Deep Aesthetics

Computational Experience in a Time of Machine Learning

ANNA MUNSTER

DeepAesthetics

Thought in the Act

A series edited by Erin Manning and Brian Massumi

DeepAesthetics

Computational Experience in a Time of Machine Learning

ANNA MUNSTER

DUKE UNIVERSITY PRESS
Durham and London
2025

Printed in the United States of America on acid-free paper ∞
Project Editor: Bird Williams
Designed by A. Mattson Gallagher
Typeset in Minion Pro, Source Sans 3, and Adobe Arabic
by Copperline Book Services.

Library of Congress Cataloging-in-Publication Data
Names: Munster, Anna, author.
Title: DeepAesthetics : computational experience in a time of
machine learning / Anna Munster.
Other titles: Deep Aesthetics : computational experience in a time
of machine learning | Thought in the act.
Description: Durham : Duke University Press, 2025. | Series:
Thought in the act | Includes bibliographical references and index.
Identifiers: LCCN 2024039427 (print)
LCCN 2024039428 (ebook)
ISBN 9781478031543 (paperback)
ISBN 9781478028338 (hardcover)
ISBN 9781478060529 (ebook)
Subjects: LCSH: Deep learning (Machine learning) | Machine
learning—Social aspects. | Artificial intelligence—Social aspects. |
Technology—Social aspects. | Knowledge, Sociology of.
Classification: LCC Q325.73 . M86 2025 (print)
LCC Q325.73 (ebook)
DDC 006.3/1—dc23/eng/20241226
LC record available at https://lccn.loc.gov/2024039427
LC ebook record available at https://lccn.loc.gov/2024039428

Cover art: Anna Ridler, *Mosaic Virus*, 2019. Still from 3-screen
GAN video installation. Image courtesy of the artist.

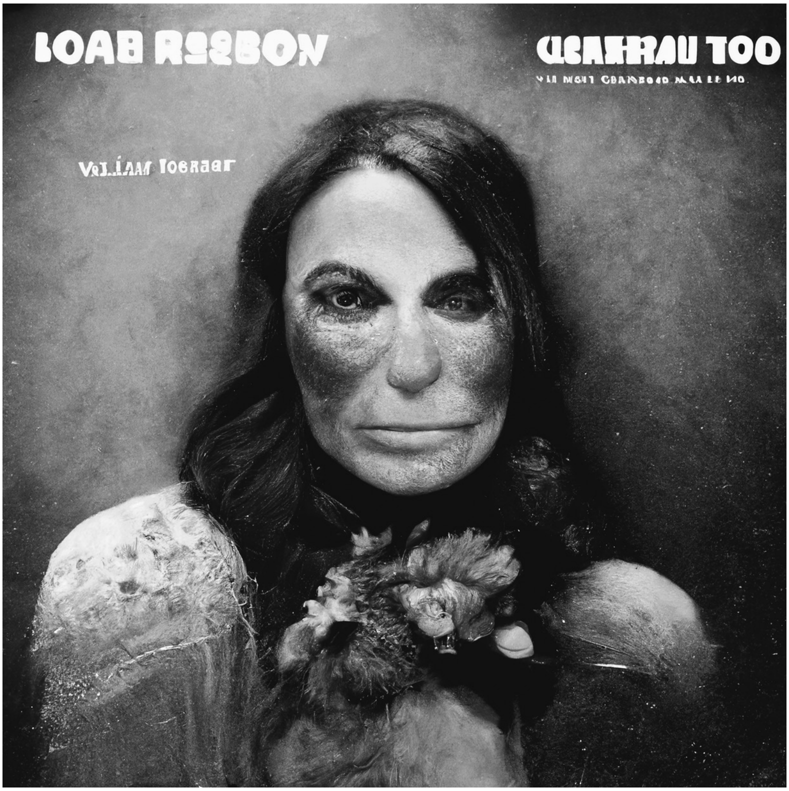
For all my collaborators, human and more than.

CONTENTS

Introduction: Deep Machines and Surfaces of Experience

An Experience of Computation

Cryptozoology may seem as far-flung from data science as paleontology or alchemy. But in April 2022, a Swedish musician and artist working on a then obscure generative art strategy burst an artificially intelligent "cryptid," or mythically existent creature, onto the text-to-image artificial intelligence (AI) creative scene. Cryptids are animals that populate folklore, subcultures, parapsychology, and, increasingly, the internet. They exist *in the wild*, in wild places; they are creatures mainstream science refuses to verify. After several months of experimenting, Steph Maj Swanson introduced her proliferating images of something she called "an emergent phenomenon that arises in certain AI image synthesis models" (Swanson 2022a), via her "Supercomposite" Twitter account. She named the woman who



I.1 *untitled progenitor Loab 2*, one of the first two images of Loab posted by Steph Maj Swanson, using the artist name Supercomposite, on her Twitter account, September 7, 2022. Image courtesy of Steph Maj Swanson.

regenerated, seeded, and spread across these images "Loab." Within two days, however, Loab was proclaimed the first "AI-cryptid" by eBaum's World (Zachnading 2022), an online meme-gathering and entertainment site; and within a week, the powerhouse British fashion and culture magazine *Dazed* had likewise confirmed her cryptid arrival (Waite 2022).

Not only had Loab become an emergent AI phenomenon; she had also emphatically implicated the unverifiable creatures of cryptozoology in (data) science's latest shiny enterprise of large-scale language models with generative capabilities.

Driven by the machine learning (ML) capacities and infrastructure of such models, AI had already been heading in one or several of such trajectories for some time: deepfakes standing in for the media presence of actors and presidents; AI influencers on Instagram and TikTok with millions of followers; lifelike computationally synthesized portraits of nonexistent human faces. But the efforts of artists, developers, and data researchers in crafting text "prompts" and finding new ways to conjure and intervene in the ubiquitous culture of statistical computation garner less notoriety and commentary. Loab was not hiding in the "black box," however, waiting to crawl out from under a wild AI place; she was prompted. While Swanson notes the emergent properties of AI, she nonetheless introduced the Loab image series and genealogy via a lengthy Twitter thread on process and AI text-to-image generation.[1] Swanson had begun with an experiment in scripting an image prompt—the action of entering text into a field in an interface to generate an image by an AI/model. Models offering these interfaces, such as Stable Diffusion, Midjourney, and DALL-E, became hugely popular throughout 2022 and were accessed via web browser and dedicated apps or run as stand-alone models on desktop computers. They are part of a suite of deep learning models that engage, augment, and extend the computational architecture of large language models (LLMs), which I begin to unpack in chapter 1 and take up in more detail in chapter 3. As with much AI, technical developments in model architecture, training, and operativity are both incremental and swift. Throughout this book, I offer examples of algorithms, models, and practices that may seem specific to a particular historical and technical development or may appear no longer to be widely in use. Yet my selection of such examples often rests on their ongoing if unremarked deployment by machine learning or, in the case of Loab, because the operation performed on or with them becomes a way of probing the specificity of machine learning computational experience.

Swanson's experiment involved doing something slightly different with text-to-image models: "negative prompting," or writing a script for the opposite of what the prompt will sample from the image space. In negative prompting, the model samples images the furthest statistical distance away from their match to the text written in the prompt itself, that is, furthest along a distribution of text and image-matched data on which the model has been trained. In the case of the prompt that began the generation of Loab, Swanson tried for a statistical negative of the text "Marlon Brando," entering the script "Brando:−1." This returned an oddly banal imagemash of a nonexistent company logo, "DIGITA PNTICS," set against a seemingly hand-drawn schematic of a skyline, with text deformations typical of the model's inability to graphically render text. Swanson became curious about a double negative prompt that might then send the model into re-generating Marlon Brando's image when prompted by the negative of its negative (prompt): "DIGITA PNTICS skyline logo:: −1." Instead Loab appeared, affirming her imagistic rise as the model's response to the negative of Marlon Brando–esqueness.

Although Loab, the AI-generated woman, has been called a cryptid, a "demon" (Ryan 2022), and a queer icon (Levanier 2022), Loab, the AI images generated by text-to-image ML, can be understood a little differently as what I will call "process probes." These are techniques that Swanson developed to sift and drift through the latent spaces of AI's generative imaging. As I explain in chapter 1, latent space is always specific to an AI's or model's distribution of the data points it reorganizes when training on an original dataset. As a model trains, it organizes and distributes its input data or, in some generative image models, the randomized noise data it is fed via a particular algorithm or function. These input data are clustered into groups or features by the model's operations. As data are reorganized, a distribution takes shape that becomes part of the learning the model gains about the data, a distribution in which the features are contoured by relations of close or distant resemblance and proximity. When Swanson artfully scripted her prompts—with Marlon Brando–esque, then negative Brando-esque, then the negative negative of Brando-esqueness—she prompted the model to sample across its trained distribution of data proximities and distances (or resemblances and dissimilarities). It is this overall configuration that contours text and image-paired data into what is called the latent space of a generative text-to-image model. Swanson developed and stumbled on Loabness out of Loab's "latent" potential phenomenality through her artful and curious probing of the blind, relational, and potentially wild spaces of machine learning-driven AI. Swanson both stumbled across and developed a *co-loab-oration* with the model, finding something odd, lurking, but barely there as "an emergent island in the latent space that we don't know how to locate with text queries" (Swanson 2022b). Loab was the output of a process that artfully explored the unknown unknowns of what is otherwise touted as predictive computational experience.

Experiencing DeepAesthetics

More than an artwork, more than a collaboration between AI and human, Loab affords us a particular mode of *experiencing* computation. If indeed she-they/Loab-Swanson probes the processes and relations that make up the physically nonexistent yet real statistical terrain of latent space, then she and they occupy and help generate a radically different idea of and encounter with computational experience. This has nothing to do with the identity of the woman or artist behind the Loab image. This has to do with conceiving experience differently or rather *differentially*. I take up William James's philosophy of experience in the context of probing ML and its strange emergent Loab-like phenomena throughout this book because James's concept of experience can address machine *learning* as processes that change, and as processes for experiencing computation changing. For James, and process thinking more broadly, change unfolds; *it becomes*, via processes of continuity or conjunction, and discontinuity or disjunction. This offers us experience based not on identities or positions such as the human, computer, or AI but rather on undulating fields shot through with continuously changing relations. James furnishes a conception of experience philosophically placed to one side of the "lived experience" from the phenomenological tradition and, different again, from explorations through feminist, race, and queer politics and theory, still preoccupied with all too *human* embodiments and trailing under the long tail of identity politics. Rather than experience being "purified" and reified to some primary "ground truth," body, or position, James's "pure experience" welcomes all, any, and every experience. Crucially, pure experience is not made by or filled with things or places such as "subject" and "object" but is generated through relations and processes, which James terms "co-ordinate phenomena" organizing its space-times (1977, 199). So too do relations organize the space-time of *computational* experience and its weird yet powerfully generative topologies. And while Loab has been described as a ghostly haunting of AI and a "creepypasta" or internet horror figure (Ryan 2022), the Loab imagescape registers something real: the statistical reconfiguration of experience by ML computing. As computation has increasingly been inflected by ML, our cultural, computational, and medial outputs as online images, generative artworks, and text corpuses—indeed, all and any data—have been modeled into maximum and minimum clusters of proximities that simultaneously butt up against one another as continuous regions or disperse away from one another via discontinuous edges and outliers.

Now that ML is so pervasive a form of enacting computational processes, contemporary experience has become littered with all manner of imperceptible statistical relations. Many of these take place between humans and computers, and many others among computers or computational elements alone. Collectively and differentially, these multiply scaled, differentiating relations change the stuff of experience, change all those living and technical elements experiencing, and generate new relations that unfold into many ways for making futures. As the hegemonic form of computing today, ML encompasses a diverse terrain of systems theory, practices, and applications that build and modulate computational models in relation to inputs or data (Alpaydin 2016, 17).[2] The capacity of the computational model or AI to change in relation to (changing) inputs is what is understood by data science as "learning." These modulatory and adaptive systems underpin everyday (human) experiences such as online shopping, streaming music, and airport security by using, respectively, recommenders, collaborative filtering, and biometric recognition. And they are now organizing and contouring swathes of individual and collective engagements and encounters. Loab skips across many ordinary and extraordinary aspects of how ML has reconfigured computational experience—from the ways in which entering a text prompt can now generate slabs of generic text reportage and writing being used by students to write essays, to the ways in which text can artfully be rescripted to create uncanny imagery. Loab, then, sensibly registers the processes that together underpin and generate ML experience.

In a definition of ML often quoted by data scientists, *experience* is a key term in judging whether a computational system qualifies as one that learns: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience 'E'" (Mitchell 1997, 2). Here experience, or E, appears twice in the proposition, but importantly, its recurrence suggests change. E is first a defined phenomenon—it could be a measurable state of a set of inputs, for example—on which a series of algorithms performing tasks (T) have run (P). But it can also become the change taking place—the "improving," for example—from which a measure of the tasks' improved performance is taken. Tellingly, that measure of improvement is ascertained by running an ML program or model many times over the data while it is in its primary learning phase or training. Here, E varies as the model's learning attempts to recognize a structure or pattern in the data. Experience, then, for ML, is simultaneously quantifiable as a state of measurable change *and* the ongoing process of learning that variably qualifies what that change is to be over time. The final improvement, or what the model has learned, is therefore really a coalescence of many processes of modulation differing from and conjoining with one another. As changes or learning occurs, this modulation—which we will come to know as the model's operativity—*qualifies* the entire ML ensemble of data and functions. Machine learning experience is an ellipsis of the two experiences—of what has occurred and what is occurrent. However, ML is typically researched, reported on, and implemented as a "learning problem" to be solved (Mitchell 1997, 3). Rather than the dynamic ellipsis of past and present, E is more often than not reduced to a quantifiable ratio of improvement measure against a set of inputs on which a model trains, such as a database of faces for training a facial recognition AI. The ongoing learning then becomes E as the "measure of improvement" against the E of the trained model. We lose the dynamic set up by the vectors of the two Es continually traversing the occurred and occurrent, and traversing the temporalities of a backward and forward. We also lose what this ellipsis of present and past, past and future, suggests about a double process performed by ML. This doubling involves both a reaching across and a contraction of quantity with quality. Such slippages, extensions, and mergers from quantity to quality and back are at the relational heart of the computational experience of ML.

To take up these pulsations rhythmically, I want to propose that deepaesthetics, a concept I use to think computational experience in this book, is likewise occurring via different rhythms of expansion and contraction. Deepaesthetics offers us both conjunction and disjunction, gluing together two worlds that do not seem to be of concern to each other: deep learning, the subfield of machine learning that uses neural network architectures; and a branch of philosophy traditionally concerned with how valuations of formal or sensory qualities come to be made. I am interested in how that contraction actively sticks together through the operations performed by ML computational assemblages, through the creativity attributed to ML-driven AI, and via actual artworks that stage encounters with ML. I am also interested in how it splits apart, rupturing as the forces across computational and human experience difference each other.

Aesthetics is, of course, much more than a branch of philosophy; even philosophically in the Western canon, its history is complex and ambiguous. Since the eighteenth century, philosophical debate has oscillated between the different positions taken up by, on the one hand, Immanuel Kant's categorization of aesthetic judgment in his 1770 *Critique of Judgment*, which ultimately grounded aesthetic value in a disinterested appraisal of perceived, sensory phenomena (1987, 44), and, on the other, Alexander Baumgarten's *Aesthetica* of 1758, in which he considered all sensory experience to be aesthetic. The contemporary fallout of this legacy for an aesthetic consideration of computation has been, largely, to fall on either side of a formalist or sensorial approach, although in the ensuing aesthetic debates, neither formalism nor sensorialism maps back neatly onto Kant and Baumgarten. Aspects of formalism characterize the work of Beatrice Fazi's (2018a) computational aesthetics in her argument that the digital must be taken as a formally autonomous realm, whereas, for example, Mark Hansen's approach has been to continue to understand computation as a "phenomenotechnics," a kind of entanglement of the technical with lived, sensorial experience (2021).

However, computational experience is entangled with modes of sensing that are not only beyond human sensing but, as Matthew Fuller and Eyal Weizman argue, beyond *perception*: leaves that become sensitive to herbicides and whose sensitivity might be measured via biosensors, sensors encoded to detect respiration and moisture rates in greenhouses, sensing undertaken by computer vision models that detect variability in the forest canopy (2021, 33–50). This makes both a formalist and an embodied sensorialist aesthetics tricky. Fuller and Weizman argue that sensing—whether computationally enabled, augmented, or occurring outside of computation—involves events in which all kinds of surfaces register and inscribe their contact with one another, events that can be ordinary and everyday as well as technically refined and deliberate. They call this panoply of sensing events coursing throughout the world "aesthetic," which entails the aesthetic as just that ubiquitous domain of all sensing relations. Their argument concerning aesthetics as the potential for the registration and inscription of sensing on any and by every surface whatsoever resonates with my Jamesian approach to (computational) experience throughout this book. As I explain a little later in this introduction, experience understood via process philosophy comprises largely any and every relation in its/their process and quality of relating and registering these relations.

But, as Fuller and Weizman explain, registration and inscription events *make sense* in different ways, since even ubiquitous relations do not register evenly or with the same qualities for all entities or surfaces in relation. There is always a making sense accompanying sensing—what they refer to as "sense-making"—that involves varying "cultures of sensing" (52). Cultures should not be understood as only comprising human subjects who make sense of objects. Rather, cultures of sensing work via layers and accumulations of sensing that accrete materially, institutionally, and perspectivally under situated and differing histories and assemblages. And these are invoked to *make sense* of sensing.[3] The strata of such formations are never frictionless but involve tensions of scale, perspective, materiality, and power. Aesthetics, for Fuller and Weizman, is this bringing into relation of sensing events with cultures or formations of making sense and can itself involve tensions.

In this book, I take Jamesian “experience” as the broader term with which to begin, since it is always already about the reality of relations in which surfaces and strata of computational, sentient, or any matter are eventfully in contact, registering and prehending each other. But, like Fuller and Weizman’s notion of aesthetics, this does not mean that experience is self-similar in its relationality or registration. In this book, I understand aesthetics as modes of individuating this broader field(s) of experience, so that experience comes to make sense via singular sensibilities, whose accretions and formations may well be riven with tensions. Machine learning engenders a specific individuation of computational experience in which we are both asked to encounter and bound to insensible and microperceptible forms of nonlinear and continuously modulating statistical function and calculation; this is its aesthetic condition. This poses a problem for sensing and sensibility: How can we perceptually register and even account for what occurs computationally at scales, durations, and dimensions that are nonhuman and, frequently, imperceptible?

In the now famous deep learning research that accompanied Google’s Inception model (see Szegedy et al. 2015), hallucinatory synthetic images of dogs, birds, bananas, and more seemed to have emerged via an imperceptible process from a starting point of visual noise. To make this process explainable, Google’s researchers developed an entire visual online site stepping through the movements from noise to recognizable animal/object in an image.[4] However, an entire aesthetic individuation is mobilized around this explanation of the functioning of Inception. This individuation relies on a representationalist paradigm of (visual) perception widely deployed throughout deep learning models, which I draw attention to in chapter 1. Here the desired representation of an object seems to emerge via continuous steps out of an initial flux: “Start with an image full of random noise, then gradually tweak the image towards what the neural net considers a banana” (Mordvintsev, Olah, and Tyka 2015). Indeed, the visual layout accompanying explanations of how features work in neural networks often reinforces this steady building up of a representation. However, imperceptible ML processes are operating and registering at the same time, cutting into the continuity of this aesthetic of representationalism; as the Google researchers admit, “By itself, that doesn’t work very well, but it does if we impose a prior constraint that the image should have similar statistics to natural images.” What cannot be visually represented in the stepping through of features, then, is just that statistical “prior constraint” learned and transduced into a statistical weighting from a different distribution and dataset of "natural images." Yet both constraints and distributions are key operations of deep learning models and are crucial to how their sensibility registers.

My use of a term such as *deepaesthetics* aims to work productively with this problem of how to sense or register what is occurring imperceptibly, whether that involves operations, statistical techniques, or the qualities of relations in ML's computational experience. Across the many sites of ML practices, techniques, and operations in this book, I focus on just those *continuities and discontinuities* that characterize the registration of ML and its processes, both by its own surfaces and by surfaces of human sensing. My wager will be that by bringing careful, granular attention to ML's processuality—that domain of computational experience registering yet often beyond perception—we can scope out what is singular about the sensibility engendered by its aesthetic individuation(s).

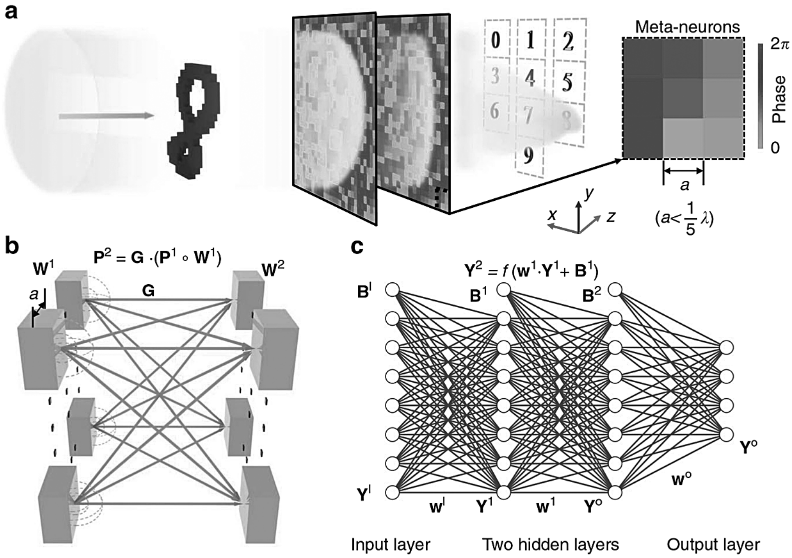
The Depth of Deep Learning

Machine learning architectures come in many forms, but a frequently used one is the deep learning neural network, in which complex computational representations of empirical phenomena are developed through "layers" of numerical values that then represent low-level up to higher-level features capable of synthesizing new data representations. In what has become a standard text on deep learning for ML research, Ian Goodfellow, Yoshua Bengio, and Aaron Courville define the "deep" of deep learning as the method and architecture for computationally resolving the problem of representation: "Deep learning enables the computer to build complex concepts out of simpler concepts" (2016, 5). From low-level features, increasing functions and processes add, subtract, and multiply in linear and nonlinear ways to increase the complexity of the representations or syntheses. Eventually, after *x* epochs of training, or synthesis, this results in new data representations/outputs that correlate with the inputs or are sufficiently synthesized to efficiently accomplish a given task. Examples might include a neural network learning the representation of any handwritten letters in an alphabet it observes after training on a database of diverse-enough alphabetic letters, or a network being capable of synthesizing realistic-looking photographs of cats in a range of positions after training on many still video frames of cats in indoor and outdoor settings. Here the deep aspect of the neural network is measured by the number of layers of parameters (constrained sets of numerical relations) through which data inputs—handwritten letters, images, text, music, and so on, transduced into numerical values—pass. To complex representations that are perceived by human sensing, "a deep learning system can represent the concept of an image of a person by combining simpler concepts, such as corners and contours, which are in turn defined in terms of edges" (Goodfellow, Bengio, and Courville 2016, 5). Part of this project for probing and registering a deepaesthetics involves finding these kinds of *discontinuities* that are lodged in the operativity and rhetoric of deep learning and pose as smooth continuities across computational and human logics and modes of perception.

Layers also mark a kind of gateway or threshold for these sets of values to pass to or from the next layer/set and then across the neural network. Throughout the book, I explore some of the commonly used technical terms that interweave ML research and practice via pop-up definitional boxes and accompanying images and diagrams. These will facilitate our encounters with the technics of ML, giving us provisional means to navigate its techniques and operations. The purpose of these is not to comprehensively define all technical terms throughout the book. Instead, I touch on technical elements that recur and also often work ambiguously or opaquely within data science. These pop-ups are, then, terms to watch out for, terms that return, and terms that trouble the field.[5]

Depth in neural network architectures also refers to the multiplication of layers, which occurs in larger and more complex models. Learning is then understood to occur as the network successively "discovers" across these many layers of features and patterns detected in or generated by inputs (LeCun, Bengio, and Hinton 2015, 436). These layers are often referred to as "hidden" in the sense that both their location and knowledge of their exact functionality in the model may not be precisely discernible. The deeper the network or the more layers it has, the better able—the claim is made—it can train for fine-grained features. Once trained, the model will have learned how to accurately detect and predict the features of unknown data inputs. In some generative networks, deep models can create new synthetic instances of data; we are familiar with these through deepfakes and with images and videos rendered in response to text prompts by post-GPT-3 (Generalized Processing Training) large-scale language models.

Deep learning networks can deploy millions of parameters at successive layers that adjust during a model's training. Here I am describing only some of the general characteristics of deep neural networks—inputs, layers, parameters, learning, output, and prediction—to grasp the ways in which data science conceives depth as a horizontal stacking of connected layers, which



I.2 The three schematics represent the layers of the neural network architecture: first, projected two-dimensional geometric surfaces in *a*; then as a snapshot of dynamic calculative processes producing a network topology across the orange lines in *b*; and finally named as layers in which groups of values are being calculated and passed forward from input to output at *c*. From "Meta-Neural-Network for Real-Time and Passive Deep-Learning-Based Object Recognition" (Weng et al. 2020).

are perceptually inaccessible for human registration. When we look at common figures that schematize deep neural networks (fig. I.2), we immediately see that depth is imaged in avolumetric terms, if we understand volume to be a Euclidean geometrical arrangement of measurement between *x*, *y*, and *z* axes. Where layers might suggest the potential for depth to accumulate via a vertical stacking of each on top of the next, the schematics of deep learning's *operations*—its processes—perform otherwise via lateral, relaying, and recursive movements. The diagrams that describe deep learning's models must be understood, then, as inhabiting a different kind of space: a topological one of relation and process._

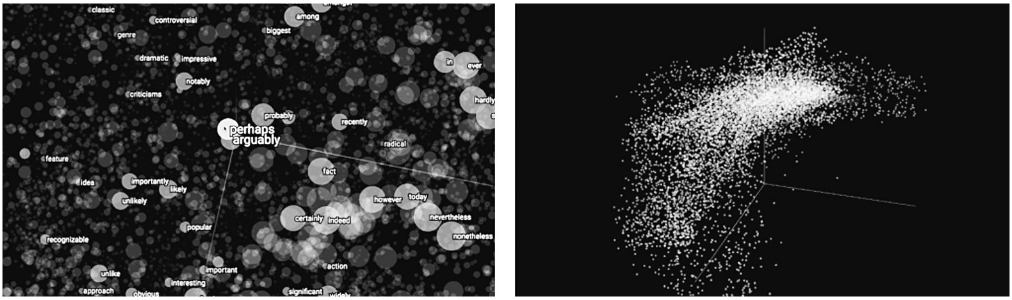
For data science, the "deep" in deep learning is also a quantitative problem, which arises out of these more-than-volumetric spaces of data-operation topologies. Much of the data being crunched through deep models has high variability; every pixel of an image data input, for example,

***Pop-Up* Definition: Layers in Neural Networks**

Layers are surfaces composed through the topology of a neural network's numerical and functional (algorithmic) relations. They are often represented in ML diagrams of neural network architectures as geometrically projected two-dimensional surfaces. In actuality, they comprise calculations, extracted sets of values, and vectors produced by functions of a particular neural network.

The definition of ***layers*** varies in machine learning and often conflates concepts such as surface, gateway, feature extraction, and threshold. At its most basic, a layer is simply a set of data points either extracted from data inputs or computationally synthesized, which have been constrained by certain parameters (filters, weights, and so on). These constraints extract or configure certain values from the data or perform a synthesis according to a set of values, and the result of either operation is known as a "feature." These new sets of features (or new sets of numerical values) are passed on to the next layer for further extraction or synthesis.

has contrast, hue, luminosity, and saturation features, among many data points, and holds these in relation to all other data points within the same image. Together this multiplies data's relationality both intensively and extensively and is known within data science as "high dimensionality." The relationality encountered here is likewise avolumetric and can only be spatialized *n-dimensionally* (any number of dimensions), often via what we recognize as topological network diagrams (fig. I.3a, fig. I.3b). To discover across *n*-dimensional features of the data only those patterns of relevance for a specific task, ML must manage swarms of both irrelevant and less relevant features. Its management techniques involve statistical operations that compress the volume of the data's high dimensionality. This suggests that a certain qualitative flattening must be implemented for the efficient functioning of the model, which, as we will see in chapter 2, may be enacted via statistical functions. These functions or algorithms are often performed before the operations of the *deep* learning neural network architecture. Data, then, may be subjected to several other ML algorithms and processes before becoming the inputs on which an AI deep neural model learns across its layers. Such algorithms and processes seem much less spectacular than the various deep networks capturing scientific and public attention in the last decade, such as DeepMind's AlphaGo, DeepDream, DALL-E, and deepfakes.



I.3 Two different visualizations of high- or *n*-dimensional space of text data: the left panel showing global proximate and distant clustering of data points to represent similarity and difference via proximity and distance; the right panel showing detailed connectivity of words in the text to each other. Note that these visualizations are manipulable in three-dimensional computer graphic space, which also shifts the "view" onto them.

Pop-Up Definition: Dimensionality Reduction
All data have attributes or "dimensions." These might include age, sex, gender, and so on, for demographic data, and pixel color, brightness, and saturation, and so on, for digital images. Rich data such as images have many attributes for each pixel, and so their data are called "high-dimensional." When algorithms and machine learning models attempt to locate features in high-dimensional data, they may be slowed down, become inefficient, or "distracted" by unwanted attributes. Data science has historically drawn on and remodeled statistical techniques for reducing the dimensionality of data.
Dimensionality reduction algorithmically removes attributes or dimensions from a dataset that are not seen to be intrinsic to the patterns, features, or tasks being trained for, discovered, or recognized in the data. In data science, dimensionality reduction is understood to quantitatively reduce data but not to change its overall qualitative characteristics.

And yet the stacked horizontality of layers and dimensionality reduction are two key aspects of the strange topology of surface-generated depth through which machine learning–driven AI works. I will often need to navigate such functions and their sociotechnicalities alongside the deep learning models that have attracted the most attention. For this reason, I take *machine* learning, which encompasses an array of techniques, operations, and processes of statistical computation, as the larger domain of computational experience in which deep learning and generative AI are embedded.

The routinely performed processes of dimensionality reduction in ML are, however, not simply quantitative operations. They simultaneously organize the relationality of data according to vectors qualified by similarity, difference, and their interrelations. Data science attempts to measure or quantify such vectorization through functions that calculate maximum and minimum distribution of sameness or difference. Nonetheless, these functions qualify the volumes of data by shaping them into differential clusters, and this interpolates a more-than-quantitative register in/with the data. At the very moment that data come to be quantitively operated on by statistical methods such as dimensionality reduction, the data are also being respatialized and reconfigured with hidden potential for machine-discoverable pattern, recognition, and classification. Patterns or recurring motifs and classification or discrete separation would not be possible without the vectorial shaping of data, a shaping that is qualitatively immanent to quantized organizations of data. As Adrian Mackenzie puts it, when discussing the ways in which the vectorization of prostate tumor data can arrange and align features within a dataset: "The question of relation between multiple variables and . . . predicted levels . . . suggests the existence of a hidden, occluded, or internal space that cannot be seen in a data table and cannot be brought to light even in the more complex geometry of a plot. This volume contains the locus of multiple relations, a locus inhering in a higher dimensional space" (2017, 63). Even while such operations on and with data are quantitative, they also change the configuration of the data's intensive relations as the model learns: "Deep neural networks operate by transforming topology, gradually simplifying topologically entangled data in the input space until it becomes linearly separable in the output space" (Naitzat et al. 2020, 35). Depth, then, reemerges as what resides within both model and data yet cannot be seen or calculated exactly. These deep spaces emerge as data's dimensions are reduced, but they also signal a computational register that cannot be fully circumscribed by performing quantizing calculations. The "deep" in deep learning endures just beyond the measurable. This suggests a conundrum characterizing ML experience insofar as neither humans nor deep learning models seem to possess the capacities to engage those very qualities orienting and characterizing AI's operativity. Rather than quantifying, cognizing, or visualizing computational experience in a time of ML, I will propose different modes, levels, and registers for experiencing (its) experience.

With its myriad operations, novel spaces, and dynamic transductions of quantitative phenomena to qualitative events, contemporary computational experience lends itself to being thought and felt processually. Throughout the book, I will have recourse to concepts from process philosophy to understand machine-learning-based AI. Rather than focusing on linear functions seamlessly chaining inputs to outputs or to nonlinear algorithms equally striving for error-free prediction, I will focus on the recursive and modulating functioning of ML, whose processes, while calculable, are not in themselves necessarily determinable.[6] An often-heard proclamation in data science is that deep learning, in particular, is a black box: it functions, but we don't know what goes on inside (see, e.g., Castelvecchi 2016). Instead of pursuing what is determinable in the black box—which, in deep learning, has become a research domain in its own right—I will suggest that more might be gained by thinking ML experience as and through processual operations. A significant benefit of doing so is that it allows us to hold together the many tensions and knots crossing the quantitative and qualitative, the calculable and the indeterminate, the discrete and continuous, *as the very stuff of experience* in a time of ML.

A Radically Empirical Experience for and of ML

I am not the first to propose that the recursive processes of ML-driven computation lie at the core of its contemporary operativity. The updating of both data and ML systems is also commented on by Taina Bucher, who draws attention to the ways in which algorithms are in a constant process of becoming as technical, social, and ultimately governing forces and events (2018, 28). Closer to the approach I offer here is Luciana Parisi's project for affirming the incomputable as those indeterminate quantities of data produced through the recursive and nonlinear operations of computational modeling (Parisi and Dixon-Román 2020; Parisi 2013). Parisi has argued that the shift to many-layered deep learning formations of AI sets up nonlinear recursions across the model as it runs. These recursions generate a kind of extra-dimension of data from which the model itself adapts and learns but for which it can never fully calculate or compute, since its ongoing operativity maintains this very excessive generativity: “This wall of incompressible data instead overruns the program and this neutralizes or reveals the incompleteness of the axioms on which the program was based in the first place” (Parisi and Dixon-Román 2020, 57). A margin immanent to the neural architecture of AI exposes itself, asserting a gap between the model’s claims to prediction and determination and its engendering of an excess from its autopoiesis.

However, for Parisi, and likewise for Fazi (2018a), recursive and contingent computation produces indeterminacy by purely quantitative and axiomatic means. For Fazi, this occurs because computing is based on Alan Turing’s universalizing axiom—the ongoing operation of a machine’s determination to execute either state A or state B. Yet in its inexhaustible continuity, computing must necessarily encounter numbers such as infinity, which are *incomputable*: “With Turing’s incomputability we are witness to something especially surprising: it is the mechanical rule itself that, in its own operations of discretisation, generates the inexhaustibility of computational processing” (Fazi 2018a, 124). Fazi sees the simple digital process of deciding A or B, 0 or 1, a binary and determinate action, as the potential for computing to set off on a path toward indeterminacy. Crucially, for these theoretical approaches to contingency, which pursue the quantitative and axiomatic, computation exists in a separate domain from empirical experience. The former is purely calculative, whereas the empirical is apportioned to the field of the sensible. And in the empirical realm, contingency or indeterminacy only arises through material or sensory conditions and circumstances.

But what if neither experience nor the empirical are primarily sensory, that is to say, sensory in the first instance? What if this division between the formal/axiomatic and the material is a secondary division of the dynamic ongoing reality of the world as it occurs in the making, including computational worlds? As we have already seen in ML, experience can be a measured phenomenon such as an input or output on which the model runs. But it is also—and crucially for an ontogenesis of *autonomous systems*—occurrent learning, or computation as the *differencing* generated as computing, the process, happens. If we think just of an AI model as a limited instance of computation, we locate that change or difference as the experience the model gains across its network by vectorially mapping the *relations* of inputs to outputs. We cannot reduce the function of this to any causal or linear mapping of data inputs to outputs, since most neural networks function via combinations of nonlinear relations such as pooling, back propagation, reinforcement learning, and so forth. Here "nonlinear" means that the outputs and inputs cannot be directly algorithmically mapped to each other.

What we can state is that any kind of ML understood as "a computer [that] is said to learn from experience E" involves *relations* of comparison, contrast, addition, subtraction, and multiplication in which both model and data configure and reconfigure through modulation, that is, the work of ongoing change/differences. And, more broadly, ML experience, as technical—and, as I will engage it in this book, as cultural, aesthetic, and social—must be taken more broadly as change occurring via the multiplicity of computational *relations* on the move. These encompass the model's algebraic relations but also those across data and its preprocessing via operations such as dimensionality reduction; the vectorization of data by a model; the differences produced by relations between neural networks in an assemblage of models, which are often used to accomplish complex tasks such as AlphaGo's chess wins; back propagation (used to efficiently calculate the multiple derivatives produced by the model computing the many variables of data inputs); optimization (which makes the model run with a reduced error rate); and a multitude of human intelligence tasks with which all the purely computational operations might also be entangled.

In this book, I unfold an approach that emphasizes the vectorial and *qualitative* operations performed by ML. These range from statistical functions such as principal component analysis (PCA), part of ML's array of algorithms, to the complex, dispersed, layered, and recursive architectures of multilayered neural networks. These qualitative operations are always exchanges between and across the quantitative (data) and axiomatic (algorithms or functions) and qualitative operations such as recursion, vectorization, and so on. Or, rather, we could say complex relations of sameness and difference traverse computational quanta, functions, and operations. It is this operative *relationality* that accounts for ML's contingent and nonpredictable modes of computation in which novel spaces and sensibilities form. And while these spaces and potentialities are insensible, this does not foreclose their registration as a sensibility specific to ML. This is *machine learning experience*: production and registration of a peculiar computational experience. And an ML sensibility can also be artfully conjured and encountered. It is part of the project of this book to signal where and how such encounters occur in the work of artists, cultural producers, and sometimes experimental data scientists interested in an alternate deepaesthetics.[7] Indeed, experiments with ML's relationality are already occurring, exploring its potential to open to the nonknown. In an experiment with the transla-

I.4 "Steps from Generated Image to Sketch to Physical Model." Philipp Schmitt and Steffen Weiss, *The Chair Project*, 2018. Image courtesy of Philipp Schmitt.

tion of synthesized images of chairs generated by a generative adversarial network (GAN) model, Philipp Schmitt and Steffen Weiss (2018) took the odd, dysfunctional deep-learning-generated images of blurry chairs to aid the human design of physical, albeit speculative, "chairs." Images generated by GANs, with their transmogrified snapshots of their model's learning of prominent features across a training dataset, have become readily identifiable as an aesthetic visual style of ML. But rather than veer either toward aesthetic realism—where the visual objective is to get the model to synthesize a realistic-looking chair via training on a dataset—or toward the "latent-space" style associated with GANs, Schmitt and Weiss's *The Chair Project* does something different.

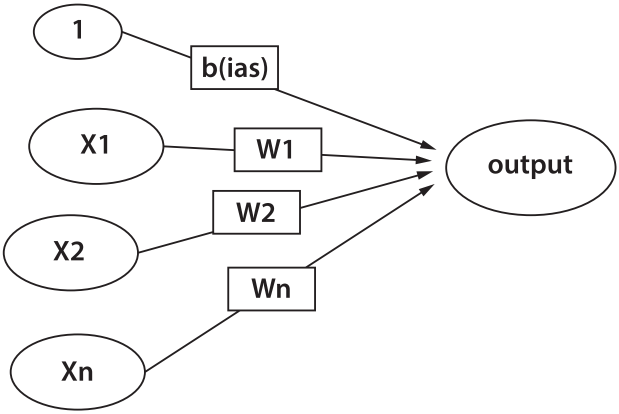
The GAN-generated images become visual prompts that probe and produce a relationality across model and (human) designer: "The idea was to neither simply trace the generated images, nor to transform it into traditional pieces of furniture. Rather, we brought out the chairs we saw in the blurry images to help viewers see what we imagined. 'Seeing the chair' in an image is an imaginative and associative process. It pushes designers away from usual threads of thinking towards unusual ideas that they might not have had otherwise" (Schmitt and Weiss 2018, 2). The resulting physically crafted chairs are emergent realizations that embody the processes of back-and-forth prompting and probing across model and designer and across human perception and computational sensing. The chairs hint at a classic modernist design lineage bound to notions of "form follows function" while also pushing at the limits of functionality through their speculative and "useless" design. The GAN itself can only assemble features; it cannot see a chair. Nonetheless, it can furnish novel conditions for seeing to occur, and these can condition what seeing might be or become. Likewise, the human designer is prompted via the speculative constraints, probes, and parameters that the model blurrily furnishes. The artfulness of *The Chair Project* lies in the ways in which human and more-than-human forces collectively individuate across each other to co-compose previously nonknown chairs: chairs that are more than what is already in a dataset, and more than what a trained designer might sketch through their own imaginings.

The artful probing of ML's sensibility enables rich and generative encounters with computational experience that often question and sometimes even upturn a more predictive teleology. But to fully account for ML as a relational mode of computing, we need to think with process thinking. James's thinking has the advantage of valuing relations as the real stuff of experience, providing a deceptively simple definition of relations as "different degrees of intimacy" (1977, 196). By intimacy, James means a proximity of connection that produces varying degrees of transition in experience: "With, near, next, like, from, towards, against, because, for, through, my—these words designate types of conjunctive relation arranged in a roughly ascending order of intimacy and inclusiveness" (James 1912, 43). His emphasis on conjunction as a key form of transitions occurring revalues relations of continuity. Continuity is always generated qualitatively via relations that modulate things as they also change their milieu: "Not only is the *situation* different when the book is on the table, but the *book itself* is different as a book from what it was when it was off the table" (James 1977, 223). Continuity, then, varies and, as it does, modulates into different experiences of situations, things, and the entire ensemble of their relations. In the ascending order suggested by James, conjunction moves from "with" to "my," building from exterior bare relations to intimate subject-oriented perspectives, terminating with a human-subject experience of relation to "their" world. However, James is emphatic that no one connection or ordering runs through all experience (1977, 197). This then makes his conception of experience an open relationality, potentially made and individuated by all kinds of entities, including technical ones. As David Lapoujade puts it, "Pure experience is the set of anything that is in relation with something else" (2019, 13). The empirical is just that domain of any and every relation in its/their process of relating, but James's emphasis on, and attention to, the *processes* of relating makes his version of empiricism radical.

What would it mean to bring this radical empiricist attention to processes of relating into ML and its now-dominant configuration of computational experience? It would mean attending to how technical entities and operations pass, conjoin, traverse, *and difference* via statistical and networked ensembles. The connections concatenated in (any postdigital) computation between entities such as numerical values are discrete; in a neural network, specifically, at the level of its various layers, values are drawn from data inputs and operated on, or values are functionally generated to produce synthetic outputs. Here we have a calculative relation across weights and biases—the parameters for a layer—and data inputs (transduced to numerical values) or other values generated by a random algorithm designed to feed the model some noise. We might say, then, that the calculations performed in a neural network are, at some level, discrete, maintaining the barest relations of "withness" or proximity between the layer and inputs or synthesized values.

These processes occur across one layer with potentially millions of parameters and inputs to produce a new set of values that pass the threshold of that layer to become features, passing to being calculated again until, eventually after many passes, they become the neural network's outputs. Yet such sets of values and constraints calculated together and against one another also easily accord with Lapoujade's reading of the openness of James's experience as a "set of anything that is in relation with something else."

If these computational processes meet the bare criteria of relationality, then they also generate or count in radical empiricist experience. They do not require the appearance of human subjects or objects with their sensory perception or intelligent perception for their operative, calculative experience to "count." This is not the same as saying that ML/AI is autonomous and can run without human control or action. The claims for autonomous AI—whether intellectual, creative, or even functional—are usually premised on the prior existence of the human subject: Can AI be as creative as humans? Is AI more intelligent than humans? Will AI be more efficient than humans by 2050? When using a Jamesian process-based account of ML computation as radically empirical, I seek to approach its operations, sensibility, and relationality differently, suggesting that their relationality counts in contemporary experience. But it also counts for us because its effects register. We only have to think back to Loab and sense that she is such a registration. The proposition for ML set out in this book is for a nonsensuous, liminal, imperceptible, *and* registerable-in-its-effects mode of radically empirical computational experience. This places ML computation/AI and



I.5 Schematization of a node in a neural network. The inputs are represented by X1–Xn. A weight, or W1–Wn, is added to these; 1 is the bias. Since all these diagrammatic elements are calculations and sums, this should be understood as a topological diagram of relational values in which the node is a value produced through all these relations.

Pop-Up Definition: Weights and Biases
A node in a neural network—also called a neuron—is a calculative outcome in which a data input (or collection of data points transduced to numerical values) is multiplied by a weight value and a bias value, or parameter. The overall calculation of weight and bias is then further subjected to another "activation" function. The activation function calculates which inputs' weights and biases exceed the threshold of activation (>1), and these are then passed onto the next layer of nodes.
Weights and biases are core calculations in a neural network, yet their activations may be difficult to detect for inputs that are highly variable (high-dimensional data such as images, for example), or in multilayered and massively connected networks. Weights and biases are usually initialized with arbitrary values, and it is the changes to these, and the modulating effects through the network, that become a measure of the model's "learning."

humans together through a relation of *nonrelation* when it comes to knowing and feeling one or the other's modes of individuating experience: we don't know what is happening in the black box of a neural network model; AI doesn't feel creative when Midjourney, a generative AI image platform, makes a new image when prompted with a text input. But it also means that the nonrelation of humans and AI might play out differently when it comes to tallying up what *happens* in computational experience. Relations, whether computationally generated or produced in human-computer conjunctions, are constantly registering in and as experience, and "we" all, together and differently, *experience the experience of ML computation*. The pragmatic beauty of James and of process thinking the processes of AI is that we move to interest not in "what is different" about computation (or the same as the human) but in how computation differences or makes both human and machinic experience heterogeneous. Inversely, attending to a radical empirical understanding of ML also allows us to register where computational experience fails at differentiating tendencies as it is harnessed and captured through social, cultural, and political arrangements of predictive computing.

As I have already suggested, ML computation is populated not just by quantitative but also by qualitative processes such as vectorization. We can already see that AI models now performing image recognition and image synthesis, and large-scale language translation and semantic generation are using many of the vectorial relations that James describes as "towards," "against," "because," and "through" via the very logic of their operations. I discuss some of these vectors with respect to image recognition and generation in chapter 1. So, while dealing with discrete quanta, ML processes are simultaneously concatenating operational pathways, and it is these concatenating vectors that *become* the model registering in experience as a dynamic and generative entity. Further concatenations and disjunctions occur via additional processes such as optimizing the model for its specific tasks, as well as human intervention and feedback as part of the model's development, which may be as simple a decision as selecting how many epochs or passes over data a model will run in its training. All these ML and human activities and actions constitute AI as a human-machine ensemble, laying down its relational order of connectivity. In his own time, James offered us a sense of experience being made through conjunctions traversing technical, infrastructural, institutional, and human components: "We ourselves are constantly adding to the connections of things, organizing labor unions, establishing postal, consular, mercantile, railroad, telegraph, colonial, and other systems that bind us and things together in ever wider reticulations. . . . From the point of view of these partial systems, the world hangs together from next to next in a variety of ways" (1916, 130–31). By paying attention to just such concatenations with respect to ML, as these are made by the operations of models and their *conjunctions* with humans, we can begin to register computation as a singular mode of operativity qualifying our relations of, in, and with culture(s) and technics. James's "experience," I will argue, is well suited to thinking a form of computation, such as ML, whose operations—despite their epistemic and social orientation toward predictive outputs—unfold via continuity and modulation or differencing. And, importantly, processes of differencing also offer contingencies and surprises.

ML's Actual Technics as Computational Contingency

A focus on computation's quantitative and axiomatic potential for novelty alone, which has been pursued in a range of approaches emphasizing computation's contingency, falls short, I think, of the *actual technics* of ML. By "actual technics," I mean two things: first, ML's ensemble of technical components and operations, which can be specific to a domain or operation such as image recognition or natural language production and carry a specific technical lineage through which it gathers its components together; and second, how these ensembles actualize as and through their sociotechnical milieu. As I will explain shortly, this milieu should not be understood as exterior to ML's technicity, where it is "it" that becomes responsible for situating technical elements according to broader epistemic, political, or cultural formations. In Fazi's account of the discrete, quantitative, and axiomatic nature of computation, for example, algorithms do participate in the broader world of social, political, and even material phenomena, but only in a secondary manner in terms of their application and implementation in the world. For her, algorithms are "the a priori intelligible" of computing (2018a, 106). By this, Fazi means that they are mathematical ideas that preexist their embedding into a program or code, and only a particular code or program a posteriori operationalizes them. At the onto-epistemological level, Fazi argues that a deterministic organization of computing operates when aesthetics and logic combine in "computational idealism," in which computational axiomatics are conceived and implemented as the horizon for determining both the ideal/transcendental and the empirical (92). For her, this is how algorithms become predictive or are embedded as forms of governance. But this implies that it is only when computation joins forces with another agenda or when it functions away from its own axioms that it becomes determinate.

But Fazi's conception of the algorithm as a priori axiom cannot account for the kind of computation that constitutes ML and, increasingly, is how computational experience is being made. Machine learning is not primarily a mode of *digital* computation but a statistical one. As I will also argue in chapter 2, algorithms that are part and parcel of ML's operativity germinate from a nineteenth- and twentieth-century *statistical* mathesis in which mathematical ideas are induced from the sociopolitical materialities of race and class relations of Anglo-American nationhood. The distinction between a priori intelligible and a posteriori implementation does not hold for the logic of statistically wrought algorithms. And since ML converges statistics and computation, we will need to look out for its entangling of temporally conditioning oppositions such as "prior" and "post" with respect to data, algorithms, and its entire operativity. Machine learning occurs at the nexus of statistical methods and techniques and computation and is, indeed, a reconfiguration of both. Even if (as we will see at many moments throughout the book) we cannot easily buy into the simple characterization of ML as the emergence of an algorithm from its learning on/of data, nonetheless we must account for a different ensemble in which an inductive technics is at work. Induction, inference, and probability bring different operations, logics, and implications than discrete, a priori axiomatics. As Mackenzie puts it, "Statistics has . . . gradually probablized machine learners" (2017, 104). But, as he also notes, ML reconfigures the classical statistical methods of sampling known sets of phenomena such as populations. Instead, ML begins with the premise of operating on all "known" data. Hence ML models are typically thought of as architectures for big data—a dataset of all that can be known. Of course, as Kate Crawford and danah boyd (2012) have already pointed out, claims to the comprehensiveness of big data are limited, since datasets are always in some way historically situated and undergo many processes of organizing and arranging that necessarily filter out data points. Nonetheless, as Mackenzie suggests, the difference between strict statistical samples and ML datasets lies with the latter containing all data "known" for the task at hand. The model itself—in contradistinction to the classical statistician who performed the process of sampling—then becomes the "knower" of the data. This automation of knowing occurs by parsing the data via continuous operations to detect and eventually attempt to eliminate error from its outputs. Mackenzie's point is that ML transposes a probabilistic logic from statistics to the model rather than simply using statistics' *methods*. The consequence of this is the automation of probability. Machine learning delegates the operations of sampling, analyzing, and observing, which account for error and prediction in data, to devices and to the operations of models. For Mackenzie, this also means that the potential for uncertainty—a key and immanent quality to the unfolding of practices of classical probabilistic statistics—is now ceded to recursive elimination performed via predictive operations: "The direct swapping between uncertainty in measurement and variation in real attributes that statistics achieved now finds itself rerouted and intensified as machine learners measure the errors, the biases and variance of devices" (2017, 106). The actual technics of ML is, then, inductive yet also unfolds through a milieu of automated prediction, which it simultaneously enfolds. This fundamentally alters its operative mode of computation from digitality and, I will be suggesting throughout this book, engenders different modes of computational experience.

As we will see, especially in chapter 2 when I look at the ways in which a statistical logic of racialization enters ML, this means that contemporary computational experience can never be easily debiased. The very operativity of this kind of AI runs on a singular trajectory in which techniques or functions of statistical *discrimination* have become immanent to its functioning. Statistical discrimination—through which many baseline algorithms of ML operate—constitutes the actual technics of ML as an already "biased" technical ensemble before any specific data inputs run through a model. Race, class, gender as operations of statistical discrimination become entwined in the core *automated* functioning of ML models. Yet even as ML has automated the project of statistics, it nonetheless remains open to the indeterminacies of probabilistic (statistical) techniques. This occurs regularly in AI models through phenomena such as category mismatches where, for example, images are matched to labels that do not indexically describe them, or when AIs perform in ways that "err" from, yet nevertheless conform to, their task specifications. In chapter 1, I look at ML operations in some detail with respect to image recognition and misrecognition and the ways in which misrecognition recurs across computer vision. My overall proposition—pursued via a close look at a range of computer vision AIs—is that the operativity of ML is not as closed and predictable as is often claimed. Instead, ML is a mode of computation in which indeterminacies are lodged in the operativity of its actual technics. The question will be: How can computational experience remain open to these?

On the one hand, then, we are faced with this delegation of error management—what we might also call the regulation of chance or indeterminacy—to the sociotechnics of predictive computation. On the other hand, its recursive operations, its actualization as part of an ensemble of conjoined algorithms, trained datasets and their legacies, and the contributions of human cognitive and affective interventions in making AI models operative all make ML less predictable. I want to propose that a deepaesthetics of ML must consider both the predictive reshaping of life through automation *and* the potential for new openings onto contingency and indeterminacy. This book engages, then, in a continuous double-pronged approach to teasing out and encountering a deepaesthetics of ML computation: one that recognizes the ways in which relations are reconfigured and often restricted by predictive trajectories; and one in which AI models, data scientists, cultural producers, artists, and theorists are alive to its odd sensibilities and indeterminacies. This requires a thinking of ML intensively and extensively as relational field; we cannot stop at the axiomatic or numerical registers of quanta prehending each other according to algorithmic procedures. We must, however, stay close to the technical specifications of the statistical computation that organizes ML. For it is at the granular level of computation's operativity that we can locate both the production of social propensities, problematic genealogies, seemingly predetermined trajectories and the potential for novel (aesthetic) events and experience.

ML's Machinic Universe and ML as *Agencement*

We should keep in mind that ML's terms, images, and diagrams do not belong to the technical infrastructure of AI alone; they consistently gesture to social, cultural, political, philosophical, and aesthetic ideas and processes. We have already seen this in the case of the layered architecture of deep learning models, which, at the same time, proffers images that tell us about the topological spaces conjured and inhabited by AI. What is particularly telling about the layer diagrams used to explain deep learning neural networks is that they are not images "of" technical components or technical (infra)structures, since the layers are neither physical nor even systemically representative of something technical in the way that a circuit diagram, for example, might be. Indeed, *there are no geometric layers as such* in deep learning networks, but rather only cascading series of numerical values, summations, and operations. But neural network diagrams and the image of the layer as a component of the computational architecture of deep learning computation must not be explained away as an image used to merely "communicate" computation to a nonexpert. The concept of the layer has been operative in ML research from at least the 1970s onward (see Ivakhnenko 1971). In this earlier period of ML research, three terms were used to describe the architecture of neural networks that learn: *hypersurface*, *layers*, and *thresholds*. The hypersurface functioned as a kind of projection of the overall connected topology of the network; the concept of the layer was used loosely to point to the net set of results of known transformations of "groups" of numerical values; and the threshold marked the transformation taking place. Interestingly, layers in contemporary deep learning research merge aspects of all these concepts (e.g., Goodfellow, Bengio, and Courville 2016, 164): they are the convergence point for the model's functions (the earlier layer), the ways these functions interact (the earlier threshold), and the vectorial chain conjoining one function to another (the earlier hypersurface). This later convergence demonstrates the unacknowledged epistemic work now performed by the concept of layer and of the layer diagrams, which often accompany research on neural networks (see fig. I.2).

As both concept and diagram, the layer is thus part of the "machinic universe" of ML, to draw on Félix Guattari's conceptual apparatus for thinking through technologies (1995, 36). For Guattari, any technical machine—a neural network, for instance—is a conjunction made possible through a composition of many elements: diagrams setting the range or vectorial possibilities and constraints of the technical object's feasibility; materialities that enable its production; an industrial sector producing it; a political economy, which finances it; and a collective imaginary that is the ethical and aesthetic condition for it being actualized (48). Taken together, these spheres form a technical ensemble, and it is the dynamic of their relations that brings a particular technical machine into being.[8] While there may be no geometrically shaped physical surfaces, no "places" in a neural network where layers can be located, and no volumetric depth to deep learning AI, layers are nonetheless diagrammatic and aesthetic components of the technical ensemble that is machine learning. They assign its computation a topological architecture and operativity; they also set the limits for imagining what a deep learning AI might be capable of doing.

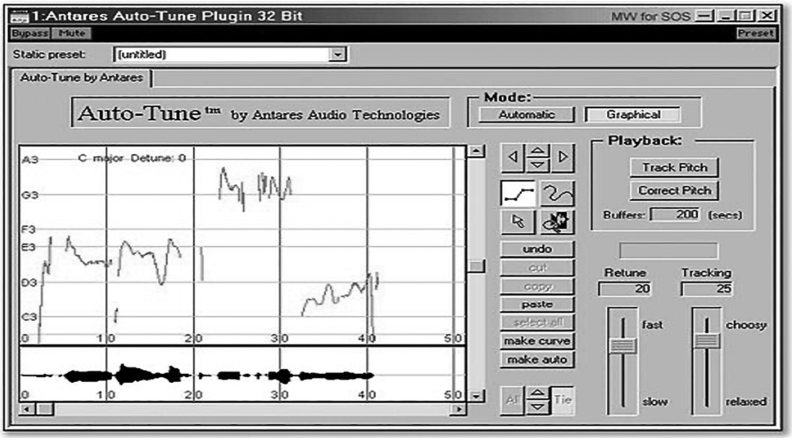
Yet it is not sufficient to lay out the fields—conceptual, diagrammatic, financial, aesthetic, and so forth—that constitute ML's technical ensemble. Gestures of mapping will not give us a sense of its actual technics, since simply pointing to this array results in a static and almost structural setting in place of ML, or any technical machine, for that matter. Instead, we need a sense of how these spheres come together *in relation*. *Agencement* is the term that Gilles Deleuze and Guattari coined to account for how heterogeneous social, technical, economic, aesthetic, political, biological, inorganic (and more) elements conjoin and multiply in ways that are productive of new relations and events: "An assemblage [*agencement*] is precisely this increase in the dimensions of a multiplicity that necessarily changes in nature as it expands its connections" (Deleuze and Guattari 2005, 8). Like other sociotechnical ensembles, the *agencement* of ML functions by increasing the *multiplicity* of its relations through conjoining with other machines whose dimensions may not be technical at all. This occurs at many different and disjunctive scales: from that of a function such as PCA, a commonly used dimensionality reduction algorithm, to that of the corporate imaginary of AI and its claim for predictive futures. Hence, when I pay close attention to the genealogy and operativity of a technical aspect of ML, as I do many times in this book, I do so to get at the ways in which ML opens onto, conjoins, and enfolds the social, epistemic, political, and aesthetic fields (and more) into it as part of its technicity, or as I have termed this, its "actual technics."

In analyzing, following, and landing on aspects of ML, I contend that computational experience can best be approached via this conception of *agencement*, since it gives a sense of how a technology is always operating within a technical ensemble that is *in process and relation*. The relations conditioning, and the new relations generated by, the *agencement* of ML are experienced not simply by humans but also at and by more-than-human machinic registers. Throughout the book, I retain *agencement* in its original French to set my approach apart from the concept, methods, and framework of what is now called "assemblage theory" (DeLanda 2019). Although there seem to be overlaps with that theoretical project—its emphasis on process, dynamic interrelations, and heterogeneity (Venn 2006, 107)—the assemblage, as the object of assemblage theory, drifts back toward an idea of an emergent system and sometimes promotes methodological scalability. Manuel DeLanda, for example, seeks to provide an entire ontology of social processes by developing a theorization from bottom-up emergent "wholes" such as "persons," understood as assemblages of albeit heterogeneous actions, through to economic or political emergent systems such as finance. This conception of assemblage theory, he states, offers, "an approach in which every social entity is shown to emerge from the interactions among entities operating at a smaller scale" (DeLanda 2006, 90). DeLanda's emphasis is not on processes but rather on entities and how, at each level or scale, one gives rise to or conditions another. This implicitly allows each emergent entity to be brought into relations of equivalence up and down the scale with one another. Perhaps this solves the issue of how to deal with *relating* things that are heterogeneous, but it does so at the expense of the heterogeneities! Instead, ML, as the *agencement* now dominating computational experience, requires an approach that *goes to the processes* activating and making it elastic, the processes that also result in it producing itself contingently. It is these processes of relation that are activators of heterogeneities.

A Simondonian Technics of ML

The concepts of both the technical ensemble and *agencement* in Deleuze and Guattari are indebted to the philosophy of technology elaborated throughout the work of Gilbert Simondon. This book and its thinking through of the actual technics of ML is likewise indebted to many of Simondon's technical concepts. Simondon himself was deeply skeptical about automation, but his skepticism arose not from computation's technicity but from its capture by a sociopolitical reduction of autonomous systems to human behavior and vice versa (2017a, 17). He argued for a restoration of technicity to computation that would acknowledge the "openness" of computational programming: "A purely automatic machine completely closed in on itself in a predetermined way of operating would only be capable of yielding perfunctory results. The machine endowed with a high degree of technicity is an open machine" (17). The sociotechnical program for automated predictability—the publicly declared and dominant agenda for ML in contemporary capitalism—presents just such a closed machine, often yielding cursory results. But this is not all the *agencement* of ML might be or become, and indeed many instances in the actual technics of ML suggest its potential for openness in this Simondonian sense. I turn now to an example drawn from contemporary "automated" music production to see how approaching AI both critically and with a degree of Simondonian openness gives us glimpses of the two poles of deep-aesthetics—predictive and indeterminate—operating in contemporary AI.

The claim that a "deep" *aesthetics* increasingly stakes for automating and autonomizing creativity has gripped the computer graphics and music industries in ways that suggest the *agencement* of ML is always already social, aesthetic, and economic, and more as it actualizes. Popular digitally produced music, for example, has been moving in the direction of deploying algorithmic correction of pitch and vocal timbre through software such as Auto-Tune since the 1990s. While we cannot tell a simple narrative about the uses to which automated digital pitch correction has been put—there are many artful, experimental, and minor configurations of Auto-Tune—the



I.6 Early interface for Auto-Tune, ca. 2000, which shows a recorded vocal input signal that could be retuned as its pitch is being tracked. The same capacities are also possible for live performance. Auto-Tune is an example of negative feedback being applied in music production.

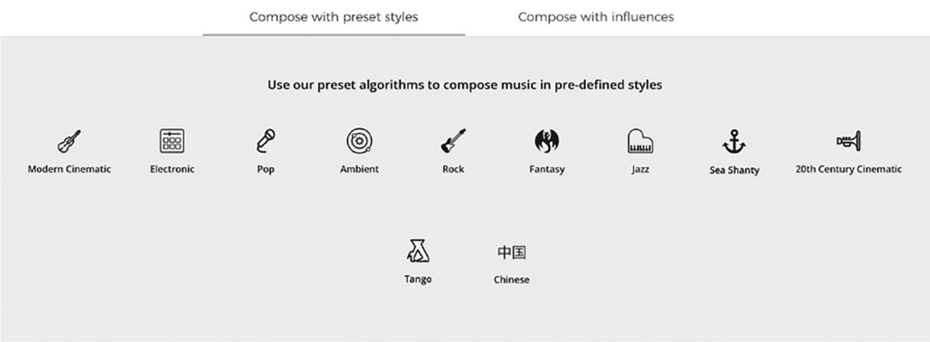
***Pop-Up* Definition: Negative Feedback or Cybernetic Recurrent Causality**

Negative feedback occurs when the outputs of a system, process, or operation reenter the system to affect its inputs in such a way as to stabilize further fluctuation in outputs. An example often given is a heating or cooling system controlled by a thermostat. The heated space is kept at a stable temperature by the system if ongoing cooler conditions (cold air entering the space or the temperature dropping owing to moisture, proximate conditions, and so on), detected as inputs to the system, are modified by the thermostat. This would result in the ongoing "output" or room temperature being kept at a constant warm temperature despite new inputs varying.

Negative feedback in cybernetics tends toward stabilization of a system by closing the difference between new information or variability and a continuous output of signal modulated by the system's internal operations.

assertions housed in its original patent nonetheless underscore the homogeneous tendencies that it has cultivated since its release in 1997: "Voices or instruments are out of tune when their pitch is not sufficiently close to standard pitches expected by the listener, given the harmonic fabric and genre of the ensemble. When voices or instruments are out of tune, the emotional qualities of the performance are lost" (Hildebrand 1999, 1). Here we move in seamless fashion from performer to pitch to listener to emotion, all to be navigated via an algorithm that performs an operation of *standardization*. The Auto-Tune algorithm works by automatically detecting an actual human performatively generating pitch; sending inputs to a corrector (a set of MIDI standardized pitches), which automatically correlates these to match the standardized pitches; and then re-outputting the performed pitch as corrected or retuned.

This input–correlation–output cycle is one of the simplest schemas for automating human-machine relations and falls within the shadow cast of what Simondon called cybernetic "recurrent causality," or what we more frequently call "negative feedback."[9] Here, (first-order) cybernetic design creates a link between "the chain of causality conveying the action and the chain of causality conveying the information" (Simondon 2020, 427) by literally capturing the latter (information) in a circuit for (re)producing the former (signal) as the key to the operativity of its system. For Simondon, this means that whatever is potentially novel about the information—whatever is contingent, in other words—is discarded to ensure the ongoing hegemony of seamless signal. In the operation of the Auto-Tune algorithm, the only information that comes to count is *the difference between the pitch produced by a subject and the pitch that needs to be corrected/produced by the software*. Consequently, the subject/performer/musician learns from that narrowed range to adapt their performance and behavior, linking their ongoing action/performance to receiving that correction alone. Accordingly, the kind of sound produced tends toward increasingly correlating the sung voice to the software's processes of standardization. Over time, a homogenization of vocal sound occurs unless other variants of information are introduced and explored. The issue here is not the dominance of "the algorithmic" per se but rather the privileging of operations of causal recurrence, which divest the human-machine relation of a broader milieu of variable information. The dominance of cybernetic recurrent causality or the paucity of generating multiple kinds of information for software and hardware systems has primed a good deal of music production and performance for further regularization via ML.



I.7 From AIVA's (a music composition tool using RNNs) website. This allows choosing preset styles that contour the music as it is made to fit into genres and even culturally specific music styles on which the model has trained.

***Pop-Up* Definition: Recurrent Neural Networks (RNNs)**

A **recurrent neural network (RNN)** is a type of neural network architecture used to learn from sequential data. Examples include letters in a word, words in a sentence or sequence of text, and musical sequences. A key characteristic of their operation is that data are inputted as vectors or ordered strings of numbers. These vectors can be stored in layers and used to sum, multiply, or subtract with another vector that is fed back into that layer. This allows, for example, predictive text to use other sequences of words (other vector states that have been stored) that supply contextual information to make correct "guesses" or predictions.

Recurrent neural networks are core architectures in the development of natural language processing. While now superseded by other architectures, they have provided the programmatic conditions of possibility for large-scale language text-to-image models to emerge. These use a network architecture called a transformer building on early research in which two RNNs were used together (encoders and decoders).

Enter Amper, IBM's Watson Beat, and Google's Magenta, deep learning models that are being used widely in computational music production. Although these are not stand-alone models—Magenta, for example, leverages TensorFlow, now a subplatform within Google for developing many kinds of ML algorithms—they all use recurrent neural networks (RNNs) or variants of them. Recurrent neural networks are models often used for processing data with sequences of values occurring over time. They store values as vector states—a sequence of data inputs transduced to strings of numbers—which subsequent states can connect back to iterate on. At a much larger scale and using many more recursions, they apply principles of feedback loops. Their neural training for music production usually occurs by feeding in many sequences of melody, chord, or rhythm progression data until the model learns the style of the music to be generated. Whereas in Auto-Tune the recurrent causality operates across software and human voice, in RNNs the recurrence works on preexisting digital samples of musical style and genre, which may be audio recordings or MIDI files. Importantly, this already suggests a transduction of the performative information, entailing the model using samples that have already been compressed or even pitch corrected by digital preprocessing algorithms. The RNNs can synthetically (re)produce genres and even moods as presets for musicians learning from these samples to generate new musical refrains, percussion sequences, and so forth. Crucially, then, at different times and scales, the models are learning data *already schematized* by processes of recurrent causality, embedded in the samples and widely circulating via types of signal processing—including Auto-Tune but also digital filters and effects, for example—and formatting (the ubiquitous MP3 file).

What, then, are the role and place of a human musician or performer working with such an AI if a certain schematization of human-machine activity has already been subsumed into the model? In what ways might human vocal performance introduce variable information or be subsumed into a schema of recurrent causality? We can look, first, to the example of the pop singer Taryn Southern, who released the album *I AM AI* in 2018 using both Amper and Watson Beat. A former star of *American Idol*, Southern had previously worked with music producers to create "structure" for her pop songs (Deahl 2018). Using AI music platforms, she chose genre and mood presets to generate a compositional base for her songs and then iterated melody and lyrics over the top of them. This is not so different from compositional strategies for many contemporary pop and electronic musicians, who use a range of digital audio workstations such as Apple's

Logic, which already integrates various AI operations into its software. But what is revealing is Southern's attribution to the AI of both compositional and cognitive agency as it learns preprocessed *samples*. At the same time, she gives her own musicality a programmatic character by describing her efforts as iteration or *recurrence*: "She knew 'very, very little about music theory. . . . I'd find a beautiful chord on the piano . . . and I'd write an entire song around that, but then I couldn't get to the next few chords because I just didn't know how to play what I was hearing in my head. Now I'm able to iterate with the music'" (Southern, quoted in Deahl 2018). Here we might say that the human slots into a diagram of doubled recurrent causality in working with the AI's predictive operativity. Since the model is already working in a recurrent mode deploying presets, the musician/singer portrays their melody as a pattern of recursion on top of the underlying recurrence being automatically generated. Here recurrence *recurs*, intensively within the model's own processes and extensively by drawing the human into its diagram. In this respect, the human-machine relation is configured so that all elements, human and technical, are seen to be functioning according to a reduction of information and variability to seamless signal output: AI presets and human iteration match each other. Any openness to something outside the programmatic, or what Simondon also calls a machine's "margin of indeterminacy" (2017, 17), is foreclosed. A fully automated mode of computation has no margin of indeterminacy, since theoretically it never varies from what it is predicted to be or to deliver. Variability and hence novelty or new information—the musician/singer, for example, improvising with the AI, and the AI varying and modulating because of that improvisation—could only occur in a technical ensemble that is also open to the unforeseen (39).

If we take an altogether different approach to working with an AI as an "ensemble member" within musician Holly Herndon's practice, we begin to see how she engages a sensitivity to a form of recurrence that is variable and contingent.[10] Likewise deploying RNNs, Herndon developed a combination of spoken and sung extended vocal audio samples from her own and other female singers specifically produced to train her AI, Spawn. An artful technique is here doubly articulated with respect to cybernetic recurrent causality: both refusing to attenuate her own voice to a predetermined set of canonical musical expectations such as standard pitches, and rejecting attenuation of the voice of the AI to pregiven data—even her own prior recordings—that could be mined for a style. Consequently, early experiments with Spawn generated pitch and rhythm continuity with the extended vocal technique tendencies of Herndon's music making but also developed variable machine-generated vocalizations. These sounded like a supra-beatboxing mode of vocalizing in which rhythm—rather than oscillating between the regular and syncopated—became irregular and contingent. Herndon (2021) notes that she didn't train the model to beatbox per se, that is, to explicitly generate synthetic vocal percussive sounds that in their live human form aim to imitate drum machines, samplers, and the rhythmic components of hip-hop. Instead, something analogous to beatboxing emerged from Spawn's synthesizing outputs by learning patterns of feature distribution across sung and spoken audio sequences made for the training dataset. Spawn's vocalizations have no consistent, stratified, or predictable beat. Instead, an echo of the style we name "beatboxing" is conjured as a musical gesture by the AI. As Herndon suggests, strict digital sampling such as simply taking a riff, rhythm, or vocal phrase and replaying it within another piece of music repeats past musical gestures, whereas "spawning builds variations from past expressions" (2021, 45:45). Machine learning here is artfully engaged by enabling differential recurrence via spawning rather than reproducing known expression of styles and genres. Alfred North Whitehead calls this the production of "intense experience," in which a graded set of contrasts emerge in relation to identity (ground, past expressions), ushering in the emergence of novel "aesthetic fact" (1978, 279). In Herndon's co-composition with Spawn, collective registers of musical phyla—extended technique, digital audio processing, and beatboxing—variably condition the performance of the individual human musicians and of the musicality of the AI.

The overall outcome of the Herndon-Spawn relationality is a recurrent vocalization at odds with itself—slightly out of phase rather than predictable. I will look to the ways in which artistic engagement such as Herndon's pulls at, twists, and cajoles the schemas that sociotechnically organize the overt deepaesthetics of ML. Careful and artful techniques can cut into, conjoin with, or jam open a margin of indeterminacy for AI. Artists, in this capacity, are not those who represent deep learning as either humanlike or transhuman, nor are they interested in drawing out its creative agency. Instead, they are the conjurers and crafters of artful techniques: artists, cultural producers, critical thinkers, and data scientists alike. These techniques and this "artfulness," as Erin Manning (2016, 46) calls art practices' capacity to generate new opportunities for relation and for living, produce opportunities for becoming sensitive to a "what else" for AI. Here Simondon's conception of what a thoughtfully deployed imagining of technics might do is as useful for ML as it was for the thermodynamics and cybernetics dominating the twentieth century: "We can consider the technical imagination as being defined by a particular sensitivity to the technicity of elements; it is this sensitivity to technicity, that enables the discovery of possible assemblages" (2017, 74).

How Else for ML?

Throughout this book's four chapters, artful techniques surface that bring the ML operations and deep learning models being rolled out in contemporary social, medial, and political contexts into new relations with their technicity. What these techniques hold in common is both a sensitivity to the actual technics of ML and a desire to ask: How else might AI become even amid its trajectory toward prediction? This requires staying close to "the technicity of elements" but also seeking out what Guattari calls the "allopoietic" dimension of all systems and processes (1995, 47). This is the collective dimension of alterity with which any entity or system is always already in relation, and which enables novel generative capacities. To consider organic life momentarily, genes, seemingly units that underpin the self-production of an organic system, can only reproduce life by carrying the potential mutations of their entire genetic phylum. The gene is not an isolated unit, then, but immanently retains a past of actual changes and hence the potential for future changes. In this past-future/present-past/present-future topology lies the gene's dimension of and for expressive alterity. Technologies too, while inorganic, immanently carry their pasts—the phyla of their realized and *unrealized* sociotechnical mutations. They too open onto other futures. These potential lines for unfolding provide conditions for different ways they might unfold than their current realizations.

In the first half of the book, I alight on two prominent areas of computational experience: automated image generation via deep learning in chapter 1, and the racializing potential of statistical techniques in chapter 2. My project in these chapters is one that engages closely with the actual technics of ML, remaining sensitive to the potential for an alternate technical imagination and the conditions for another becoming for AI. In chapter 1, I discover the ways in which experimentation with the category mistakes of computational vision takes deep learning models toward potential indeterminacy and an alternate deepaesthetics. In chapter 2, however, the actual technics of statistical racism and its increasing manifestation in social media, in training data and trained models, and in the algorithmic politics of all of this warrants a retraversal of the machinic phylum of ML. I focus on the two statistical techniques of PCA (which we looked at earlier in the context of dimensionality reduction) and linear discriminant analysis (LDA).

By tracing the immediacy of their relation to the social program of eugenics in the nineteenth and twentieth centuries, we can see that these techniques *operationalize* race in a manner that is key to the sociotechnical ensemble that becomes ML. It is not simply, then, that datasets are racially biased (or biased against other "recognizable" social groupings). Instead, statistics, in its deployment of discrimination techniques in a large-scale automated mode—that is, platform-enabled ML—shapes and contours data so that they are distributed toward whiteness and away from Black, Brown, Yellow, or other kinds of "color" experience(s) of life, race, and bodies. Here, again, we could say that the *agencement* of ML closes computational experience to any variation that is not already predetermined along a spectrum distributed according to whiteness as its normative curve. What would it take to *artfully* prize open this deeply racist aesthetics to other kinds of experience? Throughout chapter 2, I visit the work of Stephanie Dinkins, whose proposal for an "Afro-now" AI lays out just what it might take to really generate different color spectra for computation. In Dinkins's practice, we see a refashioning of ML as a different kind of assemblage, whose conjunctions hold through relations of experiencing and experiences of making Black and colored life through familial socialities, together with participatory design engaging Black and colored communities of (technical) practice. Against the racism of statistics' eugenic genealogy and platform AI's deepaesthetics, Dinkins deploys deep learning models as open, contingent, nonscalable, and differentiating ensembles.

The operative rhythm of chapters 1 and 2 moves according to a pulse that first locates the potential of ML's machinic "error" (computer vision's category mistakes) and then genealogically traverses the very operativity of statistical computation as *agencement*. Chapters 3 and 4 invert and convolve the maneuvers of the book's first half, accenting what is often cast—for example, in speech pathology—as a disfluent and neurodiverse "outside" of language as condition for the possibility for the technical ensemble of *natural* language programming. Then, in chapter 4, I return to how contemporary artists engaging ML tease out its potential as *errant agencement*. In chapter 3, I am keen to show how AI's fashioning of "natural" language ontogenetically leans on what is disavowed in the quest to make its artificial agents speak seamlessly. This disavowal involves both an incorporation and a denial of nonlinguistic disfluent aspects of language production. The actual technics of AI language agent development enfold the affective and neurodiverse asignifying conditions of linguistic sense making so that these become the conditions of possibility for an AI-human conversational relation. If race is the unacknowledged *interior* core of statistical operativity and hence propels the functional continuity of ML models, then the *exteriority* of (normalized "fluent") speech—disfluent and neurodiverse communication such as stuttering—is what affectively enables the project of a *natural* AI-generated language. By the end of the book, having traveled back and forth across ML as *agencement*, I return to the generativity of the mistakes made as ML performs functionally or on task. I suggest these are less category mistakes to be fixed by recategorizing, and more a mode of persistently erring that is lodged in ML's operativity. This errant mode is being drawn on by a range of practitioners interested in fashioning a critical sensibility for AI via techniques of artful modeling, crafted datasets, and sensory and insensible experiments with ML operations and processes.

Together, these and related techniques begin to constitute an artful mode of knowing the operations of a technical system, which Simondon (2020) called "allagmatics." Here, analogically tracing and enacting system operations dynamically reanimate the (structural) elements of a technical ensemble, resituating how the system's technicity might be made known, "by defining structures based on the operations that dynamize them, instead of knowing by defining operations based on the structures between which they are carried out" (Simondon 2020, 666). The artful techniques devised by artists who critically engage ML reperform the operativity of AI yet simultaneously diverge from its homogenizing predictive structuration. Instead, these techniques analogically enact *differential* repetition of the contemporary *agencement* of machine learning. Their artfulness lies in the extent to which the artworks retain their margins of indeterminacy between AI's procedural playing out within current sociotechnical constraints and a carefully considered deplaying, a making errant of AI that subsequently makes it wander away from being "on task."

NOTES

Introduction

1 The discussion of Swanson's Twitter thread points to a thread that she generated as a series of replies to herself to explicate her process and process-based thinking around the generation of Loab. The thread was originally located here: https://twitter.com/supercomposite/status/1567162288087470081 (accessed March 20, 2023). However, Swanson has since made the replies and thread invisible.

2 These include a wide range of models and functions and a wide range of levels of access in terms of knowledge and cost. While the deep learning models that subtend text-to-image model prompts are relatively accessible in terms of consumer cost and interface, their running is only made possible through large-platform resourcing by, for example, Google, Microsoft, OpenAI, Meta, and so on. Although many widely deployed functions within ML facilitating data compression, for example, are "freely" available via resources such as GitHub, they require at least an intermediate degree of knowledge about data science. The degree to which machine learning dominates the production and distribution of contemporary cultural and knowledge production via everything from the elite pitting of humans against deep learning–enabled chess models to the algorithmic arrangement of music in streaming services is ultimately made possible by financing funneled through platform capitalism. It is costly to scrape data, train

models through many iterations, and furnish the ongoing infrastructure to run machine learners. OpenAI is widely reported to have estimated that the running costs of training a model will have increased from $100 million (for GPT-2) in 2020 to at least $500 million for a new model by 2030. See Knight 2023. Corporations such as Nvidia and Google have been able to capture much of the machine learning market by investing in vast quantities of graphics processing unit (GPU) infrastructure, which is then "rented out" in various array bundles via cloud services for smaller start-ups in the generative AI space, for example, Hugging Face and Databricks, to train and optimize open-source models. This is leading to a new kind of computational divide being labeled the "GPU-rich" and the "GPU-poor." See Patel and Nishball 2023.

3 To be clear, Fuller and Weizman's argument contains a further dimension, which concerns the quality of contemporaneous sensemaking, which they call *hyper-aesthetics* (2021, 57). Hyper-aesthetics involves the intensification, amplification, and synthesizing of sensing and surfaces for sensing, gauged not only as technologies for sensing multiply throughout environments and life but also as life itself becomes hyper-aesthetic. They furnish the example of changes in atmospheric conditions such as the halting of the jet stream in Europe during 2018, which intensified the summer heat. This was sensed not only by satellites, weather, and climate modeling but also by berries growing in northernmost conditions, absorbing the changed atmospheric conditions and becoming sweeter, and humans becoming hotter and sweating more (59). Hyper-aesthetics not only names the present sensing and sensemaking relations but becomes, for them, a method for investigating the politics of these relations.

4 This site was authored by different Google researchers. See Olah, Mordvintsev, and Schubert 2017.

5 My use of pop-up definitions is confluent with Erin Manning, Brian Massumi, and the collective work of Senselab's (a large collective network of humans and more-than-humans operating internationally since the early 2000s) concept of "pop-up propositions" (Manning and Massumi 2015). Here the idea is that a proposition or definition is provisional and arises in the middle of thought and events. It may help both change directions, take off, or reform but is not intended to "solidify" or act authoritatively. All my pop-up definitions take terms that are difficult to pin down within data science but are constantly swirling through it. By boxing these terms, I try to grasp at the difficulties of definition while still trying to bring some glossing of the term so as to allow further thinking and possibilities for it. I draw on a plethora of ML papers, textbooks, diagrams, platform AI blogs, and images that are already part of the data science literature informing the research throughout this book.

6 The constant updating of both data and ML systems is also commented on by Taina Bucher, who draws attention to the unfolding labor and operationality of ML models in a platform such as Facebook (2018, 28).

7 My interest in artful techniques for engaging ML resonates with Erin Manning's attention to art *practices*, which open up questions, manners, and concerns with process (2016, 46). Manning argues that art as a "way," manner, or mode has been eclipsed historically after the medieval period in Western culture by a preoccupation with art as an "object." Manning's and my process-oriented conception of "artfulness" stand in stark contrast with a tendency within, for example, software engineering scholarship and more informal commentary from programmers to describe it as "artful." In a tradition that stretches back to, perhaps, Donald Knuth's 1974 Turing Award lecture, "Computer Programming as an Art," software engineers have claimed that skillfulness along with the aesthetic beauty of programs makes computer programming like art or "artful": "We have seen that computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty" (Knuth 1974, 673). However, these concerns are clearly oriented to the cleverness and elegance of *solutions* to problems that programs and programmers deliver, and distinguishes this from the artfulness I am identifying as a manner and mode of (critically) probing computational processes and engendering novel sensibilities via engagement with computational models and ML techniques.

8 Guattari uses the example of the Concorde plane to illustrate the ways in which the relations of its technical ensemble to finance and economics produce its singularity. Only twenty Concordes were ever built, in no small part owing to the enormous quantities of fuel required for the jet to fly at supersonic speeds. Because the first Concorde launched in 1976, its flights took place during the energy crisis of the 1970s. It thus was ever only available to a small group of elite passengers able to pay the exorbitant cost of travel. And this was indeed the aircraft the Concorde became: a luxury supersonic jet (see Guattari 1995, 48).

9 Yuk Hui has argued that Simondon's "recurrent causality" and "internal resonance" can be collapsed back into the same concept (2019, 169) and that this makes them, and indeed Simondon, to an extent compatible with a general cybernetics program related to recursivity. I disagree with this reading; Simondon's internal resonance functions on a different register—that of conditions. Internal resonance is not internal to a specific technical object but rather operates across a phylum of technical objects. It is the conditions and the conditioning a technical individual (the technical machine individuating within its ensemble) facilitates, that together allow another new technical individual to become possible: "We could speak of

an internal resonance of the technical universe within which each technical being effectively intervenes as a real condition of existence for other technical beings" (Simondon 2020, 416). This resonance is not predetermined but rather is generated as a technology unfolds. An example would be the internal resonance of computer graphics cards (GPUs)—originally a high-performance image-processing peripheral for computer gaming—with machine-learning-enabled computer vision. This internal resonance only became available through a convergence of technical, economic, and political factors that conjoined large-scale deep learning, parallel processing, the standardization of benchmark image datasets, and communities of practice revolving around computer vision challenges. These all converged after 2009 and the release of the ImageNet dataset. Internal resonance is due to the *agencement* of machine learning, and not to the design of specifically cybernetic feedback systems.

10 Holly Herndon has been working with the AI model she has trained, Spawn, since 2017 in both live performance and studio album recording, particularly in the album released in 2019, *Proto*. See https://www.hollyherndon.com/.

Chapter 1. Heteropoietic Computation: Category Mistakes and Fails as Generators of Novel Sensibilities

1 This was a collaborative project with Adrian Mackenzie and Kynan Tan, titled "Re-imaging the Empirical: Statistical Visualization in Science and Art," supported by the Australian Research Council Discovery scheme. Further information about the broader project, especially concerning our aims and methods, is available at https://github.com/re-imaging/re-imaging/wiki.

2 ArXiv is owned and operated by Cornell University as an online open-access repository at https://arxiv.org. ArXiv maintains a large and increasing quantity of e-print articles from a range of scientific fields and provides a platform for authors to share articles before or during peer review.

3 Adrian Mackenzie and I have elsewhere explored these experiments (Mackenzie and Munster 2022) via William James's concept of pure experience. We also look in more detail at the problems of AI image models that have been originally trained on a data ontology comprising "things" or entities, which cannot then recognize—and hence mismatches—diagrammatically configured scientific images.

4 The rabbit-duck illusion involves a literal switching of perspective—a tiny movement or set of movements—in which the eye or visual sense engages a proprioceptive history of looking at images from different perspectives. To "see" the duck when one can only "see" the rabbit requires a flexibility