least energy is the most likely to fold (Gonzalez 2007).

In the beginning, the program relied on randomly selecting and altering protein segments, trying many different protein structures until it found the one with the lowest energy. If a move resulted in a lower energy level, it was accepted, and the program continued to modify the protein structure until it found a more optimal arrangement. This brute force approach required a lot of computing power to be successful. The researchers had about 400 computers on which to run their calculations (Kim in Gonzalez 2007, 3:30). However, it became clear that to make real progress with the protein structure prediction problem, they would have to drastically increase their computing power, building on thousands of computers. Biochemist, computational biologist, and director of the Baker Lab David Baker explained in a 2006 interview with the *Team Picard Distributed Computing* team, a community of distributed computing participants: "[b]ut there was simply no way to scale up our in house computing facilities significantly" (Baker 2006). Inspired by VDC projects such as Folding@home, David Kim, a project scientist in the Department of Biochemistry at the University of Washington, started to modify and adapt the Rosetta program to connect to the BOINC (Berkeley Open Infrastructure for Network Computing) distributed computing platform (University of California, n.d.). BOINC allows individuals to donate their idle computing power to scientific research projects that require large amounts of computing power to complete their calculations. Rosetta@home, the distributed version of Rosetta, was launched in 2005 (Zimmer 2017). Kim explained that "now with BOINC, we have thousands of computers that we could run our jobs on located all around the globe, and it's really exciting to see how it developed" (in Gonzalez 2007, 3:33–3:42). Researchers gain access to the power of supercomputing to tackle their computationally intensive research by distributing the computational problem and inviting volunteers to provide their computing power in VDC (Holohan 2013, 28).

Baker described Rosetta@home in an explanatory video about his laboratory's protein folding research as follows:

> What we're doing with Rosetta@home is analogous to searching the surface of a large rocky planet for the lowest elevation point […]. Imagine you have a team of human explorers working with you, and they're all exploring around the planet. If the team is small, it's quite likely that no explorer will actually find the lowest elevation point, in particular, if there are a lot of tall mountains that lead to explorers getting trapped in pretty good places on the planet. Now, instead, imagine that you have a very large team of explorers, and they each parachute down randomly on the surface of the planet and then start searching for the lowest elevation point. The more explorers you have, the more likely it is that at least one of them will find the lowest elevation point on the planet. Now, on Rosetta@home [we're] instead searching the energy landscape for a protein, trying to find the lowest energy structure for an amino acid sequence. The more computers there are doing these searches, the more likely it is that somebody will actually find it. (Baker in Gonzalez 2007, 3:43–4:35)

People could contribute to solving the scientific problem of protein structure prediction by downloading and installing the Rosetta software. Unlike the later Foldit project, participants would not interactively fold proteins themselves, or even personally invest any of their free time to contribute. Instead, Rosetta@home would simply run in the background, using the computational power of the participant's computer when it was idle. If they wanted, participants could observe Rosetta's moves through a screensaver that came with the software. Baker described this approach as a collaborative effort between professional scientists and the public that would also change the relationship between them:

> Because it's a whole new step forward in the relationship between scientists and the public. To solve the problem of protein structure prediction, it's quite clear that it's really not possible without the contributions of, of people from all over the world […], like yourselves because it's such a big computing problem that there, it just cannot be done with any in-house resources. So we can only do it collaboratively as a collaboration between us and you and through this collaboration we can solve the problem which I really think couldn't be solved otherwise. (Baker in Gonzalez 2007, 6:00–6:32)

Baker's emphasis on the potential of Rosetta@home to change the relationship between scientists and the public must be considered in the context of how science was perceived by the public when VDC emerged as a new phenomenon "after a time when science's relationship with the public was at a low ebb. In the mid-1990s, Carl Sagan observed that the general public's attitude toward science was increasingly one of alienation and even hostility" (Holohan 2013, 27). Against this backdrop, VDC presented a promising approach to bridging this gap. Distributed computing—especially with the BOINC architecture—made computationally heavy science projects, which had previously been restricted to a selected group of professional researchers, accessible to a wider range of people (Holohan 2013, 27–28). Despite the enthusiasm for VDC and the growing number of projects in this space, Holohan declared in 2013 that "VDC is still in its infancy and the democratization of science enabled by the fewer barriers to knowledge production that the Internet offers has not yet crystallized into institutional paths" (Holohan 2013,

28). In addition, the user's contributions in the distributed computing setting remain somewhat passive. To consider the assignment of volunteers' roles in Rosetta@home, the active role of human participants—beyond downloading and installing the software—is reduced to providing computational power, to the very act of stepping away from the computer so that the idle cycles can be used for Rosetta@home. In this way, the following observation by Mackenzie, who refers to a similar distributed computing project called THINK created by researchers at the University of Oxford to analyze the interactions of specific molecules and proteins in cancer research (2006, 187–188), also applies to Rosetta@home's participants. "The relatively anonymous membership has no claim or control over the research. Their mental or intellectual effort has been deliberately figured out of the software, and their computers' execution of the 'virtual screening' processes largely disappears into a calculative background." (Mackenzie 2006, 188)

The observatory or passive role of participants makes Rosetta@home very different from Foldit, which, in fact, originated from Rosetta@home. This development may seem somewhat surprising at first. How did a project like Foldit, in which participants are the primary entity actively engaging with protein structures, evolve from a project that is all about harnessing scalable (machine) computational power? How was the decision-power in the individual protein folding steps transferred from the randomly proceeding program to human participants?

Foldit's Legend

The shared narrative (or "legend," as one Foldit team member described it) is that it grew out of Rosetta@Home participants actively requesting to play a different role in solving the protein structure prediction problem. Computer scientists and one of Foldit's creators, Adrien Treuille, described in an interview with Neil Savage for the *Communications of the ACM* magazine that participants "started noticing they could guess whether the computer was getting closer to or farther from the answer by watching the graphics" (Treuille in Savage 2012). Foldit developer Daniel further explained in our interview in early 2020 that participants observed what the program was doing, identified its move choices as questionable, and, thus, began to voice their feedback to the team (Jan. 24, 2020). Foldit researcher José described the feedback from volunteers in our conversation:

> [L]ots of them really liked this research, and […] there were the screensavers that came with [the Rosetta@Home software] so they could watch what the computer was doing […]. And there were a couple of requests to be able to interact with the computer. So that kind of, I think the legend goes, that blossomed into an idea to make an actual game where, instead of the computer running things idly, you could actually sit down at the computer, and you could try to direct how the simulation progressed. Well, not the simulation but how the computation progressed. (José, Jan. 22, 2020)

According to this "legend," the motivation of the participants eventually led to the development of a new CS game. "[T]hat's how Foldit was born," José recalled (Jan. 22, 2020). The collaboration between the Department of Biochemistry and the Center for Game Sci-

ence at the University of Washington began working on the new project in 2007, and it was publicly launched in 2008. Gidon, who has been part of Foldit since its early days, described that when they started to develop Foldit from Rosetta's software,

> [they were] trying to […] get some kind of human reasoning involved in […] what was Rosetta@home's sort of purely computational kind of directed random approach to folding proteins, and maybe if we had people get involved and help direct some of that search essentially, then that might come out with something different or something better or something, like a different way of searching through this space of proteins than just the kind of computational approach. (Gidon, Jan. 31, 2020)

The hope, as Gidon described it, was to come up with better protein structure solutions and new ways of exploring the space of protein structures than was possible with the purely computational approach of Rosetta@home.

It should be noted that the reference to "human reasoning" in Gidon's quote echoes the imagination of the human in HC systems described earlier (Chapter 4), which reduces humans to thinking about something in a logical way and, hence, to their cognitive processes.

In the beginning, Foldit's development, driven mainly by a core team of three developers, was highly exploratory and included many experiments and iterations[1] because, as Gidon stated, the team could not simply build upon well-established processes to create such a CS project: "we worked on it for about a year trying lots of different things cause we didn't really know what was gonna work and what people were gonna be good at" (Jan. 31, 2020).

However, even after its official launch, Foldit was not complete or finished, but remained a laboratory for CS games and human–technology relations. Since its initial design, the Foldit team has continuously been improving and updating the design and features of the software so that the Foldit of today includes many more new means of interaction between participants and software than when it was launched in 2008. Even though the Foldit team has driven these main developments, in the following, I will show how all different human and nonhuman actors are involved in this process. The existing participant–technology relations have most prominently formed and influenced Foldit and its becoming in crucial ways.

In its earliest moments, Foldit consisted of a basic UI and controls for the participants to interact with the protein. These controls already included "automated algorithms where the player [could] let the computer take over and figure out some of the details" (Gidon, Jan. 31, 2020) and, over time, Foldit developers added more and more such tools to make the interactions more user-friendly and allow for more ways to manipulate proteins. These tools, as the team member with the username Zoran explained in a Foldit blog post, were designed according to "the way players tend to manipulate proteins, and according to the way expert biochemists would like to alter the configuration" (2009). The tools were, thus, presented to participants as an offer to facilitate protein

1 To test Foldit, they invited some of the Rosetta@home participants via the project's forums (Gidon, Jan. 31, 2020).

manipulation. Only a year after the launch, it was again participants who requested the introduction of automation tools that could help them in their approach: "players began requesting the addition of automation tools so that they could more easily carry out their strategies." (Cooper et al. 2011, 2) With the introduction of an editor and aforementioned *recipes*, effectively computer scripts for controlling Foldit itself, participants were given the opportunity not only to use the tools provided but also to create their own tools by writing and sharing recipes (Cooper et al. 2011, 2). The team explained in the announcement text that

> [i]n the spirit of allowing you to shape the course of scientific research, we've been planning to do something much more powerful: allow you to design, share, refine, discuss and rank new tools by combining the low level building blocks into more complicated operations through a simple visual interface. (Zoran 2009)

The goal of this new feature was that anyone, even participants with no programming experience, could build their own tools. Besides facilitating game play and puzzle solving for participants, the creation of tools or scripts also served to discover new approaches that would improve the computational performance of protein folding (Zoran 2009):

> It was also our intention to infer optimal strategies from the Foldit players and use them to improve fully automatic approaches. Rather than performing machine learning on gameplay traces of Foldit players, we decided that the players themselves would likely be much better at systematic abstraction of their strategies. (Cooper et al. 2011, 2)

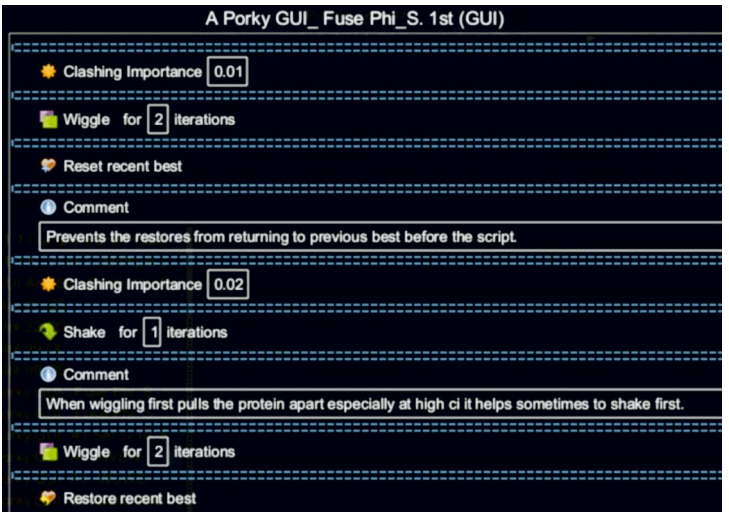
Although Foldit's developers would continue to introduce new tools, participants could now build their own tools, automate the steps of their choice, and were, thus, equipped with new "powerful means of managing [...] complexity" (Cooper et al. 2011, 1).

First, so-called GUI (Graphical User Interface) recipes, now also referred to as "the original type of Foldit recipe" (Foldit Wiki 2017b), could be created. Participants could choose from existing Foldit tools, such as "Wiggle" and "Shake," and create new combinations and sequences of these tools in a "simple block-based visual programming interface" (Cooper et al. 2011, 2) without actually having to write code in the dedicated editor (see Figure 6).

While GUI recipes were relatively easy to create, they had drawbacks compared to text-based high-level programming languages, lacking several features often found in these programming languages. For example, it was not possible to create loops of specific steps, i.e., to repeat instructions until a condition was met, or to define code paths depending on conditional choices (Foldit Wiki 2017b). The GUI recipes also had a technical limitation that made it impossible to run them in the background when the client was minimized (Foldit Wiki 2017b), which, given the practice of many players to use multiple clients and run recipes 24/7 when not actively playing Foldit, could make their use quite cumbersome. They were discontinued in 2021. However, just a few months after the introduction of the GUI recipes and even before their end, the "Foldit Lua 1" interface was added, again at the request of participants who wanted to take advantage of

the full potential of scripting languages and move beyond the limitations of GUI recipes (Cooper et al. 2011, 2). This interface allowed the creation of "script recipes" using the Lua programming language (Cooper et al. 2011, 2). The introduction of a "full" programming language enabled the creation of more advanced scripts with loops, if-then-else logic, and the ability to define and call functions. To this day, improving the editor and adding new capabilities and features remains a focus of the Foldit team. This also led to the introduction of a new editor, the "Lua v2" interface, which is the current version.

Figure 6: Screenshot of a Foldit GUI recipe



Source: LociOiling 2017

Participants in Foldit do not necessarily write recipes only for their own use, instead they often share them with other participants. In fact, many participants do not create their own recipes from scratch but refine existing recipes, which they can download and add to their "cookbook" (Cooper et al. 2011, 3). In these cookbooks, each participant can manage their collection of recipes in the Foldit client. Participant Lucas, for example, explained to me during our interview that he had an extensive collection of recipes in his cookbook. However, he argued,"I have a very short list that I use all the time and … I just tend to use the same ones over and over. And there are people who cycle through a much larger collection of recipes, and they can beat me; they can get more points" (Mar. 17, 2021). Participants can search for and navigate the recipe catalog on Foldit's website to explore new recipes (Foldit Wiki 2017b). This web-based catalog also allows participants to rate the recipes they use to assess their usefulness better.

With the introduction of the possibility to write, run, and share recipes, human–technology relations within Foldit intraverted from automated algorithms/tools as ready-made utensils used by Foldit participants to self-made means. Furthermore, with

the introduction of the recipe editor, participants became developers[2] and tool creators, and even though participants could continue to play as before, their core activities within Foldit—folding and designing proteins— now extended beyond manually manipulating protein structures to programming scripts that, as participant Salma explained in her written response to my questionnaire, "automate actions that would get very tedious to do manually" (May 8, 2021).

Most participants in Foldit today rely heavily on recipes in their gameplay or are "carried" by them, as participant Sylke pointed out during our conversation in early 2021: "The algorithmic tools carry me; they are the basis of all my solutions. […] I use recipes all the time" (Feb. 27, 2021).[3] With this development toward automating manual steps, recipes reintroduced what participant Friedrich described as "staring at what the algorithm does" (Mar. 9, 2021). Participants estimated that recipes would "work" on the puzzles about 90 percent of the time and that only ten percent of the time would consist of so-called "handfolding" or selecting and starting new recipes. In a paper on the use of recipes in Foldit that was written in collaboration with Foldit participants, Ponti et al. (2018) even calculated that, in the case of expert Foldit users, only two percent of the total time spent working on proteins can be attributed to human contributions, while computers contribute 98 percent. It is worth noting, though, that to become an expert in Foldit, human participants have to invest a lot of time not included in this calculation. Similarly, even if recipes require minimal editing, at a certain point, many hours have been invested in writing them, usually by different participants. James, who has been participating in Foldit for many years, described in our conversation that he had even written a recipe to automate the entire design of the protein (and not just the steps to fold it) (Feb. 11, 2021).

"Staring at what the algorithm does" is very similar to the pre-Foldit times when participants could only run the Rosetta@Home software on their computer and watch the computational steps via the screensaver. However, despite this similarity, there is a significant difference between observing Rosetta@Home and running scripts in Foldit. Individual participants in Foldit ultimately control the process and the individual steps of protein folding. Team member Daniel emphasized the player's choice of recipes as opposed to Rosetta@Home: "[L]ate game is running automated recipes and scripts to further refine and optimize. Which, at that point, is almost resembling Rosetta@Home in

2 "Developers" here refers to the development of scripts for automating steps in the Foldit game and not to the development of Foldit itself. This distinguishes participants as developers from the Foldit team developers.

3 At the same time, some participants still prefer manipulating proteins manually and advocate strongly for "hand folding." Brandon argued, for example, in his written response to my questions that he no longer uses recipes because they would still include bugs and have to be restarted upon interruption: "Early on (in my Foldit experience) I would use them a LOT, and found them to be seriously helpful to my work. The last 6–8 months, I've actually all but ceased their use, becoming entirely a 'hand folder.' This primarily came to be due to a bug that was resulting in Foldit randomly crashing, which is not conducive to a Recipe (algorithmic tool) producing results. Primarily because if they are interrupted, they have to start over (there is no 'resume'-like feature), but also because Foldit hasn't the ability to start itself back up and automatically restart what it was doing. (It's very much like a DVD player and watching a movie, where you're half-way through it, and the power goes out. You can approximately get back to where you were, but it won't resume back to the split-second that it left off before the power outage)" (Mar. 4, 2021).

terms of the computer doing a lot of the work. But the player is choosing what scripts and what algorithms and computations to run." (Jan. 22, 2020)

"Late game" refers to one of three stages that are distinguished in Foldit. "[E]arly game is getting the basic structure, the basic shape. Mid-game might be like moving some things around and refining that" (Jan. 22, 2020), Daniel explained. In the late game, the main puzzle-solving task that the human participant has to perform is what I call "orchestrating" recipes. It consists of selecting, starting, and stopping recipes, practices based on knowledge of which recipes are best applied at what time, with what parameters, and for how long they should run in each case. David, for example, a participant with many years of experience, described his approach as follows:

> [I]f I have to make a model, then I first make a rough draft, I then set it up on those servers and then let the servers run scripts to further develop it, and after a few hours, I look again: How's it going? Is it good? Is it not good? Do I need to run another script, […] and that's how it goes all day. And in the evening, I prepare things to run overnight. So those PCs also run 24 hours a day, at least some of them. And for important puzzles that I consider important, such as corona puzzles, or in this case, the grip-binder, I also use more machines to do larger processing. (Mar. 4, 2021)

While David manually creates the first drafts of the protein, scripts or recipes then take over to refine those drafts. But David still stays in the loop by "directing the algorithm," as Foldit team member Gidon explained. "So, the player can sort of do whatever they want, including starting up some kind of algorithm and then stopping it basically. So, […] I would say for the most part the player is sort of in control of the algorithms" (Jan. 31, 2020). As becomes clear, playing Foldit using recipes involves more than just running a script. Instead, it includes monitoring the process of each recipe and continuously analyzing and evaluating its progress and performance. It is crucial to understand the scripts and know which recipes to use and when to use them. Participants who contributed to my research agreed that "[s]electing which tool to apply in which situation is the art of it" (Salma, May 8, 2021) and requires "human creativity" (Aram, Feb. 28, 2021):

> The tools only help little if the initial design is not good. And here, human creativity is needed. Almost everything depends on it, and the tools are then only used for "finalizing" and refining. On the other hand, you will hardly get a top-score design if you do not use any additional tools. So here, there is a significant mutual dependence. (Aram, Feb. 28, 2021)

The relations between the players and the software had, thus, shifted or intraverted again. They now resembled a symbiosis; neither manual handfolding nor pure execution of recipes alone lead to a successful protein design. Rather, it is precisely the symbiosis, as described by player Aika (Apr. 10, 2021), that leads to solving a protein puzzle which can be tested in the lab. David, therefore, described the relations between humans and algorithms as "mutually enhancing" (Mar. 4, 2021). With this intraversion, instead of one subject controlling or merely using algorithmic or human input, algorithmic tools and participants played equal roles and were partners in solving a puzzle.

Some participants even took this symbiosis a step further by increasing their computing power with additional servers, running multiple clients to parallelize different approaches to the same puzzle, or working on different puzzles at the same time. Arthur, for example, described "actively" playing around for about two to three hours in the evening, but "the five clients, of course, automatically run 24/7 a week in the background with different programs" (Feb. 12, 2021). David's additional servers, which he acquired specifically for Foldit, were always ready for a Foldit job (Mar. 4, 2021). Multiple servers and clients allowed participants to go beyond working on one puzzle at a time. Participant–technology relations are extended over a larger network of computers and clients that exceed previous relations.

In this way, human participants and computational entities, in fact, become indistinguishable when viewed from the outside. For example, if recipes are running while the human participant is doing something else, the corresponding user in Foldit will be recognized as active in the game statistics.

In addition to participants who actively seek to expand and amplify the human–technology relations of which they are part, some prefer to fold proteins manually, using the algorithmic tools simply as *tools*. Foldit's task is divided into the manual manipulation of proteins and the writing and orchestration of recipes, thus delineating two main practices for approaching proteins. Interview partners commonly distinguished between a biomedical approach to solving the puzzles, which relies on a biomedical understanding of proteins, their elements, and how they fold, and a computational approach, which, instead, relies more on writing and orchestrating scripts and optimizing toward points. Not all Foldit participants follow one approach exclusively. Some have adopted a combination of both. Moreover, since the participants interviewed for my research were highly engaged in Foldit compared to the majority of registered users, it is reasonable to assume that others probably adopted other approaches, such as an "intuitive" method of folding. Or, as I did in my first attempts to fold proteins, they may primarily rely on trial and error, experimenting with the tools and scripts provided without adhering to a specific strategy. Although the two approaches described, the biomedical and the computational, are very different, they can both lead to successful protein designs. Participant Lucas described these different approaches as different options while clearly identifying himself with the biomedical approach:

> [S]ome people, their interaction is primarily choosing what script to run and being familiar with how the scripts behave and which is the one they want to use next. And they might drive that entirely based on their score. My interaction is, I'm interacting with the protein, and I'm trying to shape the protein based on what I believe it should be like, and I'm using the tools only as tools to do that. (Mar. 17, 2021)

It was important to Lucas that the tools remain *tools* that facilitate their manual handfolding so that he can interact with the protein rather than with the scripts. Similarly, when I asked what tools they would use most often, Brandon explained:

> My brain! Then my arm/hand and my mouse… :) But in terms of Foldit's internal tools… Wiggle, Shake, Bands…. Honestly, as a 'hand folder'—someone who spends

> the majority of their time on a puzzle doing things by hand instead of using Recipes—it's probably easier for me to list what I don't often use! (Mar. 4, 2021)

Not only were their brain and hands in an extension of the mouse the first "tools" they mentioned, but Brandon also distinguished between "internal tools," referring to those tools that Foldit developers had implemented from the beginning, and recipes, which they generally did not use often. They also considered themselves to be a "hand folder," someone who mostly folds the protein with manual moves and only used certain recipes, which Brandon considered to be "'utility' tools" (Brandon, Mar. 4, 2021).

Following the distinctions between hand folders vs. participants relying heavily on recipes or biomedically vs. computationally oriented participants, the introduction of the ability to create their own tools with recipes created another distinction among participants: tool makers vs. tool users. While tool makers were typically tool users, the latter built on recipes provided by tool makers. Since recipes can be complex programs, their exact functionality was not always accessible to all tool users: "I'm a recipe user and I'm not a recipe maker. So […] as much as the description would convey to me. […] [T]here comes a time when [I] […] say all right, this is a black box, but I know it works" (Aika, Apr. 10, 2021). Therefore, participant–technology relations also differ depending on the individual player's approach to solving puzzles and their practices. The role played by participants, simple algorithmic tools such as "Shake," and recipes within the relation varies: While in some cases the subject/object positions seem to be clearly separated, for example, between participants *using* algorithmic tools, in other cases, recipes and participants are equal partners forming a symbiosis.

Along Foldit's continuous development, various intraversions of its participant–technology relations can be seen in its movement from fully automated algorithms designed to solve the protein prediction problem to a hybrid system in which computational tools assist humans in manually manipulating proteins, followed by a shift on the participants' part to rely on automated "tools" (recipes) to manipulate the protein better, to today's sprawling interplay of participant-developed algorithmic recipes and individualized styles of applying the former to achieve ever more performant gameplay. Ultimately, these emerging relations even lay the foundation for entirely new human–AI relations to form beyond Foldit's initial scope in the form of the highly advanced AI system AlphaFold.

Introducing the Artificial Intelligence System AlphaFold in Foldit

DeepMind, a British-American company and Alphabet subsidiary, introduced the AlphaFold2 model in 2020, which significantly outperformed the previous state of the art. AlphaFold2 is an AI system for predicting protein folding, which far exceeded expectations at the *Critical Assessment of Structure Prediction* (CASP) conference, a scientific competition in which participating groups attempt to either experimentally assess or compute structures for specific amino acid sequences (University of California, Davis, n.d.). "This will change medicine. It will change research. It will change bioengineering. It will change everything," said Andrei Lupas, an evolutionary biologist at the Max Planck Institute for Developmental Biology in Tübingen, Germany, in an interview with

Robert F. Service from *Science* (Service 2020). While DeepMind had already participated in CASP in 2018, the protein folding problem was, in fact, declared to be solved in 2020 with AlphaFold2. AlphaFold2 builds on DL and a so-called "attention algorithm," imitating how humans might approach a jigsaw puzzle (Service 2020). What is particularly interesting in the context of Foldit and its continuous intraversions is that AlphaFold2 itself was not only partially trained on data that originated from Foldit[4] but, according to the DeepMind co-founder Demis Hassabis, its approach was directly inspired by Foldit and its game-like approach:[5]

> [T]here was this game called Foldit, where some people had created a puzzle game out of proteins. Gamers played it – and what they were doing was actually trying to turn the protein into a particular shape. It turned out that through playing this game … they actually discovered a couple of very important structures for real proteins. Firstly, it was a fascinating use of games in science – and games is another one of my interests. But secondly, it kind of suggested to me that somehow these gamers had trained their intuition and their pattern-matching capabilities so that somehow they were able to do what brute-force computer systems couldn't at the time – and actually come up with the right shapes. That made me think that AI could maybe try to mimic that intuitive capability that those gamers were demonstrating. (Hassabis 2020)

Furthermore, as indicated in this quote, the Foldit participants' "intuitive capabilities" and their development of "intuition and pattern-matching capabilities" were seen as fundamental to AI development. By mirroring the strategies used by human participants, it was considered feasible to train AI systems in protein folding.

Looking back at 2020 and summarizing the best news about Foldit from each month, Foldit member Dev Josh lists AlphaFold2's win at CASP, mentioning that Foldit was cited as an inspiration (2021a). In what can be understood as another intraversion in Foldit, an integration with AlphaFold was then introduced in Foldit itself so that participants could contribute solutions they had jointly created with algorithmic tools to the AI program. The AlphaFold feature was introduced as a button that, when clicked on, opened an interface where participants could upload their protein solutions to Foldit's server, where the AlphaFold algorithm would run and compute its prediction for the uploaded solution (Bkoep 2021a). Participant David explained the AlphaFold process after submitting his solution to the AI algorithm as follows: "[S]ee what you [AlphaFold] come up with in an hour or so. And then you'll get a result back, and you can see if it's the same shape

4 AlphaFold was trained on solved protein structures that were stored in the Protein Data Bank, to some of which Foldit participants have contributed.

5 AlphaFold is not the only AI system that has attracted attention and advanced protein structure prediction. Influenced by the successes of AlphaFold, the Institute for Protein Design at the University of Washington first developed the transform-retrained Rosetta (TrRosetta) algorithm for protein structure prediction (Yang et al. 2020; Baker Lab 2021) and later, an even more accurate model called RoseTTAFold, which was "[i]ntrigued by the DeepMind results" (Baek et al. 2021, 871). In 2022, Baker, along with Hassabis and DeepMind's senior staff research scientist John Jumper received the Wile Prize in Biomedical Sciences "for procedures to predict highly accurate three-dimensional structures of protein molecules from their amino-acid sequences" (Rose 2022).

you had in mind. If it's not, then apparently there's still something in there [the protein structure] that's not right" (Mar. 4, 2021). The ascribed role of the AI model is that of an assistant that should check the uploaded solutions and "help out Foldit players," as a team member with the username bkoep already anticipated in 2019 in an online chat with participants (2019). AlphaFold, thus, functioned as a control body and enhancement guide, once again transforming participant–AI relations.

Rather than making the game and the participant–AI interplay or the human in the loop redundant—a fear often expressed in response to AGI narratives, as was described in Chapter 4—this intraversion actually allowed new relations to emerge. With the possibility of using AI models for most cases of protein structure prediction, the role of human participants could move on from their original task and work on other problems.[6] As bkoep clarified in a forum post on RoseTTAfold, a model inspired by AlphaFold and developed by the Institute for Protein Design at the University of Washington (Baek et al. 2021, 871):

> These deep neural networks are very, very good at protein structure prediction. There will always be cases where the network predictions fall short or do not tell the story about a protein, and human predictions may still be useful in some of those cases. But, by and large, these neural networks seem to be the better option for raw protein structure prediction. We think that humans have more to contribute to other problems, like protein design or model-building with experimental data. (Bkoep 2021b)

This shows how the target and the very purpose of the system itself keep transforming. While Foldit started out as a protein structure prediction program, it now focused on protein design, which was even more difficult to tackle but became possible to address by the continuous developments, changes, and intraversions of the participant–technology relations described. Tracing human–technology relations in Foldit and unraveling their intraversions over time shows how the roles of individual elements of the system cannot be ascribed and fixed once and for all. Not only do the participants themselves contest meanings and their assigned roles, as described in Chapter 5, but their roles also change with intraversions due to new technological developments or seized potentials. Likewise, the roles of automated tools, scripts, and AI system are also subject to intraversions in Foldit. In fact, the intraversions discussed in Foldit's participant–technology relations describe only a small part of the ongoing evolution of these relations. In September 2022, a new version of Foldit was introduced that included new AI-based tools alongside the existing "classic" algorithmic tools. For instance, "Neural Net Mutate," which was much faster than its classic version, was now added to the older "Classic Mutate" tool (Haydon 2022, 0:43). This is another example of how intraversions continue to unfold within Foldit's human–technology relations.

I would now like to turn from intraverting human–technology relations in solving the problem of protein structure prediction to human–technology relations in Stall Catch-

6 Additionally, as participant Salma pointed out when explaining why she thinks humans will still be needed in the future even as AI's capabilities continue to advance, AI models like AlphaFold demand extensive computing power, whereas "[c]itizen scientists are pretty low-cost!" (May 8, 2021).

ers, where the focus is on advancing Alzheimer's disease research. Since Stall Catchers was the primary fieldsite of my research, it will be discussed in more detail regarding the overall project to be able to unravel not only participant–technology but also researcher–technology relations in the second part of this chapter.

"First Use No Humans:" Intraversions of Participant–Technology Relations in Stall Catchers

"We've always had a division of labor between machines and humans in Stall Catchers" (Michelucci and Egle [Seplute] 2020), write the director and the CS coordinator of the Human Computation Institute about an ML competition organized in collaboration with the data science crowdsourcing platform DrivenData. Although the biomedical engineering laboratory's initial attempts to automate the analysis of Alzheimer's disease research data using ML techniques failed due to a lack of AI model accuracy, which then led to the development of the CS game Stall Catchers, human–machine relations have played an important role in Stall Catchers from the very beginning. Participants' annotations, for example, were computationally reviewed before being included in the final crowd answers, and ML had been part of the data preparation steps in Stall Catchers from the beginning. Various developer–technology and researcher–technology relations had to be formed and aligned to enable the analysis of the Stall Catchers research data in the first place. The researcher–technology relations building the data pipeline create the data around and on which the participant–technology relations operate and, hence, set the stage for and frame these other relations.

In the early years of the Stall Catchers project, the ML algorithms within human–technology relations were considered "preprocessors" to facilitate the tasks to be performed by human participants. The role of human participants, by contrast, was to take over the task of annotating the presented data, which in technical terms can be understood as an image-recognition problem, of which the AI was not yet capable. Here, as in the HC imagination of the "human in the loop" discussed in Chapter 4, humans were indeed meant to be "assistants" to algorithms in the sociotechnical system. But the relations between participants and technology were mutually supportive because even though the role of AI in Stall Catchers was limited to preparing data—or rather, taking over a few steps in the very complex process of preparing data—participants would not have been able to annotate data without the preceding preparation by a network of humans, computers, and ML models. Or, to put it differently, even though computational models could not take over the analysis task, they still controlled and evaluated human input—similar to the example of CAPTCHA described in Chapter 4. The annotations of individual participants were not considered reliable because participants were imagined as nonexpert humans in the loop. To ensure the quality of the crowdsourced annotation results and to build trust in the sociotechnical system, customized algorithms were implemented in the game to evaluate individual participant's analysis skills.

The algorithmic assessment of each participant's "skill level" began with the tutorial once a participant had registered with the platform. Here, participants were presented with videos to which the correct annotation answer was already known in order to eval-

uate their "sensitivity" for identifying stalls. But even after the tutorial was completed and the participants were analyzing "real" research videos to which the correct annotation results were not known, the algorithmic evaluation of their skill level continued with so-called "calibration movies," which were regularly presented to participants between "real" research videos. These "calibration movies," to which the expert answers were known and which, for participants, were indistinguishable from research videos, served to continue the evaluation of an individual participant as they played Stall Catchers. The blue bar to the right of the video frame in the game interface indicates the participant's skill level, which is calculated from the number of videos they correctly classified. These calculations also consider the difficulty of a given video, that is, how hard professional researchers rated it based on the detectability of the stalls. In this way, each participant's skill level was not only tracked "backstage" by the algorithmic system but also revealed to the participant, who could monitor the *value* of their contribution.

Crucially, and building on the idea of the "wisdom of the crowd" (discussed in Chapter 4), algorithms also calculated so-called "crowd answers," the final output of Stall Catchers that was ultimately sent back to the laboratory. This was done by combining the individual answers and weighting them according to the skill level of each participant. Despite the fact that humans perform the core task of analysis, their impact on the results is still governed or curated by automated algorithms. However, the algorithmic control was not perceived negatively by participants. John, a Stall Catchers participant, described it more positively:

> So, you are the little machines doing the work and trying, and the algorithm makes sure that if you're having a bad day and you wanna beat somebody and you're just zipping through files, that's still okay. And if you're taking it serious cause you realize that's not what you should do, that's good too. And it's all gonna work out, and […] things are okay, and we just need you to spend some time for us if you'd like. (May 7, 2020)

John described the algorithms as safeguards to ensure that participants only provide information that is beneficial and not "harmful" to the platform, and, consequently, to Alzheimer's disease research. John even referred to the participants as machines rather than the algorithms, which he anthropomorphized. John's response shows that an interesting participant–technology interplay emerges in Stall Catchers, which goes beyond the HC imagination of humans and algorithms working in computational systems as mere assistants to each other. Instead, while participants take on the analysis task, they and their annotations are continuously evaluated and rated by algorithms due to their unreliability in consistently delivering the same quality. Valuable annotations, therefore, emerge only in the interplay of participant–technology relations.

These relations also open up new spaces and potentials for entanglements that the designers had not intentionally implemented. This became possible due to the possibility of engaging with the software differently and the participants' creative practices. Following what Mackenzie describes for software, I argue that these participant–software relations unfold within the situational context and the different practices and intentions of actors (2006, 6). The example of participants bypassing the programmed par-

ticipant–platform procedures described in the previous chapter shows the everyday becoming of these relations, which, despite endless efforts to design and implement them in a specific way, are always open to new and unintended formations and practices. These tactics, which Michel de Certeau describes as undermining pre- and inscribed directives ([1980] 2013), allow participants to enter into different relations with technology and contribute to the formation of Stall Catchers as a sociotechnical assemblage in the everyday.

However, participants and their practices are not the only actors directly influencing their engagements with the Stall Catchers platform. Instead, servers can fail, maintenance work can crash the platform, and expired certificates can make Stall Catchers unavailable. The latter, in particular, was a disruption I experienced several times during my fieldwork and collaboration with the institute. Most of the time, participants would send an email to the team reporting the unavailability of Stall Catchers. However, even though the problem was known, updating the expired certificates sometimes still led to complications. As discussed in Chapter 4, such disruptions required the full attention of the Stall Catchers team and kept them from working on their primary tasks or focusing on meetings (fieldnote, Aug. 19, 2021). These breakdowns were as much a part of the everyday unfolding of Stall Catchers as the careful design and implementation of human–technology relations (Jackson 2014). During one of these disruptions, the entire platform came to a halt, with participant–technology relations unable to form and analyze research videos. This example, similar to the data outages described previously (Chapter 5), shows how participant–technology relations in Stall Catchers must be actively ensured for Stall Catchers to "always [be] there," as participant Akin explained in our conversation (May 11, 2020).

Thus, these participant–technology relations—even if they did not always unfold due to subversive user practices that the team had not anticipated or disruptive certificate issues—had to be ensured at all times. At the same time, they were not even meant to be complete and remain as they were but were considered a work-in-progress and, as such, preliminary in nature. This is already indicated by the title "The machines are coming! (but the humans are staying)" (Michelucci and Egle [Seplute] 2020) of the 2020 blog post cited above. In fact, changes to the originally designed human–technology relations in Stall Catchers were an inherent part of the vision to develop hybrid thinking systems (Bowser et al. 2017). It was, thus, already part of the HC imaginations and, as I show below, considered a moral obligation for the institute that participant–technology relations would intravert.

"The Machines Are Coming:" Artificial Intelligence Bots in Stall Catchers

The Human Computation Institute follows the normative oath "First use no humans," discussed in Chapter 4, which means that if a computational solution to an analysis problem exists, it is not justifiable to ask humans to do it. While there was no computational solution to the task of annotating Stall Catchers data in 2014 when the platform's development began, this may no longer be the case. Therefore, the institute has been pursuing different efforts to explore new and modify existing human–AI relations.

During my collaboration with the institute, it, together with computer scientists and human–AI collaboration researchers Kori Inkpen and colleagues, conducted an exper-

iment on human–AI partnerships in 2020, in which an AI system played the role of an assistant to human participants (Inkpen et al. 2023). Visualized as a robot icon in the UI, it pointed to the annotation label the AI system predicted to be correct. In this experiment, the roles across the human–technology relations were intraverted; it was no longer the human who was considered to take an assisting role but the AI system, whose recommendation the human participant was allowed but not required to follow. The study also sought to investigate under what circumstances participants would trust the AI the most. This intraversion happened in a sandbox version of Stall Catchers, i.e., an experimental environment that was closed off from the actual Stall Catchers platform. While this study presents an interesting development shaping Stall Catchers' possible future participant–technology relations, to which I return in Chapter 7, another example of intraversions in Stall Catchers did not occur in a sandboxed experimental setting but rather during a Catchathon and, thus, directly impacted the participant–technology relations.

Around the same time as the collaborative study with Microsoft Research scientists, the data science competition platform DrivenData in collaboration with the Human Computation Institute and MathWorks, launched an ML competition called "the Clog Loss Challenge" to try again to automate the data analysis problem in Stall Catchers. This time, the automation was based on hundreds of thousands of human annotations that had been collected over the past few years and could serve as training data for AI models. The task was to build "machine learning models that could classify blood vessels in 3D image stacks as stalled or flowing" (Lipstein 2020). More than 1,300 solutions were submitted by over 900 participants in this challenge (Lipstein 2020). In a blog post published during the course of the competition, the Human Computation Institute expressed its hope that ML models could be used to create new human–AI relations:

> We could then use such models to label all the 'low-hanging fruit' – the easier vessels, and save the more difficult ones for our human catchers. In this arrangement, even if the new AI systems can only reliably label 20% of the images, that's still 20% less time that volunteers would need to spend on a dataset, and a 20% speed up in the overall analysis time. (Michelucci and Egle [Seplute] 2020)

Once again, computational systems were considered effectively as preprocessors for humans, picking the "easier" videos from the datasets before the remaining videos were presented to the participants. Other ideas included building ensembles or teams of AI systems that together could achieve the accuracy required for analyzed data (just as human annotations were combined into crowd answers) or introducing standalone AI users that would play Stall Catchers alongside humans (Michelucci and Egle [Seplute] 2020).

Even though the models resulting from the Clog Loss Challenge in 2020 were significantly better than those from the first attempts in 2014, the institute's blog post after the end of the challenge states, "machines are still falling quite a bit short of our high quality requirements in Stall Catchers" (Michelucci and Egle [Seplute] 2020). However, this did not mean that the models could not be included in Stall Catchers in "useful ways to speed up [the Stall Catchers] search for an Alzheimer's treatment" (Michelucci and Egle [Seplute] 2020). The Human Computation Institute, thus, invited the winners of the challenge

to work with the institute to program an AI bot to be introduced to Stall Catchers and to "play" it alongside human participants.

For the first Stall Catchers bot experiment, one of the winners joined the institute's team to build an AI bot based on their ML model. The team started building a "bot-wrapper," an API through which an ML model could interact with the Stall Catchers platform. The development took place during a time when I was actively collaborating with the institute and remotely attending many of the institute's meetings, such as the regular development meetings. The focus of one of these meetings in April 2021 was what the nonhuman actors in Stall Catchers could and should look like. The team quickly agreed that AI bots should be treated as similarly as possible to human participants, in part because the goal was not just to automate Stall Catchers but also to explore how "nonhuman agents" would participate in a "community with humans" in Stall Catchers (fieldnote Apr. 6, 2021). The approach was also deemed pragmatic because of certain dependencies in the Stall Catchers source code, such as the requirement that all players have a profile (fieldnote Apr. 6, 2021). Therefore, AI bots would be equipped with user profiles and points, and assigned a skill level. Once it was agreed that bots and humans would be treated almost equally in Stall Catchers, the discussion turned to how to introduce bots to human participants. It was not clear how participants would react to their new fellow AI players, since introducing AI bots to participate in a CS game alongside human participants would be a novelty. Hence, the goal was to "remain as neutral as possible with the bot and to hear from the Stall Catchers participants how they perceive the bot," explained Michelucci (fieldnote Apr. 6, 2021).

The first test bot named "Kaos" (Human Computation Institute 2021, 41:16), which had no actual underlying model to predict whether a vessel was flowing or stalled but would randomly select annotation answers, was introduced as a baseline. The second bot released on Stall Catchers was GAIA. Named by its creator Laura Onac, one of the winners of the ML challenge, GAIA is named after the second goddess in Greek mythology, the personification of Earth, who followed the god Chaos. Onac explained in an interview published on the Human Computation Institute's blog: "[s]ince [this bot] was the first one, we gave it the name of a Greek primordial deity—the great mother of all creation" (Vaicaityte 2021a). Additionally, GAIA could also be read as an acronym for "Gateway for Artificially Intelligent Agents," referring to the history of AI bots in Stall Catchers now starting (fieldnote Apr. 12, 2021).

In terms of Stall Catchers as a laboratory for new human–AI relations, the introduction of AI bots once again modified its participant–technology relations. And, as before, these relations did not fully unfold in the ways the team had imagined, designed, and implemented. For example, despite the team's prior discussions about the potential reactions of Stall Catchers participants to AI bots, they refrained from clearly defining the AI bot as a partner in the announcements to the participants, even though they aimed to encourage the formation of human–AI bot teams. The introduction of these artificial participants in Stall Catchers led not only to such teams but also to competitive human–AI bot relations, as I will show in the following. These would pose new challenges for the design of human–AI combinations in Stall Catchers.

GAIA's Debut

GAIA made its debut in Stall Catchers during the 24-hour Catchathon event on April 28, 2021, described in the introduction of this work. The goal of the Catchathon was to re-analyze a specific dataset that Stall Catchers participants had previously analyzed. This time, however, not only human participants would annotate the data, but GAIA would play alongside humans. The goal, as described by the Human Computation Institute in a blog post informing participants of the upcoming event, was to explore how well human participants and the AI bot would work together and to see "if we can do it just as well with our bot friend GAIA , as we did on our own!" (Egle [Seplute] 2021b).

During the final hour of the competition, when all participants were invited to a Zoom hangout with the institute, the team introduced GAIA as a new fellow participant who,[7] just like humans, was not perfect. When a participant asked if they could assume that all of the bot's answers would be correct, Michelucci responded by pointing out GAIA's human-like imperfection:

> Well, if it were a perfect learner, then we might expect that. But you know what it's like when you have, when you have many different teachers telling you different things, then you have to decide who you're supposed to believe, and this bot has had 35,000 teachers, and so somehow it has to integrate all of that—you know by 35,000 teachers I mean everyone who's ever played Stall Catchers. So, in principle, I definitely see what you mean, on the other hand, when you start to kind of, you know, how do they say that the devil's in the details, right?! (Michelucci in Human Computation Institute 2021, 15:02-25:40)

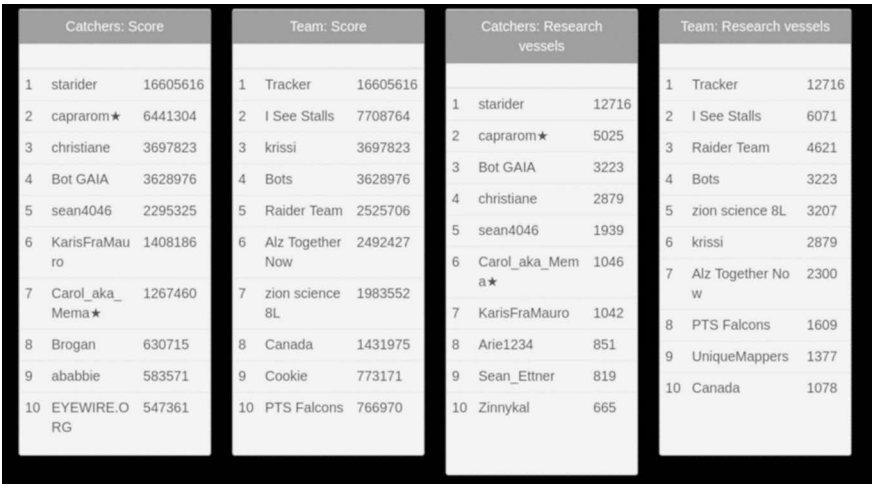
Here, while invoking one of the well-known metaphors of AI, that AI systems "learn" (and even extending it by referring to Stall Catchers participants as "teachers," humanizing GAIA as a student), this narrative simultaneously departs from common AI narratives or "myths," such as the understanding of AI systems as "coherent object[s]" (Bruun Jensen 2010, 21) that are neutral, acultural, and, therefore, infallible (Carlson and Vepřek 2022). Instead, GAIA's imperfection is acknowledged and turned into a supportive argument for the "wisdom-of-the-crowd" approach underlying HC development, according to which the combination of diverse answers from different information-processing approaches is considered particularly valuable.

Despite, or perhaps because of, the anthropomorphizing of GAIA with its imperfections, the bot's appearance on the Stall Catchers leaderboard still evoked competitive feelings, an outcome the team had not intended. This perception of GAIA as a competitor can be described as a "tacit consequence of an explicit design decision" (Forsythe [1996] 2001e, 99), namely, not actively defining or announcing the intended human–bot relations but choosing to include GAIA in the game and on the Stall Catchers leaderboards, and having it accumulate points. Once again, the play/science entanglements and how they shape the participant–technology relations in Stall Catchers become apparent. Mike

7 Because of its name and the reference to the goddess, GAIA was often referred to as "she" and somehow personified.

Capraro, a long-time active participant in Stall Catchers (known as a "supercatcher"), joined the live Zoom meeting during the Catchathon's final hour and described his perception of the bot: "I woke up, and that pesky bot was ahead of me by almost 2,000,000 points," he explained with a smile (in Human Computation Institute 2021, 29:19–29:24). "I'm pretty impressed it was, she was tooling along at about skill level 96 to 108, but then around 11:30, she got all the way up to 100 percent she was getting 116" (Capraro in Human Computation Institute 2021, 29:32–29:46). Participants can instantly gain between one and 116 points, with the latter being the maximum, in Stall Catchers scoring system for annotating a video. Capraro explained that he had been closely observing GAIA's gameplay over the course of the Catchathon. As mentioned in the introduction of this work, I got the impression from attending and observing the hangout myself that Michelucci's response to Capraro's experience with GAIA carried almost a bit of relief as he summarized his preliminary analysis of the bot: "GAIA is fast but not quite as skillful as [the best human participants]" (in Human Computation Institute 2021, 32:28–32:31). In the end, some human participants had even succeeded in beating GAIA, who came in fourth in the final ranking (see Figure 7).

Figure 7: Final leaderboards of the April 2021 Catchathon



Catchers: Score

1	starider	16605616
2	caprarom★	6441304
3	christiane	3697823
4	Bot GAIA	3628976
5	sean4046	2295325
6	KarisFraMauro	1408186
7	Carol_aka_Mema★	1267460
8	Brogan	630715
9	ababbie	583571
10	EYEWIRE.ORG	547361

Team: Score

1	Tracker	16605616
2	I See Stalls	7708764
3	krissi	3697823
4	Bots	3628976
5	Raider Team	2525706
6	Alz Together Now	2492427
7	zion science 8L	1983552
8	Canada	1431975
9	Cookie	773171
10	PTS Falcons	766970

Catchers: Research vessels

1	starider	12716
2	caprarom★	5025
3	Bot GAIA	3223
4	christiane	2879
5	sean4046	1939
6	Carol_aka_Mema★	1046
7	KarisFraMauro	1042
8	Arie1234	851
9	Sean_Ettner	819
10	Zinnykal	665

Team: Research vessels

1	Tracker	12716
2	I See Stalls	6071
3	Raider Team	4621
4	Bots	3223
5	zion science 8L	3207
6	krissi	2879
7	Alz Together Now	2300
8	PTS Falcons	1609
9	UniqueMappers	1377
10	Canada	1078

Source: ©Human Computation Institute 2021 (Egle [Seplute] 2021c)

This first experiment involving "AI participants" in Stall Catchers had produced a competitive relation between humans and AI bots. As a next step after this event, Michelucci explained in the final hangout that the team would now thoroughly analyze the data collected during the Catchathon, including the performance of all participants and the annotations of the dataset, "to see if the bot is good enough to help the research" (in Human Computation Institute 2021, 39:12–39:15). If so, GAIA would become a permanent part of Stall Catchers, though the details of GAIA's participation would still have to be worked out:

> [W]e're going to do this in consultation with others, with all our Stall Catchers players and community, to make sure everyone is comfortable with it. Whether we keep the bot visible on the leaderboard or whether it becomes something that's working in the background, we need to figure out where everyone's comfort level is. But the benefit of having a bot is that, again, it can play 24/7. If it's doing a good job, we could potentially start introducing other bots! (Michelucci in Human Computation Institute 2021, 39:21–39:53)

After the Catchathon, I reached out to some of the participants via the Human Computation Institute's forum to learn about their experiences with the bot. Most of the few participants who shared their impressions agreed that it was fun to compete with GAIA and described this first encounter as "relatively non-threatening" because GAIA was slow and inaccurate enough to be beaten by the best human participants:

> While her skill level was generally pretty good, she was not on par with our best catchers. That combination of speed and skill, I felt, made her a relatively non-threatening first encounter for those of us unfamiliar with bots as collaborators. I'm looking forward to seeing the next iteration, and how we can leverage AI to improve stall-catcher productivity. (Caprarom 2021)

A Stall Catchers participant with the username Christiane added that she hoped to get the chance to "work with her even more from time to time" (Christiane 2021). While Capraro had described GAIA as a "pesky bot" during the competition, Christiane could imagine working together with GAIA beyond the end of the event. Participant starider still perceived GAIA as a direct competitor but, at the same time, described the bot as an encouragement to focus even more during the Catchathon:

> Gaia was […] my biggest concern during the catchathon. The fact, she was capable of undertaking the task without having to take any rest, was the biggest advantage she had over us. That was enough pressure and encouragement to keep me going. If not for the initial hitches that prevented her from performing continuously during the event, I'm certain she would have beaten me. Did notice that her sensitivity level wasn't 100%. She should be at her best for the next event. It was fun and challenging, competing against her. (Starider 2021)

These impressions suggest that AI bots in Stall Catchers might not only accelerate the analysis of the data (if proven not to harm the scientific data quality) but also impact the human participants' own play practices. While GAIA was perceived as an annoying competitor during game-play due to its advantage of not having to take breaks, it was seen as a valuable resource in furthering the goal of Stall Catchers, which was speeding up Alzheimer's disease research in the laboratory.

Three Bots in Stall Catchers

The Human Computation Institute decided to run another bot experiment with the support of a grant to continue CS bots research, building on the experience and lessons

learned from the first AI bot experiment in Stall Catchers. This second experiment was scheduled for October 2021, just a few months after the introduction of GAIA, and emerged from a lack of new research data and not necessarily from a thorough analysis of the last Catchathon's results, as Michelucci had announced in the final hour of the April event. Additionally, the institute was expecting more participants than usual on the platform in October due to a collaboration with the company Microsoft, which had chosen Stall Catchers as one of its featured events in its annual "Giving Campaign."[8] Michelucci shared his idea for another bot study with his colleagues in an internal Slack channel:

> The Cornell Lab, which provides new biomedical datasets for analysis by Stall Catchers will not have new data ready for a while, and the current dataset is fully analyzed. We also have new players coming to Stall Catchers as early as tomorrow, and I'd like to have a new dataset running. For these reasons, I would like to run a new bot study using a previously analyzed dataset. We would not run it as a challenge for a particular time period, as with the previous event, but as normal ongoing data analysis. (Michelucci, Aug. 23, 2021)

Only a few weeks remained to prepare the "bot study."[9] Starting from October 6, 2021, three bots, including GAIA, joined Stall Catchers as participants, just like their human counterparts. The other two bots, named clsc2 and ZFTurbo, were based on the other two winning ML models from the 2020 Clog Loss Challenge and were built by the model creators using the same bot wrapper as GAIA. Similar to GAIA, clsc2 and ZFTurbo were named by their creators (Vaicaityte 2021b; 2021d). This time, the bots were active not just during a special event with a preset duration but for an indefinite period of time until the hybrid human/AI bot crowd had finished annotating a particular dataset. While the first experiment with GAIA aimed to understand the impact of an AI bot on the annotated data in general, the research question of this study was to investigate "how well [different] bots can work with humans and other bots to analyze Stall Catchers data" (Vaicaityte 2021a). Since each bot was based on a different ML model, they not only had different needs in terms of hardware and configuration, for example, but they also differed in their performance. These AI bots would be considered just like other human participants, so the team did not see it as a disadvantage that each had its own shortcomings and biases. The team articulated their research interest and the advantage of having differing AI bots in Stall Catchers in an interview setting in a blog post that preceded the new bot study:

> P. [Michelucci]: […] If all the bots always gave the exact same answer for the same vessel movies in Stall Catchers, then there wouldn't be any value in having more than

8 During this campaign, Microsoft donated money to Stall Catchers for every hour employees contributed to Stall Catchers (Vaicaityte 2021c).

9 The process of developing the second experiment can be described as following more of an "engineering ethos," as Forsythe described such practice-oriented approaches ([1993] 2001c, 44) rather than a theoretical, thoughtfully designed approach. The little time available to prepare the study meant that there was no time to discuss further the human–AI bot relation as had been announced in the final hour of the first experiment. Instead, the focus was on developing the technical side.

> one bot playing. But what we discovered is that the 50 bots created in the ClogLoss machine learning challenge are all fundamentally different in their design, how they are taught, and how they decide on their answers.
> L. [Onac]: Every bot is different and uses a different machine learning algorithm, each with their own strengths and biases. […] And even then no single bot is accurate enough so that we don't need the help of humans anymore. We are currently studying the performance of hybrid crowds, with both humans and bots. (Vaicaityte 2021a)

In this excerpt from the interview conducted by another institute member, Michelucci and Onac argue for the value of combining bots with different models and, hence, with different strengths and weaknesses. A few days after the publication of the interview, the bot study was launched on Stall Catchers. It took only a little time for AI bots, which now had a small bot icon next to their usernames, to consistently outperform the human participants on the leaderboard. Although some participants were motivating each other to beat the bots and congratulating their fellow human participants when they passed a bot on the leaderboard,[10] the bots not only had the advantage of not having to take a break but this time, they were even designed to be faster than individual humans. The goal was, in fact, for the three bots together to annotate data at roughly the same speed as all the human participants combined, which almost inevitably led to the bots topping the leaderboard. Observing the developments on the platform and leaderboards and noticing human participants' resentments toward the bots expressed in the in-game chat, the team tried to address this issue behind the scenes of the ongoing study, eventually informing participants about their motivation for enabling the bots' increased speed and inviting them to share their perspectives on the bot engagement.

However, even this explicit knowledge about the intended, programmed dominance of the AI bots did not lead all participants to accept their place on the leaderboard. Two human participants managed to regain the top two spots, possibly in part by "redeeming points," a Stall Catchers feature allowing participants to redeem accumulated points (fieldnote, Oct. 24, 2021). These points are accumulated in the following way: Participants initially receive only a few points for annotating each "research movie," for which there is neither an expert answer nor a computed crowd answer, since it is not yet clear whether their answer is actually correct. Once enough participants have annotated the video, an actual "crowd answer" is calculated. At this point, if a participant has previously annotated that video correctly, they receive additional points, which are added to their "redeem account." Participants can redeem these points when the redeem button turns green, indicating that points have been accumulated. If no points have been accumulated, the button is blue and states "No points yet. Check back!"

By employing this tactic, which, at the time, could not be performed by AI bots in the game, participants were able, at least for a short time, to pass the bots and regain their

10 I refrain from directly quoting participants' responses, as they sent these messages to their fellow human participants while contributing to Stall Catchers. The messages were, thus, not intended to be analyzed. I discuss this question of including chat data that was not primarily created for the purpose of my ethnographic research in Chapter 3.

lead in Stall Catchers. A participant had chosen the username ALZ_BOT_X, further blurring the programmed differences between human participants and bots. By the end of the month and near the end of the AI bots' involvement in Stall Catchers, the participant with the username starider had managed to earn more points than all the AI bots combined. This impressed the other human participants, who expressed their appreciation and pride for starider in the in-game chat and even surprised Michelucci.

Stall Catchers' participant–technology relations, as illustrated by this example of AI bots in Stall Catchers, intraverted. Initially, software prepared the videos for the participants to analyze, and participants took over the annotation task. At the same time, their competence was evaluated by computer algorithms. The latter weighted individual participant answers and combined them into the final crowd answers. Despite the team's careful design, participants came up with other practices and ways of engaging within these relations, for example, by exploring key combinations that could introduce shortcuts into the game (Chapter 5) and, therefore, changing them as well. Just as in the case study of Foldit, existing human–technology relations opened up the potential to train ML models based on the data annotated by participants over several years and to introduce AI bots into Stall Catchers, thus, transforming the participant–technology interplay. With the introduction of AI bots, even though the first two settings were experimental, the Human Computation Institute aimed to explore human/AI bot teams and how they could contribute to Alzheimer's disease research together. Participants and AI bots were treated almost equally by the system, allowing bots to earn points and climb the leaderboard. These new intraverted participant–technology relations increased the overall performance or speed of Stall Catchers, but, at the same time, they risked destabilizing it, at least slightly. Although participants appreciated the additional help of the AI bots in analyzing data to advance Alzheimer's disease research, the participant–AI bot relations were also competitive. Here, the play–science tensions described in Chapter 5 became apparent once again, as AI bots were highly appreciated for the scientific purpose of Stall Catchers. However, during gameplay itself, the introduction of AI bots as participants in Stall Catchers was perceived as unfair competition. This also illustrates the dynamic nature of participant–technology relations, which depend on context and perspective as Coleman writes for the example of hackers: "Hacker technical practices never enact a singular subject-object relation, but instead one that shifts depending on the context and activity. There are times when hackers work with computers, and in other cases they work on them" (Coleman 2013, 99). Similarly, participants sometimes worked with the AI bots toward the goal of accelerating Alzheimer's disease research, and, in other cases, they worked against and competed with them. In Don Ihde's terms, the AI bots became the "quasi-other" in Stall Catchers (1990, 107).

While the Human Computation Institute's team had deliberately designed GAIA to run on a smaller server and more slowly in the first experiment to compensate for the advantage of not having to sleep and eat, the three AI bots in the second study were programmed to work together to annotate roughly the same number of research videos as all the human participants combined. This resulted in the bots annotating much faster than individual participants. When the team noticed that participants in the second experiment perceived this as unfair competition, they responded by explaining the scientific purpose of the bots' speed to reassure them. In the end, some participants still found

ways to jump ahead of the AI bots on the leaderboard, at least for a short time, by redeeming points.

This example is one moment in the continuous intraversions of participant–technology relations. After the end of the second bot study, the institute analyzed the results of the challenge to see how well different human–AI combinations had performed and which ones were most promising. This also raised the question of how, for example, decision-making power should be distributed across these sociotechnical systems in the future to facilitate human–AI bot teams and not necessarily competitors. Ideas included having separate leaderboards for bots and human participants. In the beginning, the introduction of AI bots into regular Stall Catchers play was intended and anticipated to transform the human tasks into more challenging or interesting ones, as the AI would take over the easier videos, leaving participants with the "more interesting" ones. The institute explained in a blog post providing an update on human participants and AI models in Stall Catchers: "If and when these new bot catchers join the game, regular catchers will see less boring (easy) vessels and get a higher percentage of stalls to look at – the more interesting ones! And the dataset progress bar will hopefully move twice as fast!" (Egle [Seplute] 2020b). In line with the Human Computation Institute's oath not to use humans when machines can perform a task, AI bots *should* indeed take over this part of the analysis if their accuracy meets scientifically required data quality standards. The ethics of automation are guiding Stall Catchers' developments here. This could interfere with the meanings that Stall Catchers has for some participants as a practice and means of coping with everyday life marked by Alzheimer's disease (see Chapter 5) or as a pastime, which could then be sidelined in this regard. However, rather than simply replacing human participants, the institute's researchers ultimately aimed to change the existing tasks in Stall Catchers, including adding new ones, and to develop entirely new HC systems for humans to tackle meaningful problems that AI cannot yet solve. The Human Computation Institute's blog post "Stalls, machines and humans—an update" hinted at such potential new projects: "[W]e have new projects on the horizon with very different and quite interesting data where we will need to produce a similarly huge dataset to help teach machines. These new projects will address other disease research" (Egle [Seplute] 2020b). In this way, as in the case of Foldit, not only the tasks but also the purpose of the system itself are constantly changing with the intraversions. Stall Catchers must be adapted to stay at the edge of computational AI capabilities and to continue to legitimize HC's imaginaries.

Reconfigurations of Participant–Technology Relations

Tracing the human–technology relations in Stall Catchers and Foldit over the historical development and everyday unfolding of the projects has shown how these are neither completely predetermined by the systems' designs nor static in nature but continuously evolve and transform within and along the development of HC systems. In the example of Foldit, these intraversions unfolded from purely computational attempts to tackle the protein structure prediction problem, to participants manually manipulating protein structures with the help of algorithmic tools and, eventually, to symbiotic relations

between human participants and automated recipes, as well as the AI system AlphaFold. In Stall Catchers, relations intraverted from humans as assistants to the computational system, itself a "preprocessor" to humans and evaluator of their answers, to participants as teachers of ML models, to human–AI pairings working in both collaborative and competitive relations.

As discussed when introducing the intraversions concept, Hutchins' *Cognition in the Wild* analyzed naval navigation as distributed cognition, tracing how the introduction of new tools not only facilitated the cognitive processes involved in navigation but also presented users with different problems to solve which required different sets of abilities and their organization (1995a, 154). In a similar way, participants' tasks, practices, and forms of engagement within the HC-based CS systems were transformed, and their relations reconfigured not only by the addition of each new feature or automated tool but also through new potentials arising from within the human–technology relations themselves, by participants seizing timely moments (Mousavi Baygi, Introna, and Hultin 2021) and the software's affordances or object potentials (Beck 1997, 244). As such, changing practices and relations within the systems also led to renegotiations of responsibilities across the sociotechnical systems. And, as shown earlier, with these intraversions, even the purposes of the systems themselves are in continuous motion.

Tracing human–technology relations in HC, thus, reveals how these relations are in constant intraversion. These processual forward movements and shifts emerge not only from developers, designers, and future visions but also from the existing human–technology relations and practices. Participants' requests to developers and play practices aligned with the flow of algorithms, opened up new possibilities that both enable and constrain *how* human–technology relations unfold in the future.

As I argued in Chapter 4, researchers in the field of HC typically refer to *participants* when talking about human-in-the-loop computing. However, HC systems are not constituted solely by participant–technology relations, even if these are the focus of HC endeavors. Rather, developer–technology relations, the team–technology relations, or the researcher–technology relations, to name the most prominent human–technology relations besides participant–technology relations, are also an integral part of the formative relations of HC-based CS game assemblages. Like participant–technology relations, they also change and intravert as projects evolve, often in mutually reinforcing ways. In what follows, I turn to the study of intraverting researcher–technology relations in Stall Catchers, illustrating this dynamic with the example of the researchers' infrastructuring of and working with the data pipeline.

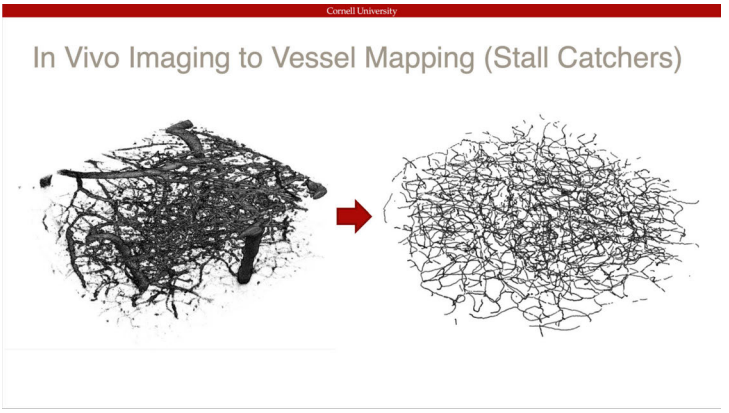
Extending the Loop: Intraversions of Researcher–Technology Relations

The biomedical engineering Schaffer–Nishimura Lab studies the role of stalls in capillaries of genetically engineered mice as part of their Alzheimer's disease research. As discussed earlier, they investigate how these stalls occur and how blood flow can be restored by exploring different treatments. The researchers use multiphoton microscopy to image the brains of living mice. By taking images of successive layers in a specific brain region

and combining the individual images into so-called TIFF stacks,[11] simply referred to as image stacks, it becomes possible to manually scroll through these images layer by layer through time, and thereby through the depth of the brain, creating a somewhat fluid view of the 3D structure. This representation allows researchers to analyze blood flow and identify stalled vessels more easily. These created image stacks form what is referred to in the laboratory as "raw"[12] data, which were initially analyzed manually by researchers in the laboratory prior to the introduction of Stall Catchers (and, as shown in Chapter 7, sometimes still are today).[13]

It should be noted that from the very beginning of the imaging process and the creation of raw image data, "*the materiality of the process gets deleted*" (Law 2004, 20, emphasis i.o.). Similar to the rats Latour studied (Latour and Woolgar [1979] 1986), the following transformation steps are not manipulations of the materiality of something like rats or mice themselves but of representations produced by scientists in relation with technologies (Latour [1988] 1993; cited in Law 2004, 20). New representations of previous representations are created with each step in the data pipeline.

Figure 8: Data transformation in summary performed in the laboratory's data pipeline



Source: ©Schaffer–Nishimura Lab 2021

11 TIFF stacks are single files of the Tag Image File Format (TIFF) that contain multiple raster graphics images. A raster graphics image displays a two-dimensional image as a grid of pixels.

12 "Raw data," as Bowker (2008) has aptly put it, is an oxymoron. Data, in this example Alzheimer's disease research data, are produced as part of knowledge production and, as such, "need to be imagined *as* data to exist and function as such, and the imagination of data entails an interpretive base" (Gitelman and Jackson 2013, 3, emphasis i.o.). In this chapter, however, I use the term to refer to its usage in the field, specifically the data produced during imaging. This data is subsequently processed and presented on the Stall Catchers platform, where it contributes to Alzheimer's disease research in the laboratory.

13 "Manual analysis" in the example studied already referred to human–technology relations in that researchers analyzed imaging data with the help of software.

In order for Stall Catchers to contribute to the analysis of the Schaffer–Nishimura Lab's research data, the data (representations) had to be transformed into short video sequences analyzable by participants without prior training. This required the creation of an infrastructure, more precisely, a data pipeline, through which the data follows a "chain of translation" (Latour 1999): from the form of image stacks, the data travels through the data pipeline of the laboratory where it is cleaned, normalized, and a vessel map is created (see Figure 8; fieldnote, Jul. 27, 2021).

After a laboratory member has submitted the data to the Human Computation Institute and further translation has taken place, the data is finally ingested into Stall Catchers as short video sequences with a highlighted vessel segment, where it is presented to participants.[14] Infrastructuring, to build and maintain the pipeline, therefore, played a major role for the laboratory from the very beginning of the project. Since the complex process to be achieved by this pipeline, consisting of several individual steps of data cleaning, transformation, and preparation, could not be fully automated by the programmers, biomedical researchers had to intervene at different points to complete a specific step manually, or to correct errors made by an ML-model, for example. In fact, human–technology relations play an important role in many steps of the pipeline. Hence, with the start of the Stall Catchers project, the very processes and practices of Alzheimer's disease research in the laboratory changed, as did the corresponding human–technology relations. It both unsettled existing, established practices and led to the introduction of new tasks, challenges, and ways of engaging with research data. The data pipeline was seen as a means to speed up the analysis of Alzheimer's disease research data and free up researchers' time for other tasks. As biomedical researcher and laboratory member Leander explained in one of our conversations during my first research visit in Ithaca:

> I don't know if it's just in research or this lab, but you kind of go for something, and it's not quite ready yet, but it's like, oh, let's try this. And then it's like, oh, it's working, keep going! And then it's like: But we don't have … [these] steps to support it. […] it's like things are working well, and then when you try to grow it, you don't necessarily have the infrastructure in place to actually support it. […] It was like, oh, this idea is working really well. It's a good idea. [LHV: But who's going to] yeah, who's going to do and […] do we have the stuff in place to actually make a pipeline that's not going to end up … costing more time. Because the goal is for the citizens to be helping the science, not for the scientists who feel it's like […] an extra thing that it's, […], I don't know. But it's getting there. (Sept. 22, 2021)

To get there, as I observed during my field research in 2021 and 2022, the data pipeline turned from a means to an end in itself (Vepřek 2022a). Researchers had to shift their

14 The data that Stall Catchers participants analyze on the platform are the preprocessed, normalized images presented as short video sequences. Other processing steps performed at the laboratory serve to create a vessel map of individual vessel segments, which ultimately facilitates the preprocessing that takes place at the Human Computation Institute. Here, on the institute's cloud servers, the orange outlines highlighting individual vessel segments are computationally drawn and the corresponding videos, focusing on such circled vessel segments, are created (step six in Figure 9).

attention away from the goal-oriented work of generating new scientific results to the "functional" work of infrastructuring. In this process, the data pipeline, which was supposed to be a means to enable and facilitate the research in the background, itself became the focus of everyday working practices. The development of the fully automated infrastructure necessary to allow Stall Catchers participants to analyze data had not been completed with the launch of the platform. Instead, when I joined the laboratory in August 2021 to learn about the laboratory's perspective on Stall Catchers, this infrastructuring, not the analysis of the crowd's output data, was the focus of the laboratory's work around to the CS game. Infrastructuring, here, was accompanied by the hope to "get there" (fieldnote Aug. 24, 2021; Leander, Sept. 22, 2021). However, the pipeline and its steps were not introduced once and for all but instead continuously developed, modified, and improved to facilitate researchers' practices. Ultimately, the goal was to achieve a fully automated pipeline. Since infrastructures must not only be built but also maintained as the design and corresponding requirements of the downstream/overarching system change, they are never complete but in an ongoing state of becoming.

Along with these developments and in infrastructuring, researcher–technology relations unfold and develop. For example, the manual tasks humans have to perform together with software are constantly changing. To show how these intraversions emerge, the next subchapter aims to walk through the individual steps of the data pipeline and its human–technology relations at the laboratory, focusing on selected moments.[15] The following descriptions are not only about human–technology relations but also about how data is created, translated, and new data are generated as representations of existing data, which is itself an imperfect representations of "real" events and information. The steps to be discussed can be summarized as follows (see Figure 9): first, the generation of research data in the laboratory in the imaging process, followed by manual and automated preprocessing of the data to prepare it for the ML model DeepVess. DeepVess then processes the data before several automated post-processing steps are performed on the ML model's result. Next, the data is curated by researchers with software tools before being sent to Stall Catchers. Finally, Stall Catchers' data annotations are analyzed in the laboratory to close the laboratory–Stall Catchers platform–laboratory loop. The ethnographic description is followed by an analytical section focusing explicitly on the intraversions observed.

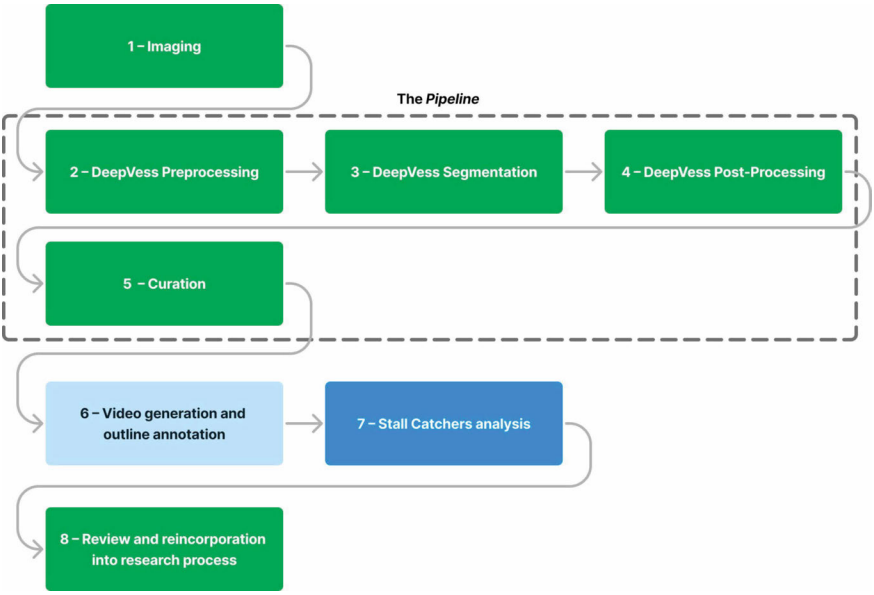
In order to focus on the HC system Stall Catchers and its human–technology relations, my analysis starts from the practice of *imaging*,[16] with an ethnographic note on

15 It is not intended to represent a comprehensive technical overview but aims to provide in-depth, microperspectival insights into some steps which have proven to be particularly interesting for this analysis, while only roughly touching on others.

16 It is in this practice that the human–technology relations associated with the digital research data, its transformations and representations within the Stall Catchers project become visible. For this reason, some essential tasks of the Alzheimer's disease research I observed at the laboratory will be left out here, such as the preparations preceding the imaging sessions, including craniotomies, preparing materials, injecting dyes, setting up lasers, or maintaining microscopes. Similarly, the end of the project or research process could be traced to the publications in scientific journals or even beyond. However, to answer my research question, I follow the digital imaging

"getting lost in the brain" in the context of how research data are generated at the Schaffer–Nishimura Lab. As will become clear, data generation in the *in vivo* imaging process of mouse brains with multiphoton fluorescence microscopy itself unfolds in various human–technology-mice relations.

Figure 9: Simplified stages of the dataflow. Green refers to stages performed at the laboratory, light blue refers to the processing stage at the Human Computation Institute, and blue refers to the Stall Catchers platform



Source: ©Vepřek 2023

Getting Lost in the Brain

During the first imaging session (step one in Figure 9) I observed at the laboratory in August 2021, researcher Benjamin explained that "it can be very easy to get lost in the brain." Sitting in front of two large computer screens and using the software ScanImage,[17] he was searching for a good spot in the mouse's brain to image (fieldnote, Aug. 18, 2021). A "good spot," Benjamin explained to me, was a region of the brain with many capillaries—which they analyze for stalls in their Alzheimer's disease research—and only a few larger vessels, which were not of interest for this particular analysis. "I don't know

data through the pipeline to the Stall Catchers platform and back, stopping at the analysis of the Stall Catchers results by the scientists in the laboratory.

17 "ScanImage is a software package for controlling multiphoton and laser scanning microscopes" (MBF Bioscience, n.d.). It can be customized toward specific research and microscope requirements.

where we are," he muttered (Benjamin, fieldnote, Aug. 18, 2021). Sometimes, it took several attempts to find such a spot in one of the brain hemispheres displayed on the right computer screen.

Meanwhile, the subject of inspection, the mouse, was out of sight behind a curtain to prevent light interference with the detection optics. Breathing regularly, as indicated by the breathing monitor on the right computer screen, the mouse lay beneath the objective, the optical component that gathers light from the mouse's brain, on a stage that Benjamin was controlling from his desk. Although out of sight, the mouse was still present, or represented, via the breathing monitor and the microscopic representations of its brain on the computer screen. Benjamin moved and adjusted the stage to a desired position with the computer program controls and the cursor traveling through the brain on the screen. On the left computer screen were two images representing the right and left hemispheres of the mouse's brain. These images had a magnification factor of four, much lower than the objective's magnification factor of 25 on the right screen. The right screen was divided into five frames, one of which was the breathing monitor. The other frames included the microscopic representation of the mouse brain, each displaying one of three different "channels." Each channel visualized different dyes used to color objects, vessels, and specific features of the brain. The remaining frame showed all channels merged together.

After a while, Benjamin found a good spot. He drew a rectangle to mark the region he wanted to image in the hemisphere on the left screen before returning to the magnified image on the right screen. Satisfied with the location, he moved the focus to the top of the brain to start the actual imaging process. After checking the settings, such as the number of slices to be imaged and adjusting the power of the laser to get a good view of the vessels, the actual data acquisition process began, imaging from the top down deeper into the brain. The laser "raster-scanned" the mouse's brain, moving in one direction, then turning, and moving back in the other direction to cover the entire region. In this process, it captured the emitted fluorescence and converted it first into electrical and then digital signals. Benjamin closely observed the process on his screen, where the completed frames incremented quickly. As the laser moved deeper, he occasionally adjusted the settings to maintain the best view and, hence, the subsequent image quality. Multitasking, he frequently checked the breathing monitor while saving the rectangle to find it later for the next imaging session with this mouse, which would take place in a few weeks. Depending on the experiment, mice were imaged several times over a certain period of time, usually several weeks. This first session would be the baseline imaging and, therefore, included finding the brain regions that would be used again in subsequent sessions.

The process ended once the laser had reached the set final depth and completed the final scan. The high-resolution images generated were stored as one image stack on the computer before Benjamin moved on to the next imaging spot. A total of six image stacks were created per mouse and imaging session. Depending on the experiment and research study, several mice were imaged—some receiving a specific treatment to be tested and others serving as control mice—in approximately three sessions, generating 72 image stacks (if four mice were imaged) per experiment. On average, a dataset sent to Stall Catchers contained around 170–190 image stacks.

Although the imaging session also included measuring blood flow and placing the mice back in their cages, here, I continue to follow the data created in human–technology–mouse relations as "raw" image stacks, the first step in the data pipeline. These 3D high-resolution images of the vasculature, including the shape and size of the blood vessels, are represented as connected voxels[18] in the digital representation and bundled into an image stack of around 500 images. In Latour's terminology, these high-resolution images are inscriptions produced by instruments and lie behind scientific texts (e.g., in Latour 1987, 64–70). They represent both time and space, or depth, and, in fact, differ in many ways from the short video sequences with highlighted vessels that are ultimately presented to Stall Catchers participants in their "virtual microscope." These raw stacks, for example, ideally include many capillaries but also other larger vessels, liposomes, or dura, making it difficult for the untrained eye to analyze individual vessels for stalls. The data then had to be cleaned, processed, and transformed in order to delegate the analysis to Stall Catchers and the crowd of participants and to create these short videos focusing on a single capillary to be analyzed; a complex process undertaken both by the laboratory, which handled the image data preparation, and by the Human Computation Institute, which then created the actual videos and calculated and drew the vessel outlines around them.

With this description of the origins of the raw research data that are ultimately analyzed in Stall Catchers in mind, I will now turn to their subsequent transformations or translations (Callon 1984; Latour 1999), which themselves involve new data generations, happening at the laboratory in the data pipeline.[19]

Processing and Translating Data

At the time of my research, the core process of the entire pipeline was centered around the ML model DeepVess, which was included to perform vessel segmentation. In the latter, individual blood vessels are isolated from tissue and other structures in the surrounding image region. Multiple pre- and post-processing steps had to be introduced because this model could not be run on arbitrary data; it performed differently depending on the quality of an input dataset. Both pre- and post-processing here refer to the data pipeline at the laboratory itself, as well as the processing that takes place before the data is sent to the Human Computation Institute. In the following, the analysis follows two scales of chronological developments, with the primary focus on the data and its transformation steps in terms of their order within the data pipeline and the secondary focus on the longer-term changes made to the data pipeline itself. While I begin with a focus on the state of the pipeline in 2021, I subsequently describe the changes that were made over the course of my research until late 2022.

18 Voxels can be thought of as volumetric (3D) pixels, depicting values on a grid in 3D space.

19 The term *data pipeline*, also used in the laboratory, creates the fiction of an *object through which data is sent*. While I use this term to describe the human–technology relations and data translations and generation steps, I understand the *pipeline* as an incomplete process that, along with its researcher–technology relations, is continuously changing and becoming.

The imaging data had to be preprocessed for DeepVess to perform as expected (step two in Figure 9). The raw images, for example, contain elements and parts of the brain vasculature that are not considered to contribute to the understanding of Alzheimer's disease and are, therefore, seen as "noise" in the research data. Dura mater,[20] for instance, could be visible in the images at the top of the image stack, taken from the upper layers of the brain, but DeepVess could not handle dura, and, additionally, it was not relevant to the scientists' research interest. Therefore, dura had to be manually removed from the images. In this process, researchers identified the image regions containing dura and drew the boundaries around these image regions. In this context, cleaning meant that the information in a given image region was "masked" by the researcher, i.e., overwritten with zero-value (black) pixels, thereby, making it invisible to both human vision and ML models.

Stalls only occurred in the capillaries (the small vessels). Therefore, large vessels were also not needed for the Stall Catchers analysis and would be a confounding variable in the overall analysis. For a long time, researchers had to manually mask these large vessels in the images before running them through DeepVess because the ML model's performance was negatively affected by them: "[T]he problem is not the capillaries. It does a great job with the capillaries. The problem is […] the larger vessels," explained researcher Leander (Sept. 22, 2021) in one of our conversations. DeepVess was often confused by large vessels and, as a result, misidentified parts of them as vessels. While masking dura was quite efficient and fast, masking all individual large vessels was (and remained) quite a time-consuming task:

> [Y]ou'll have a large vessel going down. And the best way we had at that time was to go through frame by frame and mask out the vessel. And you can't just do it straight because the vessel moves, you have to [, for] every frame, move the little thing. And there's usually multiple, and it's taking me hours. (Leander, Sept. 22, 2021)

Therefore, to improve their data cleaning and preparation work, the researchers modified or improved the data pipeline by developing an automasking algorithm to take over their task.

After these initial data cleaning steps, laboratory members actively pushed the data to the next automated data preprocessing step, which referred to "basically […] just normalization with the possibility of motion correction," as laboratory member Charles explained (fieldnote, Aug. 17, 2021). Normalization,[21] in this case, refers to adjusting the intensities of the image by mapping the values to the range between zero and one. A threshold value of 0.8 was set for a mask, meaning that any value above 0.8 is considered a logical one and included in the mask, and anything less is not included (Charles,

20 Dura mater describes the outermost layer of the membrane surrounding the brain and spinal cord, which provides protection for the central nervous system.

21 Following the feminist science and technology sociologist Hannah Fitsch and media studies scholar Kathrin Friedrich, who draw from literary scholar Jürgen Link's understanding of the "normal" (e.g., Link 2014), "[n]ormalization […] describes the processual steps required to constitute normality" (Fitsch and Friedrich 2018, 3; *cf.* Bowker and Star 2008).

Sept. 14, 2021). As Charles pointed out, this process raises the question of what is considered a capillary in the laboratory's research. It was not considered perfect but "operating fairly well" (Sept. 14, 2021) regarding the researchers' goal of identifying capillaries. This step was performed using the MATLAB programming language and environment (The MathWorks, Inc., n.d.), which provides various tools for image processing. This frame-by-frame image normalization took approximately two minutes per image stack and could be parallelized to process four stacks simultaneously (fieldnotes, Jul. 27, 2021; Aug. 17, 2021).

The "possibility of motion correction" referenced above refers to the fact that mice are active actors in Alzheimer's disease research whose actions and movements do not always conform to the scientific, or more specifically, the imaging process. If a mouse moved during the imaging session, the image quality was impacted and could include smears or waves caused by movement (fieldnote, Aug. 17, 2021). These movements could be minimal, as Michelucci explained to me during one of our interview sessions: "The mice are breathing. Their heart is beating, these introduce motion artifacts into the image stacks" (Jan. 21, 2021). These motion artifacts are described in the paper detailing the DeepVess algorithm as "one of the major challenges for 3D segmentation of *in vivo* MPM [multiphoton microscopy] images" (Haft-Javaherian et al. 2019, 5, emphasis i.o.). The laboratory implemented a motion correction algorithm based on previous work on an image registration tool (Thirion 1998; Vercauteren et al. 2009; cited in Haft-Javaherian et al. 2019, 5). However, at the time of my field research, this algorithm was not necessarily applied to every dataset as it could also worsen data quality, sometimes to the extent that further data analysis was no longer feasible (fieldnote, Aug. 17, 2021). This could mean, for example, that vessel outlines would not line up correctly due to cumulative errors introduced by motion correction (fieldnote, Jul. 27, 2021). The decision to apply motion correction was, therefore, based on the researcher's assessment of how they thought DeepVess would handle a specific dataset. In these researcher–technology relations, more of the decision-making power was on the side of the researchers, who determined the algorithm's role and the course of action. At the end of the preprocessing program, a so-called H5 file was generated, a data format that stores data hierarchically as multidimensional arrays. This file constituted the input to DeepVess.

Machine Learning for Data Segmentation

Having described the preprocessing steps performed before and *for* DeepVess, I now turn to the ML model itself (step three in Figure 9). The model was developed and introduced to automate the segmentation of vessels, which is important for the later analysis of the vessels for stalls. Stall Catchers participants should be presented with the complete segment of an individual vessel to determine if it is stalled (fieldnote, Aug. 16, 2021). Prior to using DeepVess, researchers manually traced and marked each vessel in the image stacks using the open-source image processing software ImageJ (National Institute of Health, n.d.). Manual segmentation of a 3D image stack took around 20–30 hours per stack. The

authors[22] state in the article introducing DeepVess that manual segmentation, thus, was not feasible since it "slows down the progress of biomedical research and constrains the use of imaging in clinical practice" (Haft-Javaherian et al. 2019, 12). Researcher Emily explained that manually tracing each vessel without the help of DeepVess would be "a colossal task" (Sept. 8, 2021). Adding up the number of images to be analyzed and vessels to be traced illustrates this problem: In each experiment, several mice, for instance three mice, had to be imaged in three phases of imaging with six image stacks per session and mouse, adding up to 18 image stacks per mouse and, in this example, 54 image stacks in total. If each image stack contains around 500 images, this would result in 27,000 images to analyze, each containing multiple vessels. Hence, DeepVess' lead developer Haft-Javaherian's "aim throughout his tenure with EyesOnALZ [the name of the former overarching project of Stall Catchers] [was] to replace himself with machine automation," states a blog post by the Human Computation Institute (Egle [Seplute] 2019).

Haft-Javaherian developed a convolutional neural network (CNN) to automate the segmentation of 3D vessels. CNNs are a form of deep neural networks, sometimes also called DL. Deep learning is a specific form of ML (Goodfellow, Bengio, and Courville 2016, 95). To give a slightly simplified account of what neural networks are, they combine different mathematical operations whose composition is described by a directed acyclic graph (Goodfellow, Bengio, and Courville 2016, 163), often simply a linear chain of multiplications and subsequent application of so-called "activation functions." The different components in the chain are also called "layers," while the "length" of the chain describes the depth of the model (Goodfellow, Bengio, and Courville 2016, 163). When there are two or more layers, the network is typically referred to as "deep." The CNNs are a particular type of "neural network for processing data that has a known grid-like topology" (Goodfellow, Bengio, and Courville 2016, 326), especially when there are "spatial invariants" in the data, i.e., "local" patterns that can occur independently of their "global" location. They are, thus, particularly successful in computer vision and image data processing (LeCun, Kavukcuoglu, and Farabet 2010; Albawi, Mohammed, and Al-Zawi 2017) and owe their name to the mathematical linear operation called "convolution," which is performed on matrices (Goodfellow, Bengio, and Courville 2016, 321; Albawi, Mohammed, and Al-Zawi 2017, 1). Mathematically speaking, the input and output of the individual layers and the overall model are essentially sets of matrices (or, generalized to higher dimensions, "tensors") or, in terms of code, simply "arrays." The outputs of intermediate layers in such a network are also called "feature maps" (LeCun, Kavukcuoglu, and Farabet 2010, 1). In the example of the 3D multiphoton microscopy imaging data, both inputs and outputs would be 3D tensors, each represented as a three-times nested array. The laboratory implemented their models using Tensorflow (Martín Abadi et al. 2015), a popular open-source software library for AI and ML applications for the Python programming language (Haft-Javaherian et al. 2019).[23]

The paper introducing DeepVess describes it as a system consisting of preprocessing steps (as described above), the actual segmentation with the model, and post-processing

22 The paper was authored by laboratory members and researchers of the Meinig School of Biomedical Engineering at Cornell University, Mohammad Haft-Javaherian et al.

23 The precise architecture of the laboratory's CNN is described in Haft-Javaherian et al. (2019).

steps (Haft-Javaherian et al. 2019, 3). However, during my field research at the laboratory, researchers referred to DeepVess as the ML model performing the segmentation and discussed its pre- and post-processing separately. Therefore, I follow the laboratory's usage, referring to the actual model itself as DeepVess.

DeepVess proved to achieve better accuracy computationally on the task of vessel segmentation than any of the "current state-of-the-art" (Haft-Javaherian et al. 2019, 2) computational approaches. Moreover, it took only ten minutes to calculate the segmentation of the images compared to 30 hours of manual work (Haft-Javaherian et al. 2019, 12). Once the model was trained, researchers could simply run it without thinking about DeepVess' internal steps. Laboratory member Charles explained that he had "never actually looked at [the DeepVess source code] file" (Sept. 14, 2021). When he eventually did during one of our meetings, he "realized, oh, it's only 276 lines long. It's not very long for such a program" (Charles, Sept. 14, 2021).

To actually process the data and perform the vessel segmentation, a researcher would invoke it via a simple command in a terminal. Once this process was complete, a researcher would take the resulting output file and similarly initiate the post-processing on it (fieldnote, Aug. 17, 2021). However, although this process flow was very clear and straightforward, it did not always go as designed, typically due to software errors interrupting the segmentation of individual samples. These errors were often related to the fact that DeepVess eventually had to be updated (fieldnote, Aug. 17, 2021). DeepVess had been introduced at the beginning of the Stall Catchers project and its maintenance involved, for example, updating versions of software libraries on which it depends, such as TensorFlow, from time to time. At the time of my field research, however, this maintenance was lacking and the code, or the libraries it used, were no longer up to date. Part of the problem was that none of the current laboratory members were deeply familiar with DeepVess and its codebase (fieldnote, Aug. 10, 2021) and, as researcher Leander explained to me, they had difficulties "decipher[ing]" (Sept. 22, 2021) it after the developer left. "So, it's been hard for us to […] improve it. We also don't have someone who's as knowledgeable, I think, in machine learning" (Leaner, Sept. 22, 2021). During my field research, DeepVess was sometimes referred to as a "black box" or "magic" by researcher Anna (fieldnotes, Jul. 27, 2021; Aug. 10, 2021). This is a common narrative used to convey praise for the ineffable capabilities of technology, or ML in this specific case, while simultaneously emphasizing its inscrutable and unknowable nature (e.g., Elish and boyd 2018, 63).

To get around the error messages that kept appearing, a workaround was found that allowed data to be pushed through DeepVess without interruption. Overall, Isabel summarized, DeepVess worked well "as long as there's stuff in the frame" (fieldnote Aug. 10, 2021). As it turned out, however, there was not always "stuff in the frame," and DeepVess did not always perform equally well, so the resulting segmentation was not always of the quality desired. The results could "contain some segmentation artifact such as holes inside the vessels, rough boundaries, or isolated small objects" (Haft-Javaherian et al. 2019, 8) and "[i]mages from deeper within the brain tissue […] suffer[ed] from more segmentation errors" (Haft-Javaherian et al. 2019, 14). Additional automated post-processing steps were introduced to mitigate such errors (see Haft-Javaherian et al. 2019, 8).

Automated Post-Processing

The post-processing (step four in Figure 9), as I learned during my field research at the laboratory, was significantly more complex than the preprocessing steps used to prepare the data for DeepVess (fieldnote, Aug. 17, 2021). The process, again implemented in MATLAB, consists of three main steps, which I briefly describe below. Like the segmentation process, post-processing is triggered manually by a laboratory member. The first step, Isabel described,

> [is] just the cleaning up […] of the segmentation. So, what […] we get out of DeepVess […] got lots of noise in it. So, there's, first a cleaning app and that's just smoothing the surfaces […]. We do that twice just to make it smoother, I guess. Fill in any single holes and then get rid of any single voxels that are left isolated. (Sept. 14, 2021)

Here, images are "re-filled" and boundaries "cleared" in order to recreate the shapes of the vessel segments as they appear, or would have appeared, in mouse brains, even though this digital representation never recovers its original but is practically a new image drawn on top of the already processed representation. Here, it becomes clearly visibly—not only in the post-processing but, in fact, from the beginning of the imaging process—that the object of analysis, in pursuing the aim of revealing more about its origin, is further alienated from its "original" form with each additional step.

In the next step, these new data representations are augmented with what the researchers call "skeletons." This means that the foreground regions of the stack are condensed into a "skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels" (Fisher et al. 2003).[24] The resulting structures are thin, single-voxel-wide paths creating a 3D skeleton that traces the structure of the underlying vessels (fieldnote, Jul. 27, 2021). Due to the skeletonizations, the images will also include "background noise" (Charles, Sept. 14, 2021) because "any little scattered bits will end up with a little tiny spot on them or a little line, lots of little bits of mess everywhere" (Charles, Sept. 14, 2021). Therefore, additionally, Anna explained to me in our Zoom meeting as we went through the code line by line on the shared screen, walking me through the individual algorithmic steps:

> [T]o remove any background noise and non-vessels, group the voxels together into connected components, […] that's what this […] function means. It basically says […] well this is for anything that is just off on its own and if it's off on its own, then just get rid of it. […] So, it just counts the number of voxels involved in each connected component. And […] there is a whole bunch of little things on their own and it just pulls them out. (Sept. 14, 2021)

Once the background noise has been reduced, the program shifts its focus to the actual skeleton, which has "lots of noise, there's gonna be gaps in it and stuff as well" (Anna, Sept. 14, 2021). Anna and I were now looking at an example image of DeepVess' output on

24 This process is applied at the level of voxels, not pixels, because the stacks are 3D structures.

the computer screen: "[y]ou can see down here it looks connected and it ends up being broken at some point" (Anna, Sept. 14, 2021). The next step, therefore, is to connect the vessels across these gaps, which may actually contain stalls (fieldnote Jul. 27, 2021), by expanding each voxel in the mask using a sphere of a specified radius. Anna explained that they "just inflate everything and that will cause things to then merge across gaps. So, if there […] is a stall somewhere and that ends up with a break, this will end up dilating across and they'll, the spheres on either side of the stall will end up colliding and they'll join together again." (Sept. 14, 2021)

This step of connecting the inferred vessel structures was followed by another iteration of the previous steps of smoothing, filling holes, and skeletonizing again to improve the result (Anna, Sept. 14, 2021). The individual vessel segments in the new skeleton were then reviewed, focusing on their length. If the number of voxels connected in a segment was too low, the segment was removed (fieldnote Jul. 27, 2021). In other words, connected voxels are only considered valid vessel segments if they reach a minimum length.

In the next automated post-DeepVess processing step, segments that were wrongly separated but *actually* formed one vessel should be reconnected. Here, the laboratory built upon and extended an approach described in previous research (Schager and Brown 2020), and the endpoints of vessel segments formed the focus of the operation. Suppose two endpoints of different segments are within a certain distance and "point at each other" at a certain angle. In this case, they should be connected because they were probably separated by mistake. At this point, segmenting and mapping the vessels in the network becomes a mere geometric, mathematical problem to be solved. This reconstruction step was added later and not included in the initial automated post-processing. Similarly, the computer code for most of the other steps had changed from the original implementation. New versions of MATLAB, for example, were released over time, including new functions that could be used to replace or optimize those that the DeepVess developer had previously programmed from scratch. As Isabel explained, "a lot of the functionality is now built into MATLAB so … it simplified things a great deal" (Sept. 14, 2021).

After connecting separated segments, the previously mentioned step of masking large vessels, which in the past was manually performed by researchers before sending the data through DeepVess, was now performed automatically.[25] Similar to the preceding steps, the problem is translated into a geometric problem in which the vessel's radius is compared to a "logical sphere" (Isabel, Sept. 14, 2021) of a certain size (fieldnote Jul. 27, 2021). According to the researchers' understanding, if the vessel's radius is smaller than that of the sphere, it classifies as a capillary. If, however, it is larger than the sphere's radius, it exceeds the size of capillaries and can, hence, be removed from the images by deleting its centerline/skeleton (Isabel, Sept. 14, 2021). Finally, a last round of skeletonization follows "just to make sure that there's nothing funny happening" (Charles, Sept. 14, 2021), junctions are removed, and the previous step of removing segments not

25 Therefore, the data were normalized regarding the masked volumes and compared to the "original image"—with everything that is to be masked replaced with a logical zero, and everything within a sphere are logical ones (Isabel, Sept. 14, 2021). Masking dura, by contrast, was still performed by researchers but at an even later stage in the data pipeline (see below).

meeting the required voxel count is repeated.[26] This yields the final skeleton, "which is the XYZ [coordinates] of all of the connected voxels for each segment and the number of objects in that is the number of segments […] in the skeleton" (Isabel, Sept. 14, 2021), from which the output of the automated post-processing is created.

In the first years of the Stall Catchers project, this post-processing was the final step in the data transformation process before a laboratory member would take the resulting data and sent it to the Human Computation Institute. However, while DeepVess took over the overall task of manually creating a vessel map, it did not always work as intended, introducing errors that the post-processing algorithms could not smooth out or repair:

> [DeepVess] obviously fails because […] our imaging data is very heterogeneous and sometimes we have very nice images, sometimes we have images that have a ton of noise, a ton of autofluorescence, and so DeepVess will pretend or will … DeepVess doesn't do a very good job in excluding things that are not vessels. So, it will detect things [that] are not vessels and will say, "Oh this is a vessel," but it's not; it's like a random thing that is autofluorescing. So, and then if we don't correct that, it will go into Stall Catchers, and then the users will be like, "Oh, this is not a vessel, this is like a random thing," and then they will flag the movie. […] [W]e can eventually see which movies are bad [= flagged], and there are a lot of bad movies. But then, at the end, because DeepVess doesn't really recognize a lot of the capillaries, […] the result that we get back from Stall Catchers … we get very few stalls. And we get so few that it's just not possible to work with that data. (Emily, Sept. 8, 2021)

In our conversation during my first research stay at the laboratory, Emily described not only the failings of DeepVess but also their reverberations through the Stall Catchers project and how they led to the researchers' inability to "work with that data." Stall Catchers allowed participants to "flag" videos if they could not analyze them due to ambiguous vessel segments or unclear, or "grainy" images (e.g., Eliza, May 18, 2020; Asher, May 20, 2020; Daan, May 26, 2020). However, researchers needed to know the correct number of vessel segments in an image stack to report on the impact of stalls and learn from the experiments. If DeepVess did not recognize a certain amount of the capillaries in the images, the number of detected vessels and stalls were incorrect. Therefore, despite the extensive automated post-processing put in place to correct many of the errors introduced by DeepVess, it still did not produce the required quality. While some vessel segments were drawn where they should not have been, others were separated or not detected for "convoluted reasons" (Isabel, Sept. 14, 2021; fieldnote Aug. 24, 2021). Although the post-processing steps described were "supposed to compensate for potential breaks in the vessel[s]" (Leander, Aug. 16, 2021), this did not "always work," as Leander noted:

26 "Junctions" or "branch points" refer to voxels with more than two neighbors that are logical ones (Charles, Sept. 14, 2021). These points get subtracted from the skeleton to receive the individual segments. By, once again, connecting the endpoints of the then isolated segments, "a long list of all of the connected voxels and […] every single connected segment becomes its own […] object in a structure" (Charles, Sept. 14, 2021).

> [W]hen we send [the data] to Stall Catchers, we want it to include the whole segment even if there's a stall because that's how they find whether or not there is a stall. And if it's a pretty big stall, [DeepVess] would make two vessels, which is not what you want. You want it to send the whole thing so … part of the DeepVess post-processing [is that] it will […] expand the segmentation so that it overlaps across any potential stalls and then it comes back as one segment. But that doesn't always work. (Leander, Aug. 16, 2021)

Data quality varies due to experimental conditions, the movement of mice during imaging, DeepVess, and the algorithmic processing's performance, along with the interference of other "things" in the images that float around and can be misidentified as parts of vessels by DeepVess because it does not distinguish between vessels and other bright spots (fieldnote Aug. 10, 2021). As Emily explained (see above), the quality of images differed from experiment to experiment and from imaging session to imaging session. Moreover, at the time of my field research, DeepVess had been trained several years earlier. "The neural network […] was trained on data that's older now, and things have changed a little bit," explained Leander (Sept. 22, 2021). In fact, at the time of my second research visit, the model was performing better on bad data than on data considered good quality due to the data on which it was trained (fieldnote, Oct. 25, 2022). Since then, not only has the computer code been continuously updated but so have the experimental settings, such as the microscopes, which influence the imaging data generation and, hence, the data run through DeepVess (fieldnote Oct. 18, 2022). In addition, the laboratory's PI Schaffer stated that four to five years are an "eternity in AI" and that there are already better libraries and models that could be used (fieldnote, Oct. 18, 2022). Retraining the model with current data was, therefore, discussed as an important measure but had not yet been done at the time of my research.

Together, these contingencies, changes, and data fluctuations led to varying data results at the end of the automated data processing. In the first years of the Stall Catchers project, however, the laboratory's focus had been on validating the CS approach to ensure that the crowdsourced analysis would meet the required accuracy. Only after that did they concentrate more on the data pipeline and improving the quality of the data sent to Stall Catchers. Until then, even when Stall Catchers participants accurately classified the data presented, these might have included broken or incorrectly identified vessel segments, or the image quality was sometimes not good enough to classify an outlined vessel. Following the principle of "garbage in, garbage out" (fieldnote, Oct. 22, 2022), the classified data returned to the laboratory from Stall Catchers could then also be difficult for the researchers to interpret because if DeepVess incorrectly separated vessel segments, Stall Catchers participants would sometimes separately classify videos of segments that actually belonged together (fieldnote, Aug. 10, 2021). Reflecting on the Stall Catchers developments from the laboratory's perspective, Anna saw bad data quality as the main problem from the beginning (fieldnote, Oct. 25, 2022).

"[W]e really hadn't anticipated how much the preprocessing […] was going to be necessary, […] and that was just something that we couldn't have predicted from […] the images and the data," explained PI Nishimura (Dec. 07, 2021) in our interview. In fact, at the beginning of the project using HC to speed up the analysis of the laboratory's Alzheimer's

disease research data, the plan had been to create two CS projects: Stall Catchers, to analyze the vessels, and another project to identify the vessel segments. The platform for the second project could be based on the 3D puzzle game and CS project Eyewire (Seung Lab, Princeton University, n.d.), in which participants help map neurons in the brain. This second project was never realized, however, because the laboratory and Human Computation Institute decided that "the blood vessels were pretty straightforward enough that we thought we could just automate that from the beginning," Schaffer explained (Dec. 07, 2021) in our interview. The assumption had been that the vessel segmentation could be fully automated: "But, honestly, I mean, you see, we're still struggling with getting […] the final details about the automation right and it still takes quite a bit of us manually intervening […] in order for us to have confidence in the outcomes" (Schaffer, Dec. 07, 2021). As it turned out, automating the vessel segmentation with DeepVess did not just make the researchers' work easier but also introduced new tasks and problems to be solved manually.

An additional process was introduced to improve the quality of the data sent to Stall Catchers. This process was called "data curation" in which the researchers' role was to "see if DeepVess did a good job and edit what it did not do well" (Sean, fieldnote, Nov. 4, 2022). This change also brought about new researcher–technology relations. My field research in Ithaca coincided with a crucial time regarding the data pipeline, as the researchers were in the midst of implementing this new data curation process. To have a solid, smoothly functioning infrastructure to rely on, starting in early 2021, they had been focusing on improving the data pipeline by introducing new steps and tools to facilitate the researchers' data preparation tasks and improve the quality of the data sent to the crowd. They were testing and improving a new customized editing tool for researchers to review the automated processing results (fieldnote, Nov. 4, 2022). The new editing tool "to clean up [DeepVess'] results" (fieldnote, Aug. 10, 2021) was also implemented in MATLAB. In a meeting of the laboratory in July 2021 about the data pipeline, Charles described the manual curation as an "opportunity to ensure stall count from Stall Catchers [was] directly applicable to experimental results" (fieldnote Jul. 27, 2021). With this new step, researchers now had to clean the images before sending them to the crowd for analysis.

This move introduced a new intraversion within researcher–technology relations in the data pipeline. The supportive work around the ML model that researchers had to do became tedious and temporarily even more time-consuming (at least, that was the experience of some researchers) than manually annotating stalled vessels from the start.

Data Curation: Manually Intervening and Editing

After the results of the automated processing described on the previous pages were saved in a laboratory-wide shared repository for processed data, researchers could access the data to edit it in the new curation tool, which allowed them to review the results of the automated processing steps and decide whether to omit, accept, or adjust them (step five in Figure 9). The curation tool was implemented as a MATLAB program, but unlike previous tools that were effectively just scripts run via a terminal, this tool featured a simple UI. This UI included a large image frame for the data to be edited and a collection of menus, settings, view options, and controls for scrolling through the image stack. Data

was displayed in the tool as black images with white vessels and a few white spots scattered across the frame. The vessel mask depicting the individual vessel segments, i.e., the segmentation created by DeepVess, was layered over the white vessels in red. After loading an image stack into the program, researchers could choose how many images they wanted to view at a time. Multiple consecutive image slices from the stack were then accordingly projected into a single condensed image, with the top slice being considered the image currently displayed, which allowed them to go through the image stack faster. Vessels in the currently displayed image were colored light red, while vessels in the other projected image slices were colored dark red. Some white lines were not colored in red. Researcher Leander explained that DeepVess had not identified these white lines as vessel segments. Sometimes, the resolution of the vessels was too low, or they were not bright enough, making them difficult to identify not only for the AI model but also for humans analyzing the images later. In this case, Leander continued, it was better to omit vessels because "you don't want to include vessels that cannot be analyzed" (fieldnote, Aug. 16, 2021). Most of the time, Leander "agreed with" DeepVess' results (fieldnote, Aug. 16, 2021).

After having "cut" the stacks to remove low-quality images and dura, researchers had to perform two main tasks in the editing tool. These were to reconnect vessel segments that had been incorrectly separated and to remove segments that DeepVess had wrongly identified as capillaries. While Sean usually took several passes through the image stacks, going through the projected images from the top of the image stack to the button and checking for broken or redundant segments and segments to connect separately (fieldnote, Nov. 14, 2022),[27] he emphasized that others would have their own routines for editing the data.

These two main steps of connecting vessel segments and removing falsely identified vessels allowed the researchers to improve the vessel map created by DeepVess. However, it did not allow them to retroactively include vessels that DeepVess had missed. Even though Leander considered it to be better to omit vessels that had not been identified by the ML model and could not be analyzed anyway, some vessels that had been missed could be analyzed, and he would have preferred to be able to add them manually after the fact. Some researchers, such as James, therefore, found the curation process occasionally frustrating because he would see vessels DeepVess had missed but was unable to correct the data, and had "to live with knowing that the data is not perfect" (fieldnote, Nov. 16, 2022). In addition to this discontent, editing also consumed a lot of time. In 2021, it took Leander about three hours to curate a single image stack (fieldnote Aug. 16, 2021). Although researchers agreed that this editing was necessary to get the total number of vessels in an image stack, which was subsequently needed to know what percentage of

27 In the first iteration, he focused on removing noisy segments which DeepVess often incorrectly classified as vessels, or large vessels that were overlooked in the automated preprocessing (fieldnote, Nov. 14, 2022). Once he arrived at the bottom of the image stack, he scrolled up again, checking for additional segments to be removed again before starting the second pass in which he focused on connecting segments (fieldnote, Nov. 14, 2022). Similar to the automated process of post-processing, researchers would focus on the endpoints of the vessel segments to connect broken vessels, this time manually (fieldnote, Aug. 16, 2021).

all vessels were stalled in their experiments, it was still perceived as tedious and sometimes described as a "misery" (fieldnote Aug. 16, 2021). "I even feel like this is more tedious than [manually counting stalls] cause I've been just manually counting stalls, but this is three hours per stack versus maybe … a little bit over an hour for manual," explained a researcher in a meeting related to the Stall Catchers pipeline (fieldnote Aug. 16, 2021). It is worth noting that this comparison of the work they now had to do in the curation process versus manually analyzing stalls did not include the additional 20–30 hours it would take to also segment a stack manually.

At that time, in late summer 2021, the goal was to improve the curation process so that it would take no more than an hour per image stack. The goal was to do "anything […] to make it faster" (fieldnote, Aug. 24, 2021). Laboratory members worked together to improve the tool by providing and implementing feedback. By the time I left Ithaca at the end of October 2021, researchers were confident that this goal would soon be achieved, with the process improved to the point where it would no longer be perceived as painful. Once the curation process and, hence, the data pipeline were improved, Stall Catchers could be successful not only as a game but also "successful from [the researchers'] end" (Leander, Sept. 22, 2021) (*cf.* Chapter 5). The problem, according to Leander, was that they "didn't have […] the infrastructure in place to support it being successful. So, if it's successful, […] we need to figure out […] how to do that. So, I think it's like almost there in terms of being successful from our end" (Sept. 22, 2021). Once "there"—which, in an August laboratory meeting, was defined as one hour of manual work to prepare the dataset—they could start focusing on other things again (fieldnote Aug. 24, 2021) and the data pipeline could fade into the background of Alzheimer's disease research processes.

However, when I returned about a year later, in fall 2022, the laboratory was still working on improving the data pipeline. It was not yet finished, Nishimura explained (fieldnote, Oct. 21, 2022). There had only recently been a new round of discussions about problems in the curation tool, and the understanding that they would be "there" soon had shifted. The data pipeline was now seen more as an "ongoing process," as Sean described it (fieldnote, Nov. 14, 2022), or as a "work-in-progress" (Emily, Oct. 18, 2022). Working on the data pipeline was understood as a "long-term investment" since the pipeline could be used for different data and different aspects of the data beyond the Stall Catchers project (fieldnote Oct. 21, 2022).

Compared to the previous year, however, the PIs stated that they were now in a much better situation (fieldnote, Oct. 18, 2022). The work on the data pipeline and particularly the curation process in 2021 was retrospectively described as very frustrating. Compared to 2021, when the process had been "very sloppy" (Emily, Oct. 18, 2022), Emily agreed that they were in an "okay spot." The vessel maps they now had were "much cleaner" (Oct. 18, 2022). The latest improvements included a new automated masking procedure and new features in the curation tool, such as the option to add vessels that DeepVess had missed.[28]

28 Furthermore, they even included a full re-implementation of the tool from scratch. The initial tool had been built as an image viewer with many features not required for the curation process, creating a researcher experience that was not smooth (fieldnote, Oct. 25, 2022). The new curation

However, when I asked James how they experienced the data pipeline now, he said that it was generally the "same old, trying to curate data" (fieldnote, Nov. 16, 2022). While all laboratory members seemed to agree that the changes to the data pipeline, including the curation tool, were improvements, the overall problem of efficiently and effectively preparing data for Stall Catchers was not yet considered solved. The pipeline was now producing data of sufficient quality, but curating was now (again) taking up too much of the researchers' time (fieldnote, Oct. 18, 2022). Indeed, some researchers still found the whole process related to Stall Catchers too time-consuming and saw no advantage over manually annotating the raw research images. The PIs had to invest a lot of effort in convincing researchers to "use" Stall Catchers (fieldnote, Oct. 25, 2022) and send data to the platform instead of manually annotating it themselves. This effort included improving the data curation tool and trying to standardize curation practices across the laboratory, which could vary widely between different researchers, especially in terms of the time it took. The new goal was to consistently reduce curation time to 15–20 minutes (Sean, fieldnote Nov. 4, 2022), since each research project would require several hundred stacks. They, for example, had to stop drawing lines again to speed up curation because it was too time-consuming. Drawing lines, however, was something that researchers described as very helpful and had greatly improved their curation experience. James, who had previously been dissatisfied with not being able to draw lines, described that, with this new feature, he got "the sense that I actually do it accurately" (fieldnote, Nov. 16, 2022). Now, with the need to speed up the curation process and, thus, skip drawing lines, he faced the risk of losing that sense of accuracy.

The shift in focus from achieving the required data quality through curation to minimizing the time required for curation was also accompanied, or even driven, by a new understanding of the ultimate goal of the processing beyond preparing data for Stall Catchers: While the researchers' initial goal was to create a "holistic perfect vessel network," they now focused on further improving certain segments within specific image slices that DeepVess had actually correctly identified (fieldnote, Nov. 16, 2022). This was related to the new insight that they only needed a certain total number of vessels for their research rather than correctly identifying every vessel in each individual image stack. Therefore, one of the previous first steps in the data pipeline, masking out dura, was changed completely and moved to the curation step after all the automated processing. Researchers would no longer manually mask out dura before sending the data through DeepVess. Instead, entire image slices were excluded from the final editing step, reducing the size of the final image stack and, thereby, the amount of data sent to Stall Catchers by more than half (fieldnotes Nov. 4 and 16, 2022). While the top was excluded to reduce visible dura, the image slices at the bottom were excluded because image quality tended to deteriorate with depth due to the weakened signal, sometimes causing DeepVess to incorrectly identify vessel segments where there were none. Sean described this new practice as "cutting aggressively." "[T]he more you cut, the less work you have, the faster you can go" (fieldnote, Nov. 16, 2022). "Cutting aggressively" was a relief to the researchers.

tool was cleaner and seemed to be working better, though it still took too much time to curate an image stack (fieldnote, Oct. 25, 2022).

James, for example, explained that he no longer felt like he was doing "pointless work" but could spend his time on more "productive" work (fieldnote, Nov. 16, 2022).

Speeding up the curation step and improving the researchers' experience with the tool was not the laboratory's only focus regarding the Stall Catchers data pipeline in fall 2022. Laboratory members were also dealing with a new problem that had only recently been identified. As researchers and Nishimura explained, they had not known until recently that they "hadn't closed the loop" (fieldnote, Oct. 21, 2022) of Stall Catchers. In one of their last internal meetings about Stall Catchers, they had realized that only a few researchers had been pulling data from the platform. Many researchers at the laboratory had been "dumping data" (fieldnote, Oct. 21, 2022) into Stall Catchers but had not studied the results coming back from the CS game. Therefore, these researchers did not really know what the Stall Catchers results looked like and did not include them in their research. A new goal was, thus, to close this loop by focusing the infrastructuring work on the phase after Stall Catchers participants had analyzed the data.[29] Yet another software tool was introduced to close this loop, describing another step in the continuous infrastructuring practices related to Stall Catchers and its researcher–technology relations.

Closing the Loop

Once a dataset was fully analyzed in Stall Catchers, a Human Computation Institute member notified researchers that they could download the results from the researcher interface[30] on the Stall Catchers platform. The results in the downloaded CSV files were structured with one vessel per line and columns for different information. While the file included more than ten columns, Sean focused on only three columns, one of which was the link to the corresponding video to validate the results. Even though this had not been intended in the design of Stall Catchers (as I discuss in more detail in Chapter 7), researchers went through the list of annotated vessels to review the generated annotations and to manually validate them by analyzing the video. The vessels, i.e., the lines of the file, were ordered according to the confidence level of the crowd answers for a vessel being stalled. High confidence reflected the agreement between different Stall Catchers participants that a vessel was stalled. Sean explained that he usually created an extra column for his answer and then reviewed each stall annotation with a confidence level between 1.0 and 0.5 (fieldnote, Nov. 4, 2022). The challenge of "closing the loop" was not only to get researchers to look at the data results and learn to trust them (see Chapter 7) but also to make this step more efficient and convenient for researchers.

The problem with this review approach was that the researchers were looking at the same data as the participants, i.e., the short videos of blood vessels that had been created from the original 3D imaging data through the multiple steps of data transformation described above. To answer their research questions, however, the researchers had to trans-

29 Closing the loop, PI Nishimura suggested, would also help build the researchers' trust in the Stall Catchers project as they could then engage with the results (fieldnote Oct. 22, 2022; see Chapter 7).

30 In addition to the user game interface, there is an additional interface for researchers to access their own projects and related datasets.

late the videos back or compare them to the original imaging data (and, ultimately, to the individual mice). This task was very cumbersome, as researcher Jada explained:

> I check the video and then I go back to the stack, which is a little bit difficult […] [T]hen I have a quick look whether I identify those [stalls] as well in the original video. […] [I]t's just […] for my sanity; I just would like to see these in the full stack. […] I'm kind of extremely used to it and see it in the context and there are also a lot of regions, edges, and stuff where stalls occur where it gets difficult for the program to really see those. (Oct. 27, 2021)

While the abstractions and transformations of the data were necessary both to count the vessels in a stack and to allow Stall Catchers participants to analyze research data, they also made it more laborious for the researchers to continue working with the results.

The laboratory also implemented yet another tool for the manual post-processing of the data returned from Stall Catchers to facilitate this process and further improve the researchers' experience with the project. This tool enabled researchers to open an unprocessed image stack and select the vessel they wanted to look at from the list of all vessels (fieldnote, Oct. 25, 2022). The tool then drew an outline of the requested vessel, similar to what the Stall Catchers participants had seen but on top of the original raw imaging data. This way, instead of watching the Stall Catchers videos and then trying to find the vessel in the original image stack manually, they could directly examine the vessel and its environment. Bringing the latter into view was another objective of the new tool, since the data sent to Stall Catchers only included one channel and, thus, not all the information, such as other cell types, was visualized in other channels. It was now crucial for the researchers to understand not only *if* a vessel is stalled but also *why* it was stalled.[31] Therefore, it was important to know what is around a vessel and to learn about the behavior of these other cells in relation to stalled or flowing vessels. The new tool allowed the researchers to see all channels.

At the time of my field research in 2022, this tool had just been implemented and was not yet being used by researchers (fieldnotes, Oct. 25 and Nov. 4, 2022). This new researcher–technology configuration carried the hope of closing the Stall Catchers loop and facilitating their everyday research practices, thus, solving some of the remaining problems with the project. However, how this new tool will shape and impact the data pipeline and practices surrounding Stall Catchers remains open at the time of writing.

What seemed to be clear, though, was that closing the loop would not be the end of the infrastructuring related to the Stall Catchers project. Improving DeepVess, for instance, had already been identified as a potential next undertaking, as the model was seen to require retraining or perhaps even replacement. The ML model was considered to be "lagging behind the rest of" (Anna, Oct. 14, 2021) the pipeline. Researchers Sean and Leander expressed their hope that "it would be great if we did not have to curate" (Sean, fieldnote, Nov. 4, 2022) any longer and if, instead, "the AI" were to be improved. "[I]f we can get it to the point where we don't have [to make] all of these little connections … that'd be really nice, so I don't know. The more I think that we can jump onto [improving the]

31 Charles, April 2023, private correspondence.

DeepVess part as opposed to adding on tons of post-processing" (Leander, Aug. 16, 2021). A new data pipeline focus with DeepVess had already been identified for future work.

The example of closing the loop shows how various intraverting researcher–technology relations are entangled in the data pipeline and how new potentials and tasks emerge within the same relations and enable new configurations in other interwoven ones. Closing the loop only emerged as a problem once data preparation had been improved to a certain point, and once that was solved, DeepVess could become the next target. Researcher–DeepVess relations will intravert yet again with the work on the ML model.

Reconfiguring Data, the Pipeline, and Researcher–Technology Relations

While the previous sections described in detail the implementation and steps of the Stall Catchers infrastructure on the laboratory's side, in this section, I summarize and analyze them with a focus on intraversions. I concentrate on two main points: first, how the human–technology relations keep changing with each reorientation and modification, focusing on the development of the infrastructure. This includes how the idea of the data pipeline as a means shifted to the idea of it as an investment alongside and because of the intraverting human–technology relations, what this tells us about the imagination of infrastructures, and how this impacts the researchers' working practices. Second, I show how the data pipeline and the work on it can be understood as part of an HC system itself.

The detailed descriptions of the improvements, modifications, and related reorientations revealed the complex intraverting sociotechnical interplay between humans and technology in which automated steps were interwoven with manual human tasks and vice versa. Agency and tasks within the relations were not fixed but shifted and redistributed along with these developments. For example, they shifted from the ML model originally developed to support and accelerate the researchers' work to the need for researchers to support DeepVess and other automated algorithmic steps by cleaning up their results. Similarly, the subgoals or tasks associated with infrastructuring kept changing, such as, as described above, the initial subgoal of creating a perfect vessel network, which was changed to a required number of correctly mapped vessels. DeepVess and the researcher–tool relations initially complemented each other toward the same goal of identifying the complete vessel network. In contrast, later, researcher–DeepVess–tool relations focused only on those parts of the DeepVess output considered "good enough" to achieve better efficiency and throughput, i.e., freeing up time and resources for the researchers. Retraining DeepVess to achieve better automated performance would be another step in this direction, which will go hand in hand with reconfigurations of researcher–DeepVess relations. Along with the development of the Stall Catchers pipeline, similar to what Mackenzie observed with the example of machine learners, both human and nonhuman actors were assigned new positions. "These positions are sometimes hierarchical and sometimes dispersed," Mackenzie writes (2017, 186), but the subject position in this hybrid relation is always mobile (2017, 186). Much like the introduction of new tools in the example of naval navigation described by Hutchins (1995a), the introduction of the ML model created new tasks for the researchers, such as data curation and image masking. While pushing the data from one automated step to the next was considered "pretty easy" (Isabel, Oct. 14, 2021), and the researchers' role

here was simply to guide the data through the pipeline, taking it from the computer and handing it back (Isabel, Oct. 14, 2021), new tasks, such as curation, now became a bottleneck in the data pipeline. Instead of relying on the ML tool to take over the researchers' task, researchers had to focus on assisting the tool and evaluating its performance.

It should have become clear that the data pipeline, which was originally conceived as a means to enable data analysis via Stall Catchers and to speed up Alzheimer's disease research at the laboratory, itself became a central focus of work at the laboratory, which was not completed even five or six years after the project was introduced. In fact, during the time of my field research, researchers were particularly focused on working on the data pipeline itself. This was not always perceived as satisfactory by laboratory members, as for some of them, their actual research goals fell out of sight while working on the data preprocessing. There are many translation steps from raw pixels depicting vessels in the mouse brain, to knowing the total number of stalled vessels in the dataset, to deriving insight into the effects of reduced blood flow in Alzheimer's disease in mice. In contrast to working on the data pipeline, when researchers were still manually annotating stalls—the tedious activity that the CS project was designed to replace—they, at least, had a stronger sense of directly working toward their research goal; the stalls were the center of interest, and though tedious, their work was directly concerned with analyzing them. This sense of working toward their research goal was lost when they were working *on* the infrastructure rather than *using* it; this work was effectively one level of abstraction removed from their actual research. Nevertheless, the focus on the infrastructure was also seen as a future investment that would eventually facilitate the research once the data pipeline was running smoothly. In 2021, James explained: "I guess it's even like [a] long-term investment in the future […] if this will be faster than manually, […] but right now it's hard cause it's a lot" (fieldnote, Aug. 16, 2021).

However, during the time of my research, even as the data pipeline, data quality, and the researchers' experience improved gradually, the idea of the data pipeline as a means that could slowly disappear into the background turned into an understanding that the pipeline was an "ongoing" project that would never be finished. With the solution of one problem and a better understanding of the current processes, new problems arose that had to be tackled, such as the loop that had to be closed or the curation of data with the introduction of the ML model DeepVess. By fall 2022, the laboratory was closer to its goal of obtaining correct results, i.e., not only correctly classified vessels from Stall Catchers but also the correct number of vessels in an image stack. However, it seemed that the closer they got and the more they worked on the infrastructures and improving the processes, the more they identified gaps and problems they had not thought of before (fieldnote, Nov. 4, 2022). "*[N]ew instrumentation gives new perceptions*," argues Don Ihde (1990, 56, emphasis i.o.). Viewed through the lens of intraversions, each improvement and change in the pipeline opened up a space for new scaling, revealing questions or problems that had previously been out of sight. In fact, infrastructuring, in the example of the data pipeline for Stall Catchers, involved several intraversions in the research–technology relations, most notably the introduction of an ML model to do the segmentation necessitating manual preprocessing steps, which were then refactored to semiautomated postprocessing steps, again creating the need for manual curation.

The observation of the incomplete pipeline is not to say that the efforts to improve the infrastructure did not make a difference but to emphasize that infrastructure is unruly (Wynne 1988). One laboratory member explained that the processes related to the Stall Catchers project would probably never be "standardized" compared to other established collaborative processes because of the "variability within the data:"

> [Y]ou have to spend some time, and you also have to make sure for your science that it's […] sort of correct because it is a little bit of a black box. But I don't think it's negative. We also have many other black boxes. If I send something out to some facility, they are [sending] me some excel sheet back with data that I also don't have the real insight to, but I think it's more standardized and this is just not as standardized and probably will never be. It's just because of variability within the data. (Jada, Oct. 27, 2021)

In addition to the variability within the data mentioned by Jada, Charles explained that there will always be some "nonlinear pacing" (Oct. 14, 2021) in animal experiments, which cannot be predicted and prevents the process from being completely standardized.

The continuous changes and modifications to the pipeline's steps that went hand in hand with the shifting tasks researchers had to perform were difficult for new laboratory members who were still being introduced to Alzheimer's disease research with Stall Catchers. As Benjamin explained, for example, every time he tried to learn how to curate data, the task or program had changed or a new tool had been introduced that he had to get used to again, resulting in him not being eager to try it and preferring to analyze data manually instead (fieldnote, Oct. 26, 2022).

Even as the researchers' understanding of the pipeline changed from a means that would be completed one day to an ongoing process, they were confident that Stall Catchers would eventually save researchers time (fieldnote, Nov. 4, 2022). At the end of my field research in November 2022, Sean stated they were getting closer to the point where Stall Catchers would make the analysis process easier than manually counting stalls (fieldnote, Nov. 4, 2022).

Focusing on the development of the pipeline further illustrates how digital infrastructures in the example studied, and particularly trained computational models, carry inertia. They often lag behind spontaneous changes in processes and practices, such as variations in experimental settings. While it is comparatively easy to exchange a microscope objective, adjust the laser trajectory, or alter the treatment of mice with a direct impact on research, it takes a long time to retrain an ML model to account for these changes or to develop a custom data annotation tool for researchers to use.[32] Not only does code need to be changed, but the adjustment of downstream systems and the lengthy evaluation and feedback cycles involved further increase the effort for such changes. This is

32 This is not necessarily generalizable to all ML models but specific to the situation described. Some ML models, for example, are designed to continuously get retrained and updated. However, this depends on having the necessary meta-infrastructure in place, i.e., the tools, processes, and systems required to support and maintain ML models over time, as well as enough personnel resources to support ongoing model development.

not always feasible within the constraints under which researchers work, and as the series of pipeline modifications at the laboratory shows, instead of tackling such problems at their deepest level, other workarounds are often introduced. The decision to improve the output of DeepVess via complex post-processing instead of improving the model itself, i.e., retraining it, is an example of such an improvised, temporary solution, which is also related to the fact that the laboratory lacked the necessary ML expertise after the researcher who had originally implemented the model had left. Similarly, earlier on in the pipeline, manual masking of the dura and other noisy remnants of the imaging process could also be seen as such a workaround.

The data pipeline, understood initially as a means to enable outsourcing the image analysis to the Stall Catchers platform, had become a major focus and investment. This discrepancy in the meaning of the pipeline led to tensions in the laboratory, since, even in 2022, it was still considered a "side thing" (fieldnote, Oct. 26, 2022) by some researchers rather than a core part of their Alzheimer's disease research process. Curation was often practiced in the late evenings when researchers had time to do it (fieldnote, Oct. 26, 2022). In addition to this attitude, the fact that the curation program did not work well on all laptops due to the large amount of computing power it consumed played an important role here. Benjamin explained that he preferred to curate data every now and then while in the laboratory waiting for an experiment to be completed (fieldnote, Oct. 26, 2022). But since the program was too slow on his computer, this was not an option. Sean, whose laptop managed to run the program as long as he pulled the data from a solid-state drive, tried to find time to curate data whenever possible (fieldnote, Nov. 4, 2022). Data curation was something researchers typically incorporated into their daily working practices with lower priority, organizing it around their other tasks. Changing this understanding of data curation, therefore, was one of the main aims of the laboratory's PIs during the time of my research. Stall Catchers were to become a central part of the research and not just a side thing, explained laboratory member Isabel, who tried to get other researchers to "use it" (fieldnote, Oct. 25, 2022). According to Isabel, what was needed was a "paradigm shift" in their thinking (fieldnote, Oct. 25, 2022).

Looking at Stall Catchers, including the data pipeline as a whole, the pipeline can be understood as a part of a "higher-level" HC system, even though it was not designed and considered as such by the Human Computation Institute and laboratory. When I suggested this idea of a "higher-level" HC system and the pipeline as a part of it, Michelucci explained that he had not done so because the human–computer and human–AI handshakes in the pipeline are not automated but manual, meaning that the individual steps are not fully streamlined (fieldnote Nov. 4, 2022). To move from one step in the pipeline to the next, humans must actively intervene, for example, by moving the data from one folder to another or by starting a successive program. However, I here aim to show how this is, nonetheless, an insightful perspective for understanding how different human–technology relations intravert in their entanglements and how the overall system evolves.

Viewing the laboratory as one actor in this higher-level system and Stall Catchers as another, the laboratory's role shifted from fully (but slowly) performing all of the analysis to delegating a key part of the analysis to Stall Catchers. The laboratory's own focus shifted to providing better data outputs for it to use, while introducing a new manual

validation step to reincorporate Stall Catchers' data. At the end of my field research, this system still faced considerable friction and the efficiency gains on the side of the laboratory were not yet as substantial as researchers had hoped. However, it was already on a path of continuous change and intraversion that both actors expected would eventually lead to significantly outperforming the system's initial configuration. The intraversions occurring at the scale of this overarching system are slower than those occurring within either system viewed in isolation. Given this and the project's short existence, it is still too early to give a full account of how human–technology relations and the actors' roles within this system are intraverting over time. However, during my field research, I could already observe some tendencies and imaginations of potential future modifications in these relations, tasks, and responsibilities. These included ideas for changes to the task on the Stall Catchers platform itself (fieldnote Oct. 26, 2022) as well as involving Stall Catchers participants in earlier steps of Alzheimer's disease research conducted at the laboratory and improving the ML model with input from the game platform. Even though neither the laboratory nor the Human Computation Institute seemed to actively consider or describe this higher-level HC system as such—only Stall Catchers itself was usually described as an HC system[33]—there was still an awareness among researchers that Stall Catchers' capabilities and infrastructure could be further integrated into their work.

The researchers had discussed the idea of using the Stall Catchers participants' identification of poor-quality videos by "flagging" them as input for improving DeepVess based on an HC approach (fieldnote, Aug. 10, 2021). Another idea that was discussed during my field research with the common goals of both reducing the laboratory's own workload and involving Stall Catchers participants in more steps of Alzheimer's disease research was to invite some of the more experienced Stall Catchers participants to take on some more complex tasks (fieldnote, Oct. 21, 2022), as data curation turned out to be more time-consuming and laborious for researchers than anticipated. The idea was also related to the fact that more and more participants were analyzing data on the Stall Catchers platform, increasing the data throughput to such an extent that the researchers in the laboratory sometimes could not keep up with generating and preparing new data for Stall Catchers. Improvements in the data pipeline even further reinforced this, as better data quality reduced the number of segments sent to Stall Catchers. However, this meant fewer videos for participants to analyze, which could exacerbate the problem of lack of data (fieldnote, Aug. 10, 2021). Participants asking for more data was a new dynamic for the researchers. Initially, the laboratory had a huge backlog of data to send to Stall Catchers and researchers had to wait for participants to analyze it. One concrete idea to address this asymmetry was to allow participants to work closer to the raw data in the curation steps (fieldnote, Oct. 21, 2022). According to researcher Sora, this could benefit both biomedical researchers and Stall Catchers participants, as it would contribute to the research process and potentially be more interesting for participants (fieldnote, Nov. 1, 2022). In some ways, this would not only lead to new transformations of the infrastructure and related human–technology relations but also change the relationship between researchers and participants, since the latter would

33 This is related to the human-in-the-loop understanding behind HC that I discussed in Chapter 4.

then be included in the scientific process of vessel mapping and cleaning (fieldnote, Nov. 1, 2022). When asked how this would be different from the current involvement of Stall Catchers participants, a laboratory member explained that participants would then be introduced to and confronted with the variability and uncertainty of science, which were "pretty much removed" in Stall Catchers (fieldnote, Oct. 21, 2022). If participants were included in data curation, they would have to make their own judgments, since not everything in science can be decided with 100 percent certainty (fieldnote, Oct. 21, 2022) and, hence, would have to be trained as researchers first. This illustrates the boundary work biomedical engineers do to distinguish between their research and the analysis step outsourced to CS participants. Although this idea had not been realized at the time of my research and was also controversially discussed in the laboratory, it points to possible future intraversions within the human–human and human–technology relations in Stall Catchers.

Looking back at the first five years of the Stall Catchers project and the development of the interplay between human participants and computer algorithms, the idea of introducing new types of tasks to be solved via participant–technology/computational tools relations was also considered a very complex and long process: "I've learned […] that it's difficult to make a single task pretty consistent. So, the challenges of making multiple tasks would be a whole other thing. Well, I'm still hopeful," said Nishimura (Dec. 7, 2021).

The question of whether and how this intraversion will unfold remains open at this time. However, two key points I would like to make are already evident in the stages the system has gone through so far. On the one hand, HC-based CS systems are imagined differently by their actors, with their own specific aims and needs in mind. In this, the overarching structure of the system may not always appear to or be seen in the same way by each actor. On the other hand, such systems are never understood as complete or closed systems. This is not only because, like many systems, they exhibit problems in their development and maintenance. Rather, they are specifically thought of as being continuously open to change in the hope of surpassing their current function, which manifests itself in intraversions of human–technology relations in the HC-based CS systems. As Schaffer put it: "[I]t's not *done* […]. I mean, […] there'll be a new wrinkle in the data. There'll be some new problem that comes up because we're not going to keep doing just this exact same thing over and over" (Dec. 7, 2021). This continuous evolution of HC systems is particularly apparent in the analysis of the dynamics of the ARTigo example.

Moving Forward in Concert

Shifting our focus from the microanalytical perspective on HC systems in their everyday situatedness to the broader evolution of HC-based assemblages over extended periods (i.e., years), a pattern emerges. This pattern describes the dynamic and nontrivial interactions between human–technology relations within HC systems and advances in AI computational capabilities. In the following exploratory remarks, using ARTigo as an example, I show how these also undergo a continuous transformation pattern, evolving in parallel with the intraverting human–technology relations. They influence and create each other in concert as they move forward.

The main reason for my focus on ARTigo at this point is that the development of ARTigo's life cycle presents a revealing example of the changing relations between HC and AI research despite its less extensive treatment in previous chapters. When I started my ethnographic fieldwork in the last weeks of 2019, ARTigo had been an active project for over ten years. The project idea emerged in 2007 from computer scientist Bry, who was inspired by Von Ahn's ESP game (Von Ahn 2005),[34] which "was the first system where work activity was seamlessly integrated with gameplay to solve Artificial Intelligence problems" (Von Ahn 2005, 70; see also Law 2011; Lazar, Feng, and Hochheiser 2017). In the two-player ESP game, players (without seeing each other or knowing the other player) had to describe images with words but only gained points if their words or tags matched those of the other player. Bry brought this idea into the area of art history. Computer scientist and ARTigo team member Ben described in our conversation in early 2020 that "[f]or art history, […] the idea of doing it this way and making it this big and also combining it with the several games, that is to say, making it a platform, was a very good idea. No one had done that before" (Feb. 24, 2020). In an article by digital humanist and data scientist Stefanie Schneider and art historian Hubertus Kohle, who are both part of the ARTigo project, the authors describe ARTigo and its twofold goal. They characterize it as "an internet platform in which digital reproductions of artworks are presented to an audience with unknown qualifications, who then annotate these artworks in a playful and competitive way" (Schneider and Kohle 2017, 82). Furthermore, they define it as "a semantic search engine which can master large image sets based on these crowdsourced annotations (*tags*) without having to rely on the expensive *manpower* of specialists—or even on artificial intelligence from the field of computer vision" (Schneider and Kohle 2017, 82, emphasis i.o.). There was a need for such a semantic search engine in the field of art, since several million digital reproductions of works of art existed in electronic repositories, but there was no way to retrieve them based on specific criteria (Schefels, n.d.). Computational search, at that time, was "still very limited" (Schneider and Kohle 2017, 82).

In addition to these official goals, for Ben, ARTigo was also an opportunity to "bring art history […] into the computer age and to prove with ARTigo that computer science can also do research in art history, that is, with data science, data analysis" (Feb. 24, 2020). While participants contributed text-based tags, an unsupervised ML method[35] processed these contributions to build and constantly improve the semantic search engine (Bogner et al. 2017, 53). The advantage of a semantic search engine is that it can be used to "search for [art]works whose identity cannot be determined by identifying the author and title, which are available as metadata in traditional image archives" (Schneider and Kohle 2017, 82).

While the ARTigo project was funded by the German Research Foundation from 2010 until 2013, the team continued to work on it and maintain it beyond the end of the fund-

34 The game is named after "extrasensory perception" (ESP), which is also known as the "sixth sense." Even though Von Ahn does not explicitly state so in his dissertation, it becomes clear from the context to what he is referring.

35 More precisely, the team developed a "Higher-Order Latent Semantic Analysis" (Wieser et al. 2013; Wieser 2014).

ing period. Up to 2016, over nine million tags had been gathered from "on average 150 persons a day playing on ARTigo" (Bry and Schefels 2016, 5), and by 2023, Schneider and colleagues report on 10,679,711 annotations (Schneider, Kristen, and Vollmer 2023, 4) by several tens of thousands of participants (Kohle 2018).

The ARTigo platform included several games to collect different tags. As Ben explained:

> There are different games for different tag groups. So, the ARTigo game, the classic one, that's very general, it collects tags very broadly, while the other games, they keep refining that. They then build on the dataset and then also again the other way around, other [games] build on that dataset then again and this way, it keeps refining. And this way, you then get a better description for the images. (Feb. 24, 2020)

The different tags and their corresponding games can generally be divided into "simple descriptions" collected by the ARTigo game, which resembles the ESP Game (Bogner et al. 2017, 53; screenshots of the initial ARTigo UI can be found at Citizen Science Games 2019), and "more specific descriptions [that] are collected by 'diversification games' using simple descriptions collected by description games" (Bogner et al. 2017, 53), such as the ARTigo game.

Finally, "integration games" collected "annotation clusters" based on tags collected by all different games (Bogner et al. 2017, 53). Together, these different games and tag forms or annotations—all of which were text-based—allowed the "ARTigo gaming ecosystem as a whole [to] perform[] better than each of its games alone" (Bogner et al. 2017, 56). ARTigo, like Foldit, is explicitly built on the idea that nonprofessionally trained participants will provide different tags than professional art historians (Kohle, Dec. 4, 2019). The value of annotations by nonprofessionally trained people lies in the fact that while individual descriptions may refer to objects or colors in images of artworks, when combined, "the tags demonstrate a wisdom not present in any single word, but only in the collection" (Kohle 2016, 3). By saving only those tags contributed by at least two participants, ARTigo aims to exclude "deliberately false input" (Kohle 2016, 2).

In late 2019 and early 2020, I interviewed art history researchers and computer scientists involved in the project to learn about their experiences with the project and how they would describe ARTigo's journey so far. Team member Finn explained:

> Overall, you have to see that it is somewhat of a flagship project for all citizen science projects. It was one of the earliest projects to experiment a bit. And also one of the projects that, I think, had the most far-reaching consequences because art history […] has no datasets with actual annotations available. Currently, obviously, it has decreased a little bit. Because I also believe that this system of actually assigning text annotations for images, that is no longer popular. (Dec. 16, 2019)

The team universally agreed that ARTigo was very successful as an early HC-based CS game, especially since it was launched before ML, computer vision, and deep neural networks became "such a cool research topic again" (Emilia, Nov. 8, 2019). The technologies

that exist today to automate tagging the semantic content of artworks were not available at the time, computer scientist Emilia explained in our conversation (Nov. 8, 2019).

By 2019, ARTigo seemed to have passed its peak. Due to a lack of funding, it was difficult to sustain such a large project. While various research studies (including bachelor's and master's theses)[36] had been conducted on and around ARTigo during its peak period, Emilia explained,

> not much is happening there at the moment. […] It's just kind of a bit of an old project at the moment that kind of just keeps going […]. But currently, in terms of the underlying technology, it is just […] more or less end of lifetime. So, it also constantly crashes. So, if you try to access the ARTigo page, then it can be more often that 50[3][37] or so, so that the server is not working because somehow the system is not working. And I think the […] daily business now is basically maintenance and restarting the server when it crashes again. (Nov. 8, 2019)

Thus, at the end of 2019 and the beginning of 2020, the team's main tasks were keeping ARTigo alive and maintaining the platform by occasionally restarting the server and answering press inquiries.[38] Finn sighed and explained that it "happens relatively often that I have to restart [the server] three times a day" (Dec. 16, 2019). At the same time, the platform was somewhat outdated.[39] Bry argued that "[i]f you want a platform to continue to remain popular, you need to work on the platform constantly. A game platform cannot stay the way it was for ten years. And that's hard work" (Feb. 3, 2020). For ARTigo to remain popular, Kohle agreed, it would have to be reimplemented from scratch with different games and less text-based approaches (Dec. 4, 2019). Finn suggested that participants today prefer to drag or click on things rather than type (Dec. 16, 2019). Such a reimplementation could also, according to Kohle,

> address a weakness of the application related to its logic. In particular, in cases where we also want to use the annotations to train computational neural networks, such that the computers will eventually be able to recognize the objects depicted in the works themselves, the information is too imprecise: If an image is tagged with the term "dog," then a dog appears somewhere in the image, but it remains unclear which part exactly contains the dog, and can only be deduced by human intelligence. Another version of the game could be to assign terms to certain areas of the image, for example, to drag the term "dog" with the mouse to the image representing the dog. (2018, 2–3)

---

36 E.g., Schemainda (2014); Taenzel (2017); Greth (2019).

37 Emilia mentioned the HTTP response status code 505, which refers to "version not supported." It is, thus, likely that she was referring to the response status code 503, which refers to "service unavailable."

38 Due to ARTigo's collaboration with museums, in which some version of ARTigo was featured in exhibitions, for example, the project still received public interest and media coverage from time to time (Finn, Dec. 16, 2019).

39 The sustainability problem of such CS projects and platforms was not only raised by ARTigo team members but was also a primary motivation for Michelucci to build the Civium ecosystem (see Chapter 4).

While researchers from the ARTigo team had already worked with clustering algorithms in 2017 to divide artworks into groups based on the crowd's annotations (Schneider and Kohle 2017), the resulting clusters could later be used to train CNNs. These could eventually allow the automated classification of images of artworks without prior manual annotation (Schneider and Kohle 2017, 88). However, I would like to remain a bit longer with the state of ARTigo at the beginning of my field research.[40] At that time, compared to other software and game interfaces or UIs in general, ARTigo no longer met the users' expectations (Ben, Feb. 24, 2020).[41]

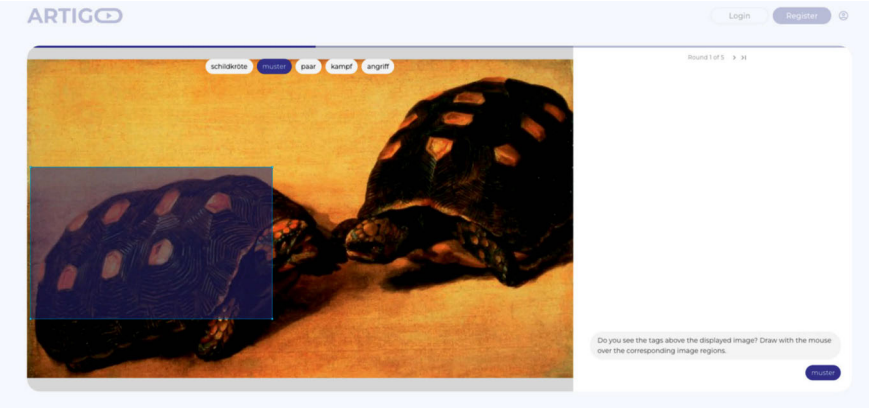
From a user perspective, not much seemed to be happening with the ARTigo platform for the next year and a half during my main field research period (2020–2021). Yet, I knew from my interviews with ARTigo team members that they were working with computer science students (Greth 2019) to reimplement the entire platform. I frequently checked the website to see if the project was still running or if a new platform had been launched. At some point in 2021, the platform was only accessible from the Munich Scientific Network provided by the data and supercomputing center Leibniz Supercomputing Centre connecting academic institutions in Munich, including the LMU and the Technical University Munich.

To my surprise, when I checked the status of ARTigo in November 2022 (as I continued to regularly do throughout my research), an entirely new iteration of the platform appeared (*cf.* Schneider, Kristen, and Vollmer 2023). Instead of the eight mini-games available on the previous platform, players could now choose between two game modes on the new platform, which was released in November 2022. The first resembled the old games in that it asked participants to annotate images of artworks in a text-based manner. However, in addition to asking for descriptive tags, the game also asked participants to explain, "What feelings does it [the digital reproduction of the artwork] trigger in you?" (Ludwig-Maximilians-Universität n.d.a). The second game mode invited participants to annotate image regions based on the word tags provided. The latter presented participants with a completely new task. Instead of creating text-based tags, they now had to draw outlines around the object region(s) corresponing to one of the provided tags (Figure 10).

40 Despite the fact that the ARTigo platform itself was not much used at the end of 2019, annotations created on the platform were still used in other projects. For example, they were included as training data in the new research project "iART" funded by the German Research Foundation (Kohle, Dec. 4, 2019). The project, funded from 2018 to 2023, was a collaboration between the Chair of Medieval and Modern Art History at LMU Munich, the Visual Analytics Research Group at the Leibniz Information Centre For Science and Technology University Library, and the research group Intelligent Systems and Machine Learning at the Heinz Nixdorf Institute, Paderborn University, and included two researchers of the ARTigo project, Hubertus Kohle and Stefanie Schneider (Deutsche Forschungsgemeinschaft n.d.).

41 ARTigo participant Helena, who had responded to my public call for participation on ARTigo's Twitter account, expressed her wish for a more "modern" website (Jan. 27, 2020).

Figure 10: ARTigo Play mode "annotate image regions based on tags"



Source: Screenshot taken by LHV on Feb. 27, 2023 (https://www.artigo.org/de/game)

This development of the ARTigo platform can be compared to the evolution of CAPTCHA described in Chapter 4, which brings us back to the very first HC. As the algorithms for optical character recognition of written text improved, the task was changed to identify objects in images, such as traffic lights or taxis, to improve the AI algorithms for image recognition. The introduction of this new game mode in ARTigo provides an interesting example for my analysis for at least three reasons. First, it corresponded to the ideas presented by team members in our conversations in late 2019 and early 2020 described above. According to these, participants today were less interested in text-based tasks and more interested in tangible modes of interaction. In the mobile version, participants could circle the areas with their fingers. This new form of engagement was understood to be more entertaining in times when smartphones and touch screens had become standard for many people. As a GWAP, ARTigo depends on volunteer engagement and, therefore, needs to be (and remain) entertaining.[42]

Second, new modalities of information beyond test-based tags are collected with the new game mode. This new information then allows researchers to improve computational analysis tools and AI models and create new models, such as a CNN, which was already described as a future idea in the quote from Kohle (2018) above. In this way, it allowed AI research to advance. When I contacted the ARTigo team in March 2023 to learn about the goals behind introducing this new game mode, they confirmed this observation with their email reply that the new game mode served the purpose of "increasing precision. So far, [the label] dog could be used during annotation, but it was unclear which area of the image was being referred to. However, this would be beneficial to teach the computer to recognize a dog independently."[43]

42 Additionally, the new ARTigo platform included an input frame that was inspired by messaging apps, in which a chat, or chatbot, was simulated to create something that resembled a dialogue between the chatbot and participant (Schneider, Kristen, and Vollmer 2023, 3).

43 E-mail exchange between LHV and the ARTigo team from March 26, 2023.

Finally, with the advancements in computer vision research in recent years, the state of the art in AI research in 2020, shortly before the launch of the new ARTigo platform, could not be compared to 2007, when the idea of ARTigo was conceived. ARTigo, according to PI Kohle, is based on a normative understanding similar to the Human Computation Institute's Hippocratic oath: "First use no humans" (Michelucci and Egle [Seplute] 2020) (see Chapter 4). In an interview with the science blog *Bürger Künste Wissenschaft*, PI Kohle argued that "[w]hat is important is that people perform real tasks and do not do things that a computer could do, just so they are involved somehow" (Kohle 2019). Following this understanding, it could be argued that the new games were necessary to ensure that participants continued to perform tasks that computational systems could not, in fact, solve. Even if this was not the primary motivation for the new games, this new development could be understood as yet another iteration in a series of intraversions. Instead of asking participants for descriptions of the images, the software now presents them with information previously entered by humans to confirm the tag and specify the image region(s) corresponding to it.

ARTigo's early games built upon each other, forming an "ecosystem" in which participants filled in where the capabilities of the computational parts of the system fell short. Subsequently, two things happened: significant amounts of useful data were collected and AI research, particularly in computer vision, made significant advances that likely made it possible to perform at least some of the human tasks in ARTigo using computational methods. This combination of achieving the system's initial purpose (at least to a large extent) and the advances in the capabilities of computer vision models led to the emergence of a new iteration of the HC system by opening the space for the system to fulfill a new, elevated purpose. ARTigo's evolution here resembles that of the visual database project ImageNet described by Gray and Suri, who discuss its evolution as an illustrative example of the paradox of the last mile of automation: "Humans trained an AI; only to have the AI ultimately take over the task entirely. Researchers could then open up even harder problems. For example, after the ImageNet challenge finished, researchers turned their attention to finding *where* an object is in an image or video" (Gray and Suri 2019, 8, emphasis i.o.). Even though, in the case of ARTigo, AI has not yet fully taken over the annotation task, new, more advanced problems could be addressed. For this new iteration of ARTigo, the platform, interfaces, and modes of engagement also needed to be adapted and developed further to remain interesting, engaging, and even ethical.

Furthermore, it could be argued that ARTigo, as an HC-based CS, *had to* undergo this development in order to remain an HC system, i.e., a system that "harness[es] human intelligence to solve computational problems that are beyond the scope of existing Artificial Intelligence (AI) algorithms" (Law and Von Ahn 2011). In the case of ARTigo, this was a specific problem at the intersection of computer vision and art. Progress on this immediate problem, for example, due to improved computational capabilities through more powerful AI models, meant that computers could increasingly take over the tasks performed by humans in the HC system. These intraversions, as in the case of both Stall Catchers and Foldit, are often largely driven by internal developments and based on the data collected or generated by the system itself. By contrast, ARTigo's intraversion was not primarily based on data collected as part of the project. This difference between ARTigo and the other systems studied does not imply that these developments should not

be viewed as intraversions. Rather, intraversions in the HC systems' human–technology relations can be significantly influenced by both internal and external factors. HC systems often directly support or drive the AI advances that cause the intraversions, but this is not a necessary condition. This difference reinforces the notion that advances in computational capabilities do not typically lead to the human side of the system being considered redundant. Instead, as Michelucci explained, "humans can move on to the next task" (fieldnote, Sept. 30, 2021). The assemblages themselves typically do not cease to exist when the current purpose of the system is achieved but are repurposed toward a new goal, which only comes within reach due to the advances made previously. By adapting to new problems and needs, i.e., by intraversions of its human–technology relations, these assemblages themselves become something new (Deleuze and Guattari 2013, 7).

Contingent, Imagined, and Emergent Intraversions

In this chapter, I have analyzed different human–technology relations and how they intravert in the HC-based CS projects Foldit, Stall Catchers, and ARTigo in everyday life's instantaneity, over time, and on different levels.

The analysis shows that the relations are not fixed but open and dynamic and how the idea of "hybrid intelligence" and, thus, the target of HC, keeps moving. The target remains partly an empty shell which gets filled with the instantiations of continuously adjusted human–technology relations as such systems move forward. Role assignments here are never fixed but in flux not only in their everyday enactment but also by the design of the system itself. Built to be changed again, the nature of the tasks to be performed and the purpose of the system are constantly changing along with the intraverting relations. Given the continuous formation and alteration of human–AI or –technology relations in HC systems, it is not sufficient or possible to define once and for all the intended and realized relations between humans and AI. Instead, they must be traced and analyzed *along* and *with* their becoming. These observations of intraverting participant–technology relations not necessarily apply only to the examples studied but also to other HC-based CS systems, such as Galaxy Zoo,[44] in which participants first provide data for training ML models and then AI models, instead of replacing human participants, are introduced into

44 The online CS project Galaxy Zoo was launched in 2007 with the goal of inviting participants to classify galaxies from the Sloan Digital Sky Survey (Lintott 2019, 41) because "the human brain is much better at recognising patterns than a computer" (Galaxy Zoo, n.d.). The project has seen many changes and modifications since its start. Despite its huge success regarding participant engagement and its purpose of classifying galaxies, the project faced the problem that they had too much data to be analyzed by human participants. To solve this problem, researchers started combining the participants' "classifications with those of machines, inspired by the idea that the combination of both automatic and human classification may be more powerful than either alone" (Zooniverse, n.d.). Similar to the approach in Stall Catchers, they used participant annotations to create CNNs "that make probabilistic predictions of Galaxy Zoo classifications" (Walmsley et al. 2020, 1555). Building on the work that some among the numerous images of galaxies are more significant than others (Walmsley et al. 2020, 1555), the Galaxy Zoo team introduced the new "enhanced" project mode in which participants are presented with those images to classify that have been selected by the classifier. With the "active learning approach" of using the human clas-

the same projects. The concept of intraversions helps to analyze these shifts and oscillations of positions, responsibilities, and tasks. Drawing on the example of user–technology and researcher–technology relations, I demonstrated how the intraverting relations reconfigure the tasks, practices, and forms of engagement of different actors within the sociotechnical assemblages.

In the example of Foldit, algorithms whose actions could be observed by human participants were first implemented to try to solve the protein structure prediction problem automatically. Participants' requests and involvement led to Foldit, in which computational tools assisted humans in manually manipulating proteins, which then evolved into a platform in which participants again relied primarily on automated algorithms to manipulate the protein but with a newly gained level of involvement and control. These evolutions of user–technology relations set the stage for entirely new human–AI relations to evolve in the form of supporting the development of Alphafold and its subsequent integration into the platform to review and comment on the protein structures developed by human–computational tool relations.

In the example of Stall Catchers, an analysis problem that could not be solved with current AI technologies led to the development of an HC-based CS in which volunteer participants were invited to fill in and assist by analyzing short research videos under the supervision of algorithms. This then facilitated the development of ML models, which were subsequently integrated into Stall Catchers to assist in the analysis. Here, the AI bots based on the ML models became both coworkers and competitors to human participants and could serve as preprocessors for humans in the future, thereby, also influencing the human participants' practices from analyzing videos of varying difficulty to focusing on the hard ones. This, in turn, would also impact the human participants' experience of contributing to Stall Catchers. If ML models can one day completely take over the analysis task, the designers and researchers believe that human participants could then move on to other tasks that are not yet computationally solvable, such as curating data before it is analyzed by AI bots, which would again intravert the human–AI relation.

Similarly, the researcher–technology relations were continuously changing in the HC system formed by the researchers of the Schaffer–Nishimura Lab and the Stall Catchers platform. They intraverted from the manual labor of analyzing data on the part of the researchers, to the development of Stall Catchers, accompanied by the introduction of computational tools and AI models to reduce this workload in the laboratory and the introduction of new tasks to be performed by researchers, such as masking data, correcting DeepVess' errors, and successfully reincorporating Stall Catchers' output into their scientific work.

Finally, the analysis of the long-term evolution of ARTigo revealed the interactions between intraversions of human–technology relations in HC systems and advances in AI. Due to advances in computational and algorithmic capabilities, participants' tasks and ways of engaging with the platform, as well as the purpose of the games themselves changed over the years. While participants initially provided text-based annotations about the images' content, they were then asked by the software to provide information

sifications to feed into the model, the latter keeps refining and, the assumption is, less and less labeled data will be required (Walmsley et al. 2020, 1555).

about their feelings when viewing the artworks and to specify image regions for existing annotations through tangible modes of interaction. These reconfigurations were necessary to ensure participation and to keep the HC system at the edge of technological development.

As the examples illustrate, intraversions are processes that, even when stabilized for a certain period of time, eventually present openings for new tweaks and improvements; the circumstances in which they occur tend to actively invite, almost require, such change. However, they are not arbitrary but contingent, imagined, and emergent. They are *contingent* in that their becoming always depends on previous relations and given materialities. They are also *imagined* in that they can occur through external deliberate design modifications, such as decisions made by developers, and *emergent*, in that they depend as much on and evolve through the underlying human–technology relations, which create new possibilities through their dynamic and partly unstable nature. In the Foldit example, participants requested to be involved in more active ways than just observing the automated program in Rosetta@Home, leading to the creation of the Foldit game, in which participants became decision-makers regarding the next steps to fold proteins and the automated tools assisted them in their attempts. In this way, participants can also be understood as designers of the intraversions of the human–technology relations in which they are involved. In the example of Stall Catchers, participants expressed their perspective on the AI bots as both assistants and competitors and, thus, actively related and contributed to the experimental research on new hybrid human–AI combinations in HC-based CS projects.[45] Participants accommodated the new presence of AI bots and, thereby, reflected on how they wanted to interact with AI bots in Stall Catchers. Following Dorrestijn, they "perform[ed] a transformation to their hybrid self" (2012a, 117).

At the same time, intraversions are still influenced and evoked by coincidental or accidental events, material breakdowns, or different relations intervening with each other. Finally, intraversions also go hand in hand with resistance and divergent understandings of the meanings of infrastructure, roles, and overall aims of CS games. On a more abstract and subject-focused level, these practices of resistance or tactics employed by different actors within intraverting human–technology relations can be understood as "coping with [the] influences" (Dorrestijn 2017, 318) of technology. In this sense, new subjectivities emerge from these intraverting relations (Foucault 1983; 1988).

Together, the HC-based assemblages studied are, thus, shaped by different actors, their intentions, and entangled human–technology relations that do not always align frictionlessly. Alignment or reterritorialization processes play an important role in bringing together the different interests, needs, visions, and material possibilities of the actors involved. In the next chapter, I discuss the example of trust building as one such alignment process, which became necessary due to the destabilization of established trust-building practices by intraversions. I show how trust emerges and must be adapted

45 As an example, even though AI bots redeeming points was being discussed as a potential future feature, the team decided to refrain from allowing it during the bot study in October 2021 due to concerns expressed by participants on the leaderboard.

alongside intraverting relations in HC-based CS. Trust, as I understand it in this next chapter, unfolds within human–technology relations.