Alien Flows: An Investigation of Computational Reason

A Division III thesis
by Ben Bailey

Advised by Christoph Cox and Lee Spector

Contents

Acknowledgments

This Division III would not have been possible without my committee, Christoph and Lee, my parents, Brian and Lori, and my friends, Jazlyn, Kristen, Libby, Brooke, and many more.

Abbreviations

CLoC	*The Cultural Logic of Computation*
CA	*Contagious Architecture*
ATP	*A Thousand Plateaus*

We have installed a backbone of glass and light that pulses throughout the world. Spinal surgery to replace rusting copper with pristine glass. The world was a patient etherized on the table, but the anesthesia is wearing off. Now, messages fly across the glass backbone before being flung out of antennae into the air, searching for their resting place in a nest of silicon nerve endings. Professor Challenger made the Earth scream, and so have we, installing and removing new spines and backbones over the course of the last century. We have installed a rhizomatic nervous system in the world and it is nervous.

I. Introduction: A Thousand Reasons

There has been no shortage of proclamations that the current era is the age of information. Largely driven by the near-miraculous decrease in size of transistors, the speed and capability of information transfer has increased rapidly over the last century. With the rise of Silicon Valley and technology industry giants such as Facebook, Google, Microsoft, Apple, and Amazon, computers and the algorithms that run on them have become central to the United States and greater world economies.

Not only are software companies and startups producing massive amounts of wealth by making things to be consumed through computers, they are using algorithms as the driving forces behind their main products. For many of these corporations, their rise to market dominance was sparked by (and remains fueled by) specific algorithms. Google owes much of its success to its development of the complex PageRank algorithm, which began as little more than a set of rules for following links and performing linear algebra operations on the results. Amazon and Facebook both built businesses from recommendation algorithms. Now, with the increasing prevalence of machine-learning algorithms, as more data flows thorough the networks of these companies, they are able to increase their recommendation abilities and, by extension, the data that gets stored with them. As a service gives users more accurate recommendations, users are more likely to use the service and give it their information, which then gives the service the ability to make better recommendations.

Because many websites on the internet are completely free to access and use (and because people expect services on the internet to be provided for free), the companies that create websites must find a way to turn a profit. A common solution that has proved successful for many websites is digital advertising. In addition to paying for traditional advertisement placement like on television or in print, digital advertising also enables website owners to charge more money for more effective advertisements made possible by those users of websites who agree to let website operators use their information to more efficiently target advertisements. Similarly, some websites make money by selling user data to advertising companies so that they can improve their own advertisements and placement algorithms. However, advertisers only want to buy advertisement space on websites that have lots of visitors and data about those visitors, so companies are incentivized to build products that keep users coming back to their websites and giving the network more of their information. By nature, platforms direct users toward behavior that is usable by those platforms. This results in a digital network economy built around garnering attention and directing human behavior. The cybernetic feedback loop of attention-directing and attention-quantifying that has emerged from contemporary social media platforms constitutes a sizable portion of internet activity.

The rapid rise in the usage of computers has coincided with an economic shift as well. While theorists have a variety of different names for this new economic system, from post-Fordism to cognitive capitalism to control societies, the description of each system centers around a shift from rigidity to flexibility, in which power becomes more decentralized but is still used for restrictive purposes.

Networks have played a central role in enabling this shift toward flexibility. With more workers gaining the ability to work from anywhere, more workers are expected to work from everywhere and be more flexible in their availability. The role that networks play in this "new spirit of capitalism" demonstrates the remarkable adaptability of capitalism as it has changed to work with flexibility and networks. It would astound many from the past to know that much of the world's communication is taking place over global networks, and yet it is possible that communication is as centralized as ever (though the loci of centralization have shifted). In the past, people often conflated networks with decentralization. This is why it is so disappointing to those who thought that networks would be liberatory; the tools that were supposed to be utilized for decentralization have. Already in 1990, French philosopher Gilles Deleuze foresaw the transition to decentralized power and the part that computer networks would play in it. In his short essay "Postscript on the Societies of Control," Deleuze identifies a transition from what Michel Foucault called "disciplinary societies," represented by the factory, to societies of control, represented by the corporation. According to Deleuze, while the disciplinary subject is a "discontinuous producer of energy," the control subject is "undulatory, in orbit, in a continuous network."[1] As the Western world transitions from disciplinary societies to societies of control, it is more and more involved with knowledge and emotion (affect), due in large part to the proliferation of computer networks.

These are not revolutionary statements—much has been written in the past decades about networks and their relationship with society, power, and control. However, less has been written about the nodes that constitute many of these networks, computational devices, and their status as philosophical objects. The world is not lacking in texts that discuss the topic of the increasing prevalence and power of computers in modern society. Initially, the advent of the Information Age promised an

---

[1] Deleuze, "Postscript," 6.

increase in the equality of information access and proliferation by creating a platform that everyone could easily and equally access and contribute to. Unfortunately, the information gap remains as large as it was before. Because computers are so adept at performing large, drawn-out calculations, companies and governments have discovered that combining data on human attention, behavior, and connections allows for the distributed prediction and direction of people on a scale previously unimaginable.

As theorists have come to this realization as well, they have attempted to find ways to subvert the predictive nature of computation. However, a majority of these theorists (such as Sherry Turkle in her book *Alone Together*) focus on the sociological causes and effects of the Information Age instead of examining the nature of information and computation. While David Golumbia's *The Cultural Logic of Computation* takes a more philosophical approach, the primary subject of the book remains the set of cultural beliefs and mechanisms that relate to computation and the spread of computers. The works that do address the philosophical aspects of computation itself (such as those of Alexander Galloway, Eugene Thacker, and Brian Massumi) mostly center on the idea that the rigid nature of the computational model and its reduction of the continuous nature of reality to discrete patterns of ones and zeros dangerously fails to reflect the way the world really is. Golumbia takes the now-traditional viewpoint that we should not allow computers as much influence over our lives, while Galloway and Thacker think we should exploit the formal structure of computation and networks to resist their encapsulating power. Golumbia's approach falls short because computation is capable of handling the things he believes it cannot, such as affect, language, and art, rendering his proposition impractical. Similarly, Galloway and Thacker's approach fails because they focus on the theoretical limitations of digitality while ignoring its practical accomplishments. As it has become more apparent that computers and networks are being utilized as mechanisms of control, theorists have proposed that we turn toward critical thought in an attempt to escape computational control. However, one must keep in mind the simultaneous turn toward flexibility in the Western economic system. There is some irony in the proposition that one should utilize affective thought to avoid the predictive and restrictive tendencies of a system increasingly capable of using affective thought to its advantage.

I claim that this view of computation and algorithms is outdated and fails to acknowledge somewhat recent developments in the fields of algorithmic information theory, nonlinear dynamics, and quantum computing. In order to demonstrate this, however, we must first answer the question of the ontological status of the algorithm.

The work of Italian philosopher and former Cybernetic Culture Research Unit member Luciana Parisi will serve as the starting point in answering this question. Parisi argues that neither Golumbia's nor Galloway's approaches are sufficient. Instead, she proposes that we need to think the nature of computation itself and view it as a form of speculative reason. At the center of Parisi's project is the proposal that computers possess their own form of reason that differs from human instrumental reason. Understanding this proposal and Parisi's argument for it requires a general understanding of ideas from A.N. Whitehead, Gilles Deleuze and Felix Guattari, Henri Bergson, and especially Gregory Chaitin, whose concept of Omega forms the foundation of her argument. Chaitin takes direct inspiration from both Gödel and Turing, and proves that there exists in any computational system an irreducible, incomputable property. For Parisi, this "irreducible infinity" is what makes algorithms fundamentally unique things. Part of the task of this project is to explicate the connections between these thinkers and Parisi for an audience not deeply entrenched in continental philosophy.

I wish to explain with a more technical background why Parisi's project is necessary, as well as its philosophical origins and its implications, in a way that is easier to understand for the lay reader. Upon examination, it seems Parisi's work is flawed in the perhaps over-zealous way she interprets Chaitin's work. In this thesis, I limit the scope of Parisi's project to a more specific class of algorithms and work to expand her conclusion that algorithmic thought is a unique, alien form of reason into a speculative prescription. Both the starting premises and the conclusions of Parisi's argument are individually compelling, and I will explore these and use them as a basis for an exploration of the potential of computation as a new form of reason.

Due to a common historical conception of computers as automated problem solvers, many theorists have critiqued computers as being "instrumental reason machines." Instrumental reason is, as defined by sociologist Max Weber, "determined by expectations as to the behavior of objects in the environment and of other human beings; these expectations are used as 'conditions' or 'means' for the attainment of the actor's own rationally pursued and calculated ends."[2] Importantly, in instrumental reason, the ends themselves are not critically examined. For example, when one employs instrumental rationality in attaching one piece of wood to another, one uses a hammer and nail as means to the end of attaching the wood. However, this amateur carpenter does not consider whether or not the two pieces of wood should be attached; perhaps there is a better solution that is more space-efficient or more aesthetically pleasing. While this is a simple, inconsequential example, the stakes become apparent when looking at issues of a larger scale. A world dominated by instrumental rationality is one in which any non-utilitarian moral system is unnecessary, or even incoherent. Thus, instrumental reason is critiqued as being uncritical, single-minded, and dangerous, particularly when living beings are used as means to an end as a result.

If computers were indeed pure instantiations of instrumental rationality, it would severely limit their usefulness, since there are many types of reason that could not be represented by them. While there certainly are many "undecidable" problems that cannot be solved by a standard digital computer, the machines are not limited

---

[2] Weber, *Economy and Society*, 24.

specifically by the type of reason required to solve a problem. It is possible to train a computer to judge works of art by an unknown mixture of parameters, but this does not mean it is judging the usefulness of the work as a means to some end. In this case, a computer is not necessarily utilizing critical rationality, but it is also not utilizing instrumental rationality. Instead, it exemplifies a unique type of reason that I will call "derivative rationality."

In accordance with the transition from first- to second-order cybernetics, and from disciplinary societies to societies of control, computation has also begun to operate at a higher order. In his essay "Facebook and Finance: On the Social Logic of the Derivative," Adam Arvidsson argues that the rise of Facebook coincides with the increasing domination of the derivative over goods in the world's financial markets. Arvidsson begins by citing sociologists Dick Bryan and Michael Rafferty's explanation of financial derivatives in simple terms as involving "deconstructing a 'thing'…into a set of constituent elements or attributes, and configuring those attributes in a way consistent with quantification."[3] Derivatives are based in abstraction and prediction, and when one views them this way, it becomes easier to see the connection to new computational paradigms. According to Arvidsson, just as financial derivatives allow the abstraction and quantification of previously non-determinate qualities, so too do social media platforms such as Facebook. Both processes of abstraction create a separate reality in which it is possible to make predictions around and comparisons between these abstracted quantities. Arvidsson uses the example of Facebook because it builds advertising profiles by abstracting qualities of individuals and using them to place them on the network's social graph and target advertising toward the users' quantified interests. However, I argue that this process applies not only to Facebook but also to a sizable portion of computational thought. In the example of judging a painting, the

[3] Bryan, Dick, and Michael Rafferty. "Financial Derivatives as Social Policy beyond Crisis." *Sociology* 48, no. 5 (2014): 892., Arvidsson, Adam. "Facebook and Finance," 5.

computer program takes a work of art and abstracts various qualities from it, creating a computational object and then assigning a value to that abstract object.

Following Arvidsson, I will term this dual process of quantification and valuation of previously non-valued things "derivation," and the mode of thought that enacts this process "derivative rationality," that form of reason that works to translate and approximate the qualitative into the quantitative, the human into the nonhuman, the analog into the digital.

Networks and computers have become increasingly involved in mediating our everyday lives, partially due to their convenience and partially due to their command of derivative rationality. In a society in which time is as valued as it is, computers have taken residence at the core of infrastructure and day to day life. As systems of control use computers to become more adept at capturing and utilizing affective thought, the total capture of human thought can seem frighteningly inevitable.

It is not that all forms of control are necessarily harmful, but a world in which everything is able to be controlled and predicted begins to lose its meaning. In the words of Russian theorist Viktor Shklovsky, "automatization eats away at things, at clothes, at furniture, at our wives, and at our fear of war."[4] If everything is predicted, then everything becomes familiar and gradually loses its meaning. In order to find novelty and the unpredictable, we must turn toward different modes of thought—alien forms of reason. To lift a phrase from Laboria Cuboniks, let a thousand reasons bloom.

---

[4] Shklovsky, "Art as Device," 5.

II. Instrumental Computation

Algorithms

Before we delve deeper into the ontological status of algorithms and computation, we must first determine how algorithms are usually defined outside of philosophical frameworks. To do this, we will combine two similar definitions.

In a textbook on algorithms for computer science students, Thomas H. Cormen et al write: "Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output."[1]

Similarly, philosopher David Golumbia cites John Goldsmith's definition of an algorithm: "Loosely speaking, an algorithm is an explicit and step-by-step explanation of how to perform a calculation…it is an operation that is guaranteed to work (that is, to be finished) in a finite amount of time (that is, a finite number of calculations)—though the specific value of the finite limit is likely to depend on the particular input that we

[1] Cormen et al, *Introduction to Algorithms*, 5.

choose to give to the algorithm."[2]

To integrate the two definitions, then, algorithms can be defined as series of instructions on how to transform inputs into outputs; discrete how-to guides for information processing.

What is notable about much of the current philosophical work around computation and algorithms is that it draws on a preconceived notion of algorithms and computation given by people who work in the field (computer scientists) or by cultural theorists rather than building an ontology from the ground up as philosophers do in many other areas. While there is no shortage of texts on the ontological status of technical objects,[3] philosophy as a field seems to have lost the thread when it comes to computational phenomena specifically. Philosophers take these definitions of algorithms from computer scientists without dissecting what they imply about the nature of computation and without considering that computation could be defined differently than it traditionally is. Golumbia's book is a prime example of this.

Computationalism

David Golumbia's book *The Cultural Logic of Computation* is a quintessential example of a critique of computation as instrumental reason.[4] Golumbia argues that we must separate computationalism (a philosophical doctrine and the main subject of Golumbia's book) and computerization (the proliferation of digital computers in the world). Golumbia's position toward computerization and computationalism draws

---

[2] CLoC 39-40. Golumbia's work will be examined in greater detail in the next section.

[3] Heidegger and Simondon first come to mind.

[4] It is important to note that Golumbia's book was published in 2009, when mass computerization was less prevalent than it is today. The third iPhone had just been released with competition for the first time, Twitter was growing rapidly and played a big role in reporting on the Iranian election protests, the technology world was experiencing an "app boom," Netflix and Spotify were in the process of changing the way media is consumed, and cloud computing giant Amazon Web Services was just getting started in earnest.

heavily from a poststructuralist background, and as a result much of the book focuses on the social and political implications of these phenomena.

Many of Golumbia's critiques focus on the culture surrounding computation (that is, computationalism), and I do not have significant qualms with those. What I do take issue with are his claims regarding the nature of computation itself. One may argue that Golumbia's claims about computationalism are synonymous with his claims about the nature of computation—indeed, he does seem to believe that the two are strongly linked. However, I propose that our social beliefs about computers are what shape our popular conception of computation, not the other way around. What follows, then, is a critique of the theory of computation implicit in Golumbia's work.

Golumbia's central thesis is that the positive political and social impact of mass computerization is greatly overstated. When the internet was first becoming popularized, it was a traditional stance of so-called "computer evangelists" such as Nicholas Negroponte (cofounder of the MIT Media Lab and early columnist for *Wired*) that the mass adoption of computers would lead to a flattening of the political and social worlds; that, due to the network-centric and individualistic nature of computerized communication, the hierarchies present in the world's power structures would be deconstructed. In the epilogue to his 1995 book *Being Digital*, Negroponte writes:

> But more than anything, my optimism comes from the empowering nature of being digital. The access, the mobility and the ability to effect change are what will make the future so different from the present. The information superhighway may be mostly hype today, but it is an understatement about tomorrow. It will exist beyond peoples wildest predictions. As children appropriate a global information resource, and as they discover that only adults need learners permits, we are bound to find new hope and dignity in places where very little existed before.[5]

This viewpoint was particularly popular in the 1980s and 1990s, but continues to be

[5] Negroponte, *Being Digital*, 231.

espoused today, particularly by Silicon Valley investors and employees.[6]

Despite many still maintain that mass computerization has led to a paradigm shift in both human thought and human kind in general, Golumbia believes that "we must assume that technological shifts are best seen as changes in degree and not in kind; that human beings remain what they are…and that human societies, too, remain largely bound by much the same fundamental forces by which they have always been characterized."[7] In other words, humans often overreact to changes brought about by technology despite those changes not dramatically altering the way society works. Golumbia finds it apparent that mass computerization, while exceeding expectations of democratization in some areas, has largely failed to live up to the hopes of mass democratization that were widespread in earlier writings on the subject (such as Negroponte's). Although the fact that the image of the Internet as a powerful force of democratization and decentralization remains prevalent, many are beginning to realize

---

[6] "The internet is too transformative for incumbents to not want to try to stifle or curb it – incumbents in the sense of multinational corporations, governments, take your pick." –Reddit founder Alexis Ohanian (Meek, Andy. "Reddit's Alexis Ohanian: 'Seeing the Obama Letter Just Blew Me Away'." The Guardian. Guardian News and Media, April 26, 2015. https://www.theguardian.com/media/2015/apr/26/reddit-revenge-porn-privacy-alexis-ohanian.)

"Millenials, and the generations that follow, are shaping technology. This generation has grown up with computing in the palm of their hands. They are more socially and globally connected through mobile Internet devices than any prior generation. And they don't question, they just learn…Millennials' tech and global savvy will make them instrumental in shaping our mobile future worldwide." –Intuit CEO Brad Smith (Schawbel, Dan. "Intuit's Brad Smith: How to Succeed in a Bad Economy." Forbes. Forbes Magazine, November 1, 2012. https://www.forbes.com/sites/danschawbel/2012/10/26/intuits-brad-smith-how-to-succeed-in-a-bad-economy/#462acf937649.)

"The Internet has changed everything. We expect to know everything instantly. If you don't understand digital communication, you're at a disadvantage." –Go Daddy founder Bob Parsons (Scharper, Julie. "Go Daddy Founder Gives Advice to University of Baltimore Students." The Baltimore Sun, November 13, 2012. http://www.baltimoresun.com/news/maryland/baltimore-city/bs-md-ci-go-daddy-founder-20121112-story.html.)

[7] CLoC 2.

that the democratization of information can, in many instances, be a net negative.[8]

Indeed, one cannot ignore the abilities that mass computerization enable for forces of control. Mass surveillance, information fabrication, and automated warfare are all made much easier by the spread of computers and networks across the globe. Observing this, Golumbia goes as far as to claim that computerization and computationalism *most* benefit those in power.

To help support this claim, Golumbia taps into a longstanding tradition which argues that instrumental reason is insufficient and inferior to critical reason by claiming that "computationalism meshes all too easily with the project of instrumental reason"[9] and is therefore insufficient for liberatory purposes. Indeed, the status of reason has major implications for the status of the algorithm. To understand Golumbia's claim, we must first take a brief detour to discuss different types of rationality.

Computation and Instrumental Reason

The German sociologist Max Weber defined four different "orientations" of social action: instrumentally rational, value-rational, affectual, and traditional. While Weber's theory has been criticized for being biased or excluding other types of social action, his framework still provides a useful distinction between instrumental reason and value-based reason (I am more interested in using Weber's categories as a starting point for this discussion than presenting it as the only valid framework of rationality). Weber debates whether or not the last two types of social action actually qualify as "meaningfully oriented action" and focuses on the first two, so I will do the same here.

First, Weber presents *instrumental rationality*, which is "determined by

---

[8] For instance, much has been written recently about the idea that American society is "post-truth," and much of this writing places the blame for this on the anonymity and equality of information on the internet.

[9] CLoC 5.

expectations as to the behavior of objects in the environment and of other human beings; these expectations are used as 'conditions' or 'means' for the attainment of the actor's own rationally pursued and calculated ends."[10]

In contrast to instrumental rationality, an action is oriented by *value rationality* when it is "determined by a conscious belief in the value for its own sake of some ethical, aesthetic, religious, or other form of behavior, independently of its prospects of success."[11]

Instrumental reason focuses on using things as means to desired ends in the most efficient way possible. Critical theory, a theoretical tradition which originated from the Frankfurt School and draws on ideas from Kant, Hegel, Marx, and sociology, generally criticizes instrumental reason as being insufficient. According to critical theorists, instrumental reason becomes harmful when applied to beings that should not be used as means, such as humans. Instead, critical theory proposes value-based examinations of ends to determine if they are morally justifiable before taking actions to realize them.

Looking at these definitions, it is not difficult to see the potential link between instrumental reason and computation. This is especially true when reviewing the history of the computer as a mathematical machine. When taking into account the popular conception of mathematics as purely rational and unbiased, it is not surprising that the computer would be seen as the ultimate reasoning machine, and, by extension, a pure instantiation of instrumental reason. According to this view, computers are only machines that one uses as an instrument for the end of rationality by giving it a logical proposition and receiving the truth value of that proposition as an output.

Due to the popularity of this view, computer users in the western world tend to think of computers as purely rational machines that produce guaranteed results, and

---

[10] Weber, *Economy and Society*, 24.
[11] Weber, *Economy and Society*, 24-25.

one can see this in the traditional definition of algorithms as well. As a trend of computers aiding the rise of predictability and control has begun to emerge, some have proposed that we turn toward non-instrumental, uniquely "human" thought to escape territorialization. However, the most powerful and influential computations today are more complicated and complex, taking advantage of feedback loops, systems thinking, and research into the human brain to predict and direct human behavior even more powerfully than was thought to be possible for machines. Through these new avenues of computation, the apparatus of control has expanded. This demonstrates both the power of control, in that corporations are using computers as means to an end (instrumentally) to capture human behavior, and simultaneously that computation is capable of more than just instrumental reason.

Regardless, the intrusion of computation into the realm of the previously unpredictable and uncontrollable demonstrates that the solution that critical theory proposes to "escape" the consuming force of automated instrumental reason, seeking out human forms of reason such as value-based rationality that cannot be incorporated by computerized instrumental reason as means to an end, is not actually effective. Control can and does use value-based reason as a means to an end, rendering the proposition of using value-based reason to escape control insufficient.

Neither humans nor computers are limited to Weber's four orientations of social action. However, if humans cannot understand the reason a computer performs an action, it is difficult for humans to categorize that action. While it is simple to attribute much of a computer's behavior to the pre-programmed routines that are given to it, it is more challenging to categorize the rationality that a computer is following when it performs an action we cannot understand, as in the instance of high-frequency trading algorithms causing sudden market crashes despite being designed as means for the end of generating profit. If we follow the hypothesis that every action is done for a reason, then this computational reason is alien to us by definition alien if we cannot comprehend it. We need to stop thinking about computers as means that are part of some instrumental rationality and start thinking of them as exhibiting types of rationality themselves. I propose that the primary type of rationality that computers and complex algorithms exhibit today is derivative rationality. Derivative rationality is a mode of thought that is incomprehensible by humans, and therefore by humans that attempt to use computers as tools of control.

I am not claiming that computers exhibit value-based rationality—the derivative rationality that computers do exhibit is quite different. While both types of reason involve the valuation of things, derivative rationality abstracts, translates, and extracts value from things, whereas value rationality simply assigns value to things. In contrast to value-based reason, derivative reason creates a new, abstract, and digital thing from qualities of a thing, while value-based rationality holistically assigns value to things. Complex computations create a film around real objects, a membrane of abstract qualities, a sort of digital image, before discarding the real object because it cannot be quantified.

Golumbia follows in the footsteps of critical theory, quoting Jacques Derrida: "Which is as much as to say that madness, a certain 'madness' must keep a lookout over every step, and finally watch over thinking, as reason does also."[12] According to Golumbia, "we do not know how to formalize this 'madness,' this 'irreason'…Most importantly, we do not want to know how: it is critical…to our social practice itself that we as social beings can escape whatever formalization we manufacture."[13] Here, Golumbia proposes (following Derrida) that there is some level of human thought that both cannot and should not be formalized, a higher meta-reason. Unfortunately, this claim is disputed by contemporary algorithms, which both formalize types of reason

---

[12] CLoC 80.
[13] CLoC 80.

outside of instrumental reason and create new, meta levels of meaning from qualitative things. However, it is too late for many things such as human behavior to avoid control by formal systems. The "irreason" Golumbia believes we do not know how to formalize can be predicted and created by automated formalization itself—computation is already mad.

Now that I have established that computational reason is fundamentally different to instrumental reason, I will present my second critique of Golumbia, which regards his claims about computers and striation.

Computation and Striation

Golumbia's second (slightly more theory-based) claim regarding the nature of computation is that computation is a force of striation. To prove that computers function more as tools of control than of resistance to it, Golumbia incorporates Deleuze and Guattari's concepts of striated and smooth space into his analysis. Deleuze and Guattari develop the notion of smooth space from the image of physical spaces that lack easily identifiable points of demarcation, such as oceans or deserts. Considering only this definition, it is possible that one would be lured into thinking that smooth space is inherently more liberatory than striated space, and that capitalism, as a system of control, is inherently striating. However, this is not the case.

Golumbia contrasts striated space ("the space of the State, of firm bureaucratic and governmental orders, of the grid, of maps, coordinate orientations, of territorialization, tree-like…organization, settlement and agriculture")[14] with smooth space (the space of "nomads, hunter-gatherers, navigation by bearings rather than maps, intuition,…deterritorialization, rhizomatic organization, relatively anarchic and

[14] CLoC 23.

local forms [of] government, and relatively mobile forms of life")[15] and uses this distinction to claim that computers are forces of striation and therefore primarily instruments of control. However, as Deleuze and Guattari point out, "the two spaces in fact exist only in mixture: smooth space is constantly being translated, transversed into a striated space; striated space is constantly being reversed, returned to a smooth space."[16] Since striated and smooth are simply poles of a continuum, every space is both smooth and striated to some extent.

In the words of Deleuze and Guattari, "smooth spaces are not in themselves liberatory…Never believe that a smooth space will suffice to save us."[17] However, Golumbia is in danger of exactly this in believing that smooth space will "save" humans from computerization when, in reality, computerization is directing humans toward the smooth space of control.

Upon further examination, it becomes clear that control is not limited to forces of striation. As Michael Hardt and Antonio Negri write in their book, *Empire*, "capital tends towards a smooth space defined by uncoded flows, flexibility, continual modulation, and tendential equalization."[18] This is in step with what Deleuze posits as the transition from Foucault's disciplinary societies to societies of control. According to Hardt and Negri, "in the passage to the society of control, the elements of transcendence of disciplinary society decline while the immanent aspects are accentuated and generalized."[19] In other words, the controlling force present in societies of control arises immanently from individuals and their interactions instead of being projected down onto them from above as in disciplinary societies.

---

[15] CLoC 23.
[16] ATP 474.
[17] ATP 500.
[18] Hardt and Negri, *Empire*, 327.
[19] Hardt and Negri, *Empire*, 331. The notion of immanence of control is also central to Alexander Galloway's conception of protocol, which will be discussed shortly.

Not only is Golumbia's assumption that striation is always more "negative" than smoothness incorrect, his assertion that computers and computation are inherently striating is also flawed. While Golumbia is correct in observing that computers provide the ability for worldwide striation on a scale perhaps unseen before,[20] he fails to account for the part of the computational process in which abstract qualities, having been quantified, are brought into a smooth computational space. Much as real space is smooth until it is striated, so is the new computational space.

Computation, then, can be seen as a mixture of smooth and striated spaces just like everything else in the world. Although computation does draw lines between real qualities by nature, it also creates a new smooth space out of the quantities that are derived from these abstract and translated qualities. This is in direct contrast to Golumbia's claims that computation is a purely striating force, and that striating forces are always forces of control, both claims that have been disputed here.

The third and final argument of Golumbia's that will be addressed here is that humans should not only resist through protocol, but also against it, in direct disagreement with Alexander Galloway. To further examine this assertion, I will first define protocol and examine Galloway's argument regarding it.

Computation and Protocol

While much has been written about the rise of networks, another digital media theorist Alexander Galloway provides a particularly pertinent explication of relevant concepts. To understand Golumbia's critiques of Alexander Galloway's proposed political response to computationalism, we must first explicate Galloway and Eugene Thacker's project in their book, *The Exploit*.

---

[20] One example of computers as striating machines is GPS and efforts like Google Maps, which literally provide users with gridded (striated) images of the smooth world. However, this claim has to do more with the striating effects of computerization than with the striating nature of computation itself. It is not as if creating a gridded map of the world was impossible before the advent of computation.

Galloway and Thacker's key thesis in *The Exploit* is not entirely dissimilar to Golumbia's. They identify an issue in contemporary society where "technology is assumed to…preexist politics,"[21] which echoes Golumbia's claims. For Galloway and Thacker, "networked power is based on a dialectic between two opposing tendencies: one radically distributes control into autonomous locales; the other focuses control onto rigidly defined hierarchies."[22] In Golumbia's eyes, the internet as a network has exhibited the latter tendency much more than the former. However, Galloway and Thacker's project is based on more sound assumptions on the nature of computation.

In order to understand Golumbia's issue with Galloway and Thacker's claim regarding networked power, we must first understand Galloway's conception of "protocol." According to Galloway and Thacker, "the concept of 'protocol' refers to all the technoscientific rules and standards that govern relationships within networks."[23] Later, they offer another definition: "a horizontal, distributed control apparatus that guides both the technical and political formation" of various networks.[24] Both these definitions have in common the idea of conditioning the shape of networks. Protocol is, for Galloway, the set of pre-established rules that guide and control communication over a network. Framed differently, drawing on Deleuze's influential essay "Postscript on the Societies of Control", Galloway writes that "protocol is how technological control exists *after* decentralization."[25] Protocol is the "manager" of control society (bureaucracy and hierarchy are the managers of disciplinary and sovereign society, respectively).[26]

To better illustrate the concept of protocol, Galloway contrasts the Transmission

---

[21] Galloway and Thacker, *The Exploit*, 10.
[22] Galloway and Thacker, *The Exploit*, 19.
[23] Galloway and Thacker, *The Exploit*, 28.
[24] Galloway and Thacker, *The Exploit*, 28.
[25] Galloway, *Protocol*, 8.
[26] Galloway, *Protocol*, 27.

Control Protocol/Internet Protocol (TCP/IP) networking stack[27] with the Domain Name System (DNS) protocol.[28] The internet is typically modeled as a series of layers. Each layer has its own specific header and footer, predetermined strings of bytes that contains information like the source and destination of the data, the time it was sent, and the length of the content. When data needs to be transferred, it is first wrapped in the header and footer of each layer, then passed down to the next. When it reaches its destination, it passes up through the layers again, and the header and footer of each layer are peeled off one by one in order to make the data readable by the next layer. IP takes care of addressing nodes on a network and determining routes to other nodes, while TCP actually manages the transport of data between two nodes. As Galloway mentions, IP is described as using an "anarchic and highly distributed model."[29] While TCP/IP are decentralizing protocols, DNS is a centralizing one.

DNS provides a very useful service by assigning memorable names to numerical IP addresses, like an address book for the internet. The DNS protocol defines rules for interaction between DNS "clients" and DNS "servers."[30] This is what allows one to visit "google.com" in a web browser instead of "172.217.11.46." The issue with having centralized a naming system, however, is that it adds a central point of failure to the previously decentralized network. This is a flaw with the internet as a whole as well, since there are points of failure at the routing layer at places where major physical (particularly intercontinental) infrastructure meets. If cables were to be cut there, it

---

[27] This pair of protocols is also referred to as the DoD (Department of Defense) model.

[28] While DNS is not traditionally referred to as a protocol in and of itself, it (like most traffic on the internet) relies on a transport protocol (in this case UDP). There is also an actual DNS protocol, which DNS relies on.

[29] Galloway, *Protocol*, 8.

[30] The client/server model is a common method of structuring network infrastructures. In this model, several "clients" interact with one "server," which can interact with other servers. A good example of this is email, where a user runs an email client that gets messages from and sends messages to a more central email server.

would sever an entire part of the world's ability to communicate with other continents.

The example of internet protocols is useful in that it demonstrates the dual nature of protocol: it can be centralized, but it can also enable decentralization. In some ways, protocol is the primary factor in making decentralization possible. It is safe to say that without protocol the internet as we know it could not exist, and, while some may prefer that it not exist, Golumbia does not seem to be one of those people. He advocates for a "possible future in which computers are more powerful, more widespread, cheaper, and easier to use—and at the same time have much less influence over our lives and our thoughts."[31] Though this is a future in which the internet is less influential, it is not one in which the internet has been completely destroyed. For this to be possible, protocol must still exist; as Galloway writes, "without a shared protocol, there is no network."[32] Protocol cannot be dismantled without the internet being dismantled along with it.

However, despite protocol enabling decentralization, it also does work as a force of control by definition. In Golumbia's proclamation for resistance against protocol, then, he is attempting to "raise the question whether the shape, function, and ubiquity of the computing network is something that should be brought under democratic control in a way that is not today."[33] However, Golumbia conflates protocol as a form of control and protocol as an inherent aspect of the way computers function, writing "what I want to articulate is the case precisely for resistance *against* what Galloway calls protocol, and what is more generally thought of as computerization."[34] Golumbia's argument against Galloway falls short in part because he does not begin from a correct definition of protocol. Computerization could not exist without protocol, but protocol

---

[31] CLoC 26.
[32] Galloway, *Protocol*, 12.
[33] CLoC 25.
[34] CLoC 25.

could exist without computerization. Where there is a decentralized network, there is protocol, and where there is protocol, there is control. While it is true that protocol *enables* technological control in the sense that Golumbia means, this does not mean that protocol *is* technological control. This confusion results in Golumbia viewing protocol as a strict negative, despite Galloway reminding the reader that protocol can be either positively or negatively valenced, depending on "varying degrees and contexts."[35] This is also what leads to Golumbia's disagreement with Galloway's proposition that "it is through protocol that one must guide one's efforts, not against it."[36] Galloway sees resistance against protocol as impossible because protocol is immanent and emergent by nature, while Golumbia sees protocol as analogous to computerization, thereby falling into protocol's trap. Due to the manner in which it arises, the only way to escape the control of protocol is through it.

In this chapter, I have demonstrated that Golumbia's suppositions about technological control (protocol) are incorrect, as are his claims about computation being an instance of instrumental reason and his assertion that computation is always striating. The flaws in Golumbia's argument come from assuming incorrect things about the nature of computation. Having refuted these claims and demonstrated why this manner of thinking about computation is insufficient, we can now turn to an examination of a theory of computation that acknowledges computation in itself, that of Luciana Parisi.

---

[35] Galloway, *Protocol*, 246.
[36] Galloway, *Protocol*, 17.

III. The Parisian Turn

Cybernetics and Chaitin

The previous work in the field of computational reason is relatively limited. Perhaps the most prominent work in the area of is that of Luciana Parisi, a former member of the Cybernetic Culture Research Unit at the University of Warwick and reader/convenor of PhD Cultural Studies at Goldsmiths University of London. Parisi's work will be central to the rest of this thesis, and so it will be prudent to dedicate some time to understanding her project.

In her book, *Contagious Architecture*, Parisi presents two different conceptions of computation through the lenses of first- and second-order cybernetics. According to first-order cybernetics, computation is a closed formal system that predicts the future "in terms of preset probabilities."[1] In this theory, "all systems…[are] probabilistic entities determined by their capacity to organize and control information through a homeostatic rebalancing of energy and information."[2]

Second-order cybernetics takes this a step further and includes "biophysical

[1] CA 10.
[2] CA 261.

indeterminacy" in the previously closed system to increase the capacity of a computational system to generate chaotic or complex behavior, which makes it easier for computers to solve certain problems. It is based in the idea that "all systems entertain a positive feedback relation between the individual and the environment, but also between a first and second level of perception."[3]

Parisi's broad project throughout the book is to achieve a sort of "third-order" cybernetics that acknowledges the randomness and complexity inherent in computational systems without ignoring that complexity and claiming that algorithms are just simple formal systems or attributing that complexity only to biophysical processes. By acknowledging that complexity comes from the algorithmic form itself, Parisi hopes to prove that algorithmic thought is separate from automated instrumental reason.

To effectively understand and evaluate Parisi's argument, we must turn to the work of the Gregory Chaitin, the Argentine-American mathematician. Chaitin made important contributions to the field of algorithmic information theory, which studies the complexity of computer programs (and, by extension, mathematical objects). Algorithmic information theory interprets everything as a "string," a list of objects of the same type (traditionally alphanumeric characters). "Kolmogorov complexity" is the measure of complexity that algorithmic information theory uses to describe the world. The Kolmogorov complexity of a string is defined as the length of the smallest computer program that produces the string as its output. For example, the string "ABABABABABAB" is less complex than the string "AABABAAABBAB", because the first can be described as "AB" repeated six times (or "AB"*6), while the most efficient way to produce the second string is presumably to repeat the string itself. It is possible that there is some more efficient method of reproducing the second string, but for this

---

[3] CA 261.

purpose, we will assume that it is incompressible. This interpretation of complexity leads to a definition of a random string as a string which is its own shortest representation, like the second string above.

While these definitions at first seem trivial, their utility may become clearer when one considers that computer programs can be expressed as strings as well. To compute the Kolmogorov complexity of a string or program is not an easy task, though the process itself is easy enough to describe.

Let us say we have a computer program which tries to compute the complexity of a string *s* that has a length of *n*. To do so, the program will run every program that has a length shorter than *n*. If the smaller program in question produces our string *s*, then *s* is not algorithmically random (since it was able to be produced by a program with a smaller length than it) and the complexity of *s* is the length of the program that produced it (due to the definition of complexity). However, some of the programs that are tested will never stop running, which means that our complexity calculator will never return a result and instead run infinitely. We cannot exclude these infinitely-looping programs because to figure out which programs never stop would be to solve Turing's halting problem, which Turing proved was impossible to universally solve.

Due to this connection, Chaitin's work is strongly intertwined with that of mathematicians Kurt Gödel and Alan Turing. Gödel's incompleteness theorem (the idea that every formal system is either incomplete or inconsistent) caused a fundamental shift in the fields of mathematics and computation. Turing was attempting to find a general solution to the *Entscheidungsproblem*, the question of the existence of an algorithm that can universally determine the truth value of a logical proposition. To do this, he created the concept of a "computing machine" (or Turing machine) and posed a separate problem: the halting problem. The halting problem is the problem of figuring out whether an arbitrary computer program will finish executing given a random input. Turing proved that the halting problem could not be solved universally by any such machine. That is, he proved that there exists no Turing machine that can universally determine whether any arbitrary computer program will stop or run ad infinitum. In doing so, Turing also proved that there is also no universal solution to the *Entscheidungsproblem*.

The notion of halting is essential for understanding Omega, as Chaitin defines Omega as a "halting probability" or the probability that "a program chosen at random [will] ever halt."[4] Importantly, Omega is perfectly well-defined, but cannot be calculated by a program of fewer bits than itself and is thus incomputable. One can know that Omega exists in an algorithm, but it is extraordinarily difficult to compute. Chaitin concludes his article by proposing that, because there is no possible grand universal theory of mathematics that can be described, mathematicians should be more willing to postulate and test new axioms like physicians.

Chaitin defines a computer program as "elegant" if it is the "smallest program that produces the output that it does" or, equivalently, if "no program smaller than it produces the same output."[5] Unfortunately, it is not possible to prove that a program is elegant if it is larger than the program which generates the theorems of the formal system which one is working in. In order to prove that a program is elegant, one would have to produce the output of the program. If the input program is larger than the evaluation program, it cannot be proven to be elegant, because the evaluation program would have produced the same output through a program with a shorter length than the input program, meaning that it is not elegant. If the input program is elegant, the evaluation program must have produced it in its determination process. However, if the smaller evaluation program produced the input program, then the input program cannot be elegant.

---

[4] Chaitin, "The Limits of Reason," 78.
[5] Chaitin, *Meta Math!*, 108.

Essential to understanding Chaitin's theory is the notion of an "axiom." Axioms are mathematical facts that are true for no derivable reason, that is, they cannot be derived from the other theorems present in a formal system. We generally take axioms to be true because it is beneficial to take them as true.

There is a strong link between Chaitin's theory and attempts to find a universal "theory of everything," that is, a formal scientific theory that can be used to explain the universe in its entirety. Scientific theories are, in essence, a form of compression. A useful theory allows one to produce theorems, that is, things that are true, from a smaller number of theorems and axioms. In other words, a theory acts as a sort of compression algorithm—it takes many facts and reduces them to fewer facts that allow other facts to be deduced. A theory would not be useful if it was as equally complex as the things it was attempting to describe. Unfortunately, according to Chaitin, there can be no formal universal theory that accounts for the universe due to the incapability of formal theories to account for incomputable, infinitely complex information. Simple sets of rules such as Stephen Wolfram's Rule 30 cellular automaton[6] that generate complex behavior do not meaningfully account for everything in the universe (or even in the space of computable things) because the rules must be played out step by step for the resultant facts to be reached. The "axiom" of Rule 30 only compresses a process to discover truths, not those truths themselves.

Following Chaitin's ideas, every formal computational system possesses well-defined but incomputable and algorithmically random quantities (Omega). Parisi uses this fact to claim that there can be no complete "metacomputational ontology" because

---

[6] A cellular automaton is a set of cells on a grid that evolves discretely according to rules which govern the evolution based on the states of the neighboring cells. Wolfram's Rule 30 is interesting because it generates chaotic behavior from a small set of rules. Other examples include Conway's Game of Life. For more information, see Weisstein, Eric W. "Cellular Automaton." *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/CellularAutomaton.html.

"any closed set of finite algorithms is imbued with incompressible data".[7] The question is whether or not this incompressible data is meaningfully present in every closed set of finite algorithms.

Parisi's basic argument can be compressed into this: algorithms have inherent irreducible complexity not just through their association with biophysical inputs and outputs (as second-order cybernetics believes) but due to their formal nature. They are both abstract and actual. According to Parisi, algorithms do not calculate or compute incomputable data, but instead "prehend" them.[8] When Parisi calls algorithms conceptual prehensions, she means that algorithms cannot help but "feel" incomputable data. These incomputable data are not actual occasions, but instead eternal objects. This prehension does not rely on the physical reality of data, as second-order cybernetics believes. Prehension is a non-anthropocentric term, which importantly enables Parisi to bypass the necessity of proving that computers can perfectly emulate human thought. It opens the door for Parisi to posit algorithmic thought as a separate, alien form of thought.

Although this theory of computational reason that Parisi presents is the most accurate and cohesive account we have, there are some flaws in her argument that can be improved upon with some revisions.

Brief Revisions

While until now I have largely aligned myself with Parisi's position, I would be remiss if I did not clarify some objections that have arisen from my engagement with her work.

---

[7] CA 7.

[8] Philosopher Alfred North Whitehead explains his reasoning for developing the concept of prehension as follows: "The word *perceive* is, in our common usage, shot through and through with the notion of cognitive apprehension. So is the word *apprehension*, even with the adjective *cognitive* omitted. I will use the word *prehension* for *uncognitive apprehension*: by this I mean *apprehension* which may or may not be cognitive" (Whitehead, *Science and the Modern World*, 70).

Once these objections have been clarified, I will present a revised theory of computational reason that addresses them.

First, Parisi struggles sometimes with providing clear definitions of the terms she is using or creating. When terms are defined, they tend to have several overlapping, but not entirely similar meanings throughout the course of the text. This makes it somewhat difficult to decipher the core argument of the book.

A prime example of this occurs early on in *Contagious Architecture*, beginning from the distinction she draws between first and second-order cybernetics. It is not problematic to draw a distinction between these two different methods of cybernetics, but the subtleties of this distinction itself remain unclear. Parisi uses second-order cybernetics to refer to a system where an individual interacts with its environment and the environment is included in the individual's feedback loops, and then draws a contrast between this and a new mode of speculative thought that she calls "soft thought," as exemplified partially by parametric architecture[9] (in which the parameters are so-called incomputable quantities, that is, incomputable real numbers). However, it does not seem that second-order cybernetics differs meaningfully from parametric architecture. The way that second-order cybernetics is defined, in combination with the fact that Parisi wishes to move beyond this toward a sort of "third-order" cybernetics, complicates her argument. Because Parisi uses parametric architecture as an example, the lack of clarity with where it falls between second- or third-order cybernetics is confusing.

Additionally, Parisi frequently draws on the work of Gregory Chaitin without citing specific passages, which leads to difficulties in figuring out where Parisi is making inferences from Chaitin's work as opposed to directly stating his argument. The line between Parisi's and Chaitin's arguments is blurred.

---

[9] Parametric architecture is broadly defined as a style of architecture that uses computer software to optimize structures based on various parameters such as noise and cost.

For instance, Parisi writes that Chaitin claims that "the output is always greater than the input" in "every computational process", but I have not been able to find a source for this claim in Chaitin's work.[10] Particularly in her essay "Instrumental Reason, Algorithmic Capitalism, and the Incomputable," Parisi frequently draws on the idea that the amount of incomputable information in a computational system is always increasing, but this claim is unsubstantiated. Although incomputable things certainly exist, and, following Chaitin's work, it is true that almost all real numbers are incomputable, it is difficult to see why this means that the amount of incomputable data within a computational process is always increasing.

The claim that the incomputable is present in every computation is incorrect. It is simple to construct a program that will always halt, or even one that will always halt with the same answer regardless of the input. As a basic example, one could easily create a program that takes any input and simply discards it and returns "0" to the user as an output.

Similarly, I disagree that all algorithms necessarily possess a unique "second nature." This claim should be limited to algorithms of unique construction in unique circumstances. Despite the fact that incomputable data may be "rather central" to many computations today, it is not true that it is central to all computational processing.

Therefore, I disagree with the claim that "the more [a computation] calculates, the more randomness…it creates."[11] I do not believe that every computation generates randomness by nature. Instead, I believe Parisi's weaker claim that computation can in principle incorporate and approximate the incomputable or the random in its calculations, and that this implies the existence of a mode of thought, one that is separate from human thought since it does not really have a notion of means or ends.

[10] Parisi, "Instrumental Reason," 132-133.
[11] Parisi, "Instrumental Reason," 136.

To obviate the need to prove that the incomputable is at the center of all algorithms, I will restrict Parisi's claims to a more limited category of algorithms. An algorithm of this specific class requires interaction between the three levels of cybernetics, that is, it must have internal feedback loops (some element of recursion), it must interact with the environment (forming an open system), and it must attempt to translate generally incomputable things into computable ones. I will call these algorithms "derivative algorithms," for lack of a better term. Not all algorithms can exhibit derivative rationality, only these.

A common example of this type of algorithm is high-frequency trading (HFT) algorithms, which finance companies use to take rapid action in buying and selling stocks based on patterns in the market and the news. High-frequency trading algorithms are ultimately attempting to represent, predict, and profit from human behavior.

Another prime example is Amazon's recommendation algorithms. Amazon uses what it calls "item-to-item collaborative filtering" to recommend products to users based on products that they have demonstrated interest in. Instead of focusing on matching users based on purchase similarities and then recommending items that one has purchased and the other has not, Amazon focuses on reducing data on customer behavior to the similarities of various items. Traditionally, sites compared the similarity of users, but Amazon compares the similarity of individual items based on the users' behavior, allowing them to compute recommendations more efficiently.[12] It is important to remember here that the computer itself is not aware of the difference—it is deriving insights about human patterns regardless.

Finally, Facebook's news feed algorithm is perhaps the most controversial of these, and the one most discussed in the public sphere. Facebook's algorithm is

---

[12] Linden, G., B. Smith, and J. York. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering." *IEEE Internet Computing* 7, no. 1 (2003): 76–80. doi:10.1109/mic.2003.1167344.

designed to present the user with information that entices the reader to continue actively engaging with Facebook. It captures data about what each user likes and dislikes, what catches their attention, and what kind of people they are friends with. This data is then fed into neural networks[13] to help fit users into profiles.

All three of these examples involve large amounts of computing power taking in "incomputable" data about human behavior from their "environment" and using it to generate results that affect human behavior. These results are then fed back into the algorithm, forming a feedback loop. Whereas Parisi focuses on the unpredictability of complex algorithms as arising from strict formal processes, I believe this unpredictability arises from the nature of the abstraction and translation that these algorithms perform. This obviates the need for unconvincing claims about the formal nature of computation itself, while still not ignoring computation itself. It is therefore true that computers cannot be reduced to tools of instrumental reason, but for different reasons than Parisi believes.

This is not to say that Parisi's project is not incredibly valuable. I believe that one of the most ideas that one can take from Parisi's work is that computers think, but not the same way humans do, and that it is naïve to assume that computers have the same reason as humans. Despite the argument's flaws, Parisi provides a compelling case for examining computation as a form of speculative thought. Although her argument that every algorithm is centered around the incomputable is flawed, the idea that computation can interpret the incomputable is powerful and unique from a

[13] Neural networks are loosely modeled after human brains, but the interpretation that they are an exact representation has greatly fallen out of favor. They consist of layers of artificial "neurons" (or nodes). Each node in the network has a weight assigned to it, and multiplies each of its input values by the weight and adds them to calculate an output value. If the value is greater than a threshold, the output is passed to the next neurons in the network. These networks are then "trained" on data supplied to them in which the weights and thresholds are adjusted to produce better, more accurate results. Neural networks are frequently used today in the detection of objects and people in images, as well as many other areas of artificial intelligence.

philosophical perspective. With these objections in mind, I will now propose my revised theory of computational reason and demonstrate how it rectifies some of these issues.

IV. Derivative Reason

The Possible and the Virtual

Before I continue with my examination fo derivative rationality, I must introduce one final set of terminology, the Bergsonian distinction between the virtual and the actual. This will be important in the coming examination of the implications of derivative rationality and its intersection with creativity and novelty. To fully realize Bergson's distinction, however, we must first understand his notion of multiplicity.

Bergson proposes a distinction between two different types of multiplicity: quantitative and qualitative. He writes:

> There are two kinds of multiplicity: that of material objects, to which the conception of number is immediately applicable; and the multiplicity of states of consciousness, which cannot be regarded as numerical without the help of some symbolical representation, in which a necessary element is space.[1]

The idea of a quantitative multiplicity is perfectly normal. The idea of a qualitative multiplicity, however, is fundamentally strange. This corresponds to Bergson's comparison of differences in degree to differences in kind. When one looks at a

[1] Bergson, "The Idea of Duration," 65.

quantitative multiplicity, one only sees differences in degree. However, upon examining a qualitative multiplicity, one discovers differences in kind. It is much easier to compare things in which there is only a difference in degree.

This notion of difference is relatively intuitive. For example, the difference between a small apple and a large apple is a difference of degree in size, while the difference between an apple and an orange is a difference in kind. It becomes more complicated when scale is involved, however. If we take both fruits to be part of the category of round things, then the difference between an apple and an orange could very well be a difference in degree of roundness. It is thus important to specify scale and category when discussing difference.

The reason for identifying the two types of differences is that Bergson describes the virtual as a qualitative multiplicity. To fully understand the virtual, however we must first distinguish between the virtual, the possible, the actual, and the real.

The notion of the virtual is one of Deleuze's key concepts, but it originates in Bergson's work. This distinction has been elucidated many times by various authors,[2] but will be examined here due to its centrality to the notion of derivative reason.

Bergson problematizes the classic distinction between the possible and the real, instead favoring a distinction between the virtual and the actual. It is important to note that for Bergson and Deleuze, the virtual and the actual are both equally real. The problem, according to Bergson, is that the notion of the possible strips the world of its creativity.

For Bergson, the idea that the possible is less than the real follows along the same lines as the illusion that disorder is less than order, or that nothingness is less than being. For included in the concept of nothingness is both being and its negation, as disorder is composed of order and its negation. In Bergson's words, "the possible is

---

[2] See Elizabeth Grosz's *The Nick of Time* for a recent example.

only the real with the addition of an act of mind which throws its image back into the past, once it has been enacted."[3] Bergson believes that the possible is not possible until it happens. In other words, "it is the real which makes itself possible, and not the possible which becomes real."[4] According to Bergson, humans project the possible back from the real into the past. Bergson's argument here is that, for something like *Hamlet* to be "possible", it would have to already be created in a sense, as to imagine its existence is to create it at some level. The virtual does not resemble the actual, while the possible does resemble the real.

A traditional idea of algorithms would see them as acting only in the realm of the possible, realizing logical and computational states. This is indeed a useful model; many problems in computer science are conceptualized as probabilistic transitions from one state to another. Bergson does not take issue with the manner in which reason traditionally acts (in states and discrete transitions), but rather with the idea that this is an accurate reflection of reality. Drawing on Luciana Parisi's work and the theory of derivative rationality, however, we can see that algorithms do more than realize possibilities—they generate actualities. Since Bergson sees the virtual as a method of the genesis of novelty in the universe, this enables the creation of new things through derivative rationality.

The notion of the virtual is important for the theory of derivative rationality because it enables an exploration of the link between the actual behavior that derivative algorithms are translating and the digital realm which they are translating behavior into. I propose that this digital realm lies somewhere between the virtual and the actual, and that this is what enables derivative rationality to act as an avenue into the novel.

---

[3] Bergson, "The Possible and the Real," 229.
[4] Bergson, "The Possible and the Real," 232.

Derivatives

Now we can return to the notion of derivative rationality that was introduced in the introduction and examine it more thoroughly. It may be helpful here to present a brief history of the derivative. In mathematics, derivatives are the study of how things change, an abstraction of the idea of change from a series of data points. In a way, mathematical derivatives act as a method of quantifying difference or change, assigning a number to a continuously varying function at a point. By looking at a graph of a function, one can see that its value is changing at a certain point, but by taking the derivative of the function at that point, one can discover exactly how much the function is changing. The original financial derivatives were also based on this principle, in that they allowed groups of people to mitigate the risk involved in fluctuating value. There are many types of financial derivatives, but most involve quantifying change in some manner. This allows traders to abstract and extract value from some underlying assets that cannot be valued in the traditional sense.

There is, then, a dual nature to the derivative. It can be an abstraction, creating a new value from an underlying value, or a reduction, reducing something complex to something less complex. Both operations extract value, but one adds a layer on top, like financial derivatives, while the other trims a layer off, like mathematical derivatives. The derivative rationality that I am concerned with does both. Derivative algorithms reduce complex human behavior to digitality, and something is indeed lost here. But something is gained when the derived is reconstituted in the realm of the digital.

So what does derivative rationality look like in practice? Recall that in the last section I named three different algorithms that exhibit derivative rationality: high-frequency trading (HFT) algorithms, Amazon's recommendation algorithm, and Facebook's advertising algorithm. Each of these algorithms receives information about human behavior as input and generates something that affects human behavior as its output. In doing so, each algorithm must form its own unique method of representing human behavior. Amazon's revolutionary idea was to focus on similarities between items instead of between users, and in its recommendation algorithm the behavior of people is represented by their purchases, that is, through the lens of products. This is an apt example of derivative rationality: at one pole, it reduces novel human behavior and uses it as means to the end of directing behavior (to buy products). However, at the other pole, it creates a novel realm of inhuman behavior, a digital world where digital agents interact and produce recommendations. It is not exactly clear with this simple example, however, where the potential for digital novelty comes in.

The second example, and maybe the one most talked about today, is Facebook's amalgamation of algorithms designed to provide efficiency in digital advertising. This example is placed second because although Facebook's algorithms do not directly act in the material world, they have arguably more influence than Amazon's, at least in its simplest form, in that Facebook also uses the beliefs about users it derives to influence what information they are exposed to. Here again, an algorithm quantifies and hypothesizes about human behavior, taking into account what users view, what they like, and what pages they follow, and using that information to group users into bins. Then, the algorithms target advertisements toward the digital specters of users. However, these advertisements are frequently inaccurate, and reveal what Facebook's algorithms "think" about users. This provides a window into the digital selves that Facebook derives from each user's translated interests (a phenomenon that will be discussed in more detail shortly).

Perhaps the finest example of derivative rationality, and indeed the one that inspired some of Parisi's (and, by extension, my own) work in the first place, is high-frequency trading algorithms. Part of the reason this is such an apt example is because, in the case of HFT algorithms, the algorithms actually have the power to act in and change the human world. HFT algorithms affect the world in several ways. Not only do they change the material conditions of the world by encouraging traders to locate their large server farms closer to exchanges, they also make trades that have direct impact on the world's finances. This example of derivative algorithms is perhaps the most opaque, as the calculations and subject matter are so complex that it can be difficult for even the people who designed the algorithms to discover the cause of a "flash crash," in which security prices deeply fall and rise in a short time span. In some ways, HFT algorithms have formed a new digital exchange on top of the old, conventional exchanges of the past. This digital exchange is unpredictable, volatile, and arises from a network of digital agents acting on translated information about human behavior—in other words, derivative rationality in one of its most obvious forms.

While these examples elucidate the nature of derivative rationality, it can be difficult to see what purpose derivative rationality serves for humans. Indeed, by name alone, the idea that derivative rationality can create something novel seems counterintuitive. Upon closer examination, however, it's hard to see why the derivative is not novel. What follows is an attempt to discover this novelty at which lies at the heart of derivative rationality.

The Derivative and the New

It must first be acknowledged that financial derivatives and derivative algorithms are mostly used to serve instrumental reason, that is, as means to the end of profit. However, I wish to explore in this section the potential of derivative reason to create the new; using derivative algorithms to serve a different purpose. This occurs when derivative algorithms reveal their inner chaotic, digital nature to their users.

The reasons for discovering and creating new and novel things were covered in the introduction, but a brief reminder will be provided here. Much has been written about how societies of control are using computers and advanced computational techniques to predict and direct human behavior. Even if one does not find the prospect of control disturbing in itself, the idea that less and less in the world is not predicted has ramifications for art, creativity, and quality of life. The question then becomes, if computers are assisting in the encroaching of prediction and predestination in our everyday lives, what remains as a source of unpredictable novelty?

Encounters with derivative rationality can feel similar to the phenomenon of the uncanny valley, in which computer-generated imagery sits in an unsettling line between artificial and human.

A useful example of this is Google's DeepDream, which takes the abstract qualities identified by a neural network in an image and projects them onto other images. Of course, the DeepDream technique was developed intentionally in order to demonstrate the way neural networks work, but the technique still allows us a glimpse into the derivative digital realm. The results of DeepDream tread a thin line between disturbing and entertaining, but they are novel regardless. Another example of derivative rationality becoming distorted is the aforementioned Facebook advertisement mistakes. When one gets a targeted ad that has little in common with one's interests, it is an uncanny encounter with a digital, derived self that is similar to one's actual self, but different in ways that one can only glimpse occasionally.

Derivative algorithms frequently give rise to unexpected things, like one sees with HFT algorithms. These algorithms will come to wildly different conclusions than humans exposed to the same data, and these conclusions are fundamentally unpredictable, since the processing of some data frequently alters the algorithms themselves. The results of neural networks are not pre-determined, since the system changes when one interacts with it.

Following these examples, I propose that we should push derivative rationality to its limits in search of the new. In some ways, this is glitch art taken to an extreme. Instead of just taking advantage of the way that computers compress and format information, taking advantage of derivative rationality means exploiting the way algorithms derive quantities from qualities and translate the real into the digital. This, then, is the purpose of this project: to use uncanny encounters with the digital realm of derivative rationality in the search for the new.

V. Conclusion: Gained in Translation

Now that I have proposed a full theory of derivative rationality, I would like to reflect on the implications of this theory and ways that it could be expanded upon.

There are, as with any theory, several potential critiques of my theory of computational reason. While I cannot fully do justice to these here, I will briefly mention the most pressing.

Perhaps the most basic objection to this position is that computers cannot think. It is certainly true that, at the moment, computers cannot think like humans do. However, if we take a “black-box” or behaviorist approach to intelligence, it seems to me that they think to some extent, particularly with neural networks where we can’t understand their “thought process” any more than we can a squirrel’s, even though we technically programmed them.

It is also not difficult to imagine this thesis being critiqued as being too in favor of computers by someone such as Golumbia. However, I’m not saying that computers are the solution to everything, nor that they are inherently a positive force in the world. I am only saying that we need to examine computation philosophically at a deeper level, and this is my attempt to do so.

Partly to address these criticisms, in the future I would like to expand more on this notion of derivative rationality, push it to its limits, and see what it can and cannot do. I would also like to spend more time exploring the political reasons and ramifications of this project, and to discuss the project in relation to a wider variety of theorists. For now, however, I am satisfied with the conclusion that there is a latent potential for novelty within complex computation.

The liberatory potential of derivative rationality comes when it is turned in on itself. In these brief moments of glitched digital vulnerability, we catch a glimpse of the digital realm as the unpredictable and novel slips out into the actual world. There is undeniably something lost when a thing is translated from the actual into the digital. However, the point of this thesis is that there is also something gained in this translation. For too long, theorists have believed that computation only exists to serve instrumental reason. Upon examination of derivative rationality, though, it becomes clear that there is indeed a creative, unpredictable force present in complex forms of computation.

References

Arvidsson, Adam. "Facebook and Finance: On the Social Logic of the Derivative." *Theory, Culture & Society* 33, no. 6 (2016): 3–23. doi:10.1177/0263276416658104.

Bergson, Henri. "The Idea of Duration." Essay. In *Key Writings*, edited by Keith Ansell-Pearson and John Ó Maoilearca, 59–94. London: Bloomsbury, 2014.

Bergson, Henri. "The Possible and the Real." Essay. In *Key Writings*, edited by Keith Ansell-Pearson and John Ó Maoilearca, 271–283. London: Bloomsbury, 2014.

Chaitin, Gregory J. "The Limits of Reason." *Scientific American* 294, no. 3 (March 2006): 74–81. doi:10.1038/scientificamerican0306-74.

Chaitin, Gregory J. *Meta Math!: The Quest for Omega*. New York, NY: Pantheon Books, 2005.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. 3rd ed. Cambridge, MA: The MIT Press, 2009.

De Landa, Manuel. *A Thousand Years of Nonlinear History*. Cambridge, MA: The MIT Press, 2014.

Deleuze, Gilles. *Bergsonism*. Translated by Hugh Tomlinson and Barbara Habberjam. New York, NY: Zone Books, 2011.

Deleuze, Gilles. "Postscript on the Societies of Control." *October* 59 (1992): 3–7. http://www.jstor.org/stable/778828.

Deleuze, Gilles, and Félix Guattari. *A Thousand Plateaus: Capitalism and Schizophrenia*. Translated by Brian Massumi. Minneapolis, MN: University of Minnesota Press, 2007.

Galloway, Alexander R. *Laruelle: Against the Digital*. Minneapolis, MN: University of Minnesota Press, 2014.

Galloway, Alexander R. *Protocol: How Control Exists after Decentralization*. Cambridge, MA: The MIT Press, 2004.

Galloway, Alexander R., and Eugene Thacker. *The Exploit: A Theory of Networks*. Minneapolis, MN: University of Minnesota Press, 2007.

Golumbia, David. *The Cultural Logic of Computation*. Cambridge, MA: Harvard University Press, 2009.

Hardt, Michael, and Antonio Negri. *Empire*. Cambridge, MA: Harvard University Press, 2011.

Negroponte, Nicholas. *Being Digital*. London, UK: Vintage Books, 1995.

Parisi, Luciana. *Contagious Architecture: Computation, Aesthetics, and Space*. Cambridge, MA: The MIT Press, 2013.

Parisi, Luciana. "Instrumental Reason, Algorithmic Capitalism, and the Incomputable." Essay. In *Alleys of Your Mind: Augmented Intelligence and Its Traumas*, edited by Matteo Pasquinelli, 125–37. Lüneburg: meson press, 2015.

Parisi, Luciana, and Stamatia Portanova. "Soft Thought (In Architecture and

Choreography).” *Computational Culture*, no. 4 (November 2011). http://computationalculture.net/soft-thought/.

Shklovsky, Viktor. “Art as Device.” Essay. In *Theory of Prose*, edited by Benjamin Sher, 1–14. Dalkey Archive Press, 1990.

Weber, Max. *Economy and Society: An Outline of Interpretive Sociology*. Edited by Guenther Roth and Claus Wittich. Berkeley, CA: University of California Press, 1978.

Whitehead, Alfred North. *Science and the Modern World*. New York, NY: The New American Library, 1948.

Worstall, Tim. “Don't Worry, Be Happy - High Frequency Trading Is Over, Dead, It's Done.” Forbes. Forbes Magazine, March 25, 2017. https://www.forbes.com/sites/timworstall/2017/03/25/dont-worry-be-happy-high-frequency-trading-is-over-dead-its-done/.