A Model of Understanding in Deep Learning Systems

David Peter Wallis Freeborn
Northeastern University London

Abstract

I propose a model of systematic understanding, suitable for machine learning systems. On this account, an agent understands a property of a target system when it contains an adequate internal model that tracks real regularities, is coupled to the target by stable bridge principles, and supports reliable prediction. I argue that contemporary deep learning systems often can and do achieve such understanding. However they generally fall short of the ideal of scientific understanding: the understanding is symbolically misaligned with the target system, not explicitly reductive, and only weakly unifying. I label this the Fractured Understanding Hypothesis.

1 Introduction

John von Neumann reportedly claimed that "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk" (Dyson, 2004; Mayer et al., 2010). Today, we are in the midst of a societal and scientific revolution, driven by deep learning, a powerful curve-fitting technology. Over the last few years it has driven major advances in natural language processing (Brown et al., 2020; Devlin et al., 2018; OpenAI, 2023; Touvron et al., 2023; Vaswani et al., 2017), computer vision (Dosovitskiy et al., 2021; He et al., 2016), image generation (Ramesh et al., 2021; Rombach et al., 2022; Saharia et al., 2022), multimodal learning (Alayrac et al., 2022; Radford et al., 2021; Ramesh et al., 2022), game-playing (OpenAI et al., 2019; Silver et al., 2016; Vinyals et al., 2019), and protein structure prediction (Jumper et al., 2021). In each case, deep learning-based architectures with millions or billions of parameters are trained on vast datasets and then adapted to downstream tasks not explicitly specified in their training objectives. We will need to understand the epistemic capabilities of these systems if we are to deploy and govern them responsibly.

Yet the epistemic capacities of deep learning systems remain unclear. Much of the debate turns on whether these models genuinely *understand* their target domains, or instead produce the mere appearance of understanding through interpolation, pattern-matching, and recombination of memorized fragments. Above all, their successes are strikingly uneven, forming a "jagged frontier" (Dell'Acqua et al., 2023): the best models can outperform humans on some difficult tasks, while failing badly on others that humans find comparatively easy. The use of deep learning models invites two contradictory reactions:

1. *How could the models possibly achieve this, unless they possess genuine understanding?*

2. *How could the models possibly fail at this, if they possess any genuine understanding?*

Skeptics argue that structures like deep learning systems are *incapable* of understanding (Bishop, 2021; Floridi, 2023; Pearl, 2018); in the case of large language models (LLMs), they are said to be "stochastic parrots" or "blockheads" that mimic competence without grasping meaning (Bender et al., 2021; Bender and Koller, 2020; Block, 1981; Marcus, 2018). Proponents counter that the *best explanation* for the impressive capabilities exhibited by frontier deep learning systems is that they exhibit at least some degree of genuine understanding. They point to the striking ability for deep learning models to solve tasks they were never explicitly trained to do (Bommasani et al., 2022; Brown et al., 2020; Grzankowski et al., 2025a,2; LeCun et al., 2015; Millière and Buckner, 2024a,2; Wei et al., 2022). Following Sellars (1963), we might distinguish two *images* of deep learning systems. On the scientific image, they are best described as highly elaborate curve fitters, whereas on the manifest image, they appear as agentic systems with some capacity for understanding.

This academic debate has serious real-world consequences, because high-stakes decisions increasingly rely on the outputs of deep learning models. If they generate accurate predictions *only* within the narrow support of their training data, they may fail catastrophically when deployed in novel settings, a phenomenon sometimes called "brittleness under distribution shift" (Amodei et al., 2016; Schneider et al., 2020). Conversely, if they do possess forms of systematic understanding, then those forms need to be articulated, and perhaps made more legible to human stakeholders. The same question bears on engineering choices between scaling and incorporating explicit causal abstractions, hybrid neuro-symbolic modules, or mechanistic interpretability constraints (Amodei et al., 2016; Bereska and Gavves, 2024; Doshi-Velez and Kim, 2017; Kaplan et al., 2020; LeCun, 2022; Rudin, 2019; Zhang, 2024), and on forecasts about more general intelligence (Bubeck et al., 2023; Chollet, 2019; LeCun, 2022; Wei et al., 2022).

Unfortunately, the dialectic has been confused, with opponents largely talking past each other. The dialectic lacks an agreed conceptual framework for understanding "understanding". In this paper I propose a model of what I call **systematic understanding**, an explication of one kind of understanding that I hope is rigorous enough to give some foundation for the debate, while also being general and flexible enough to apply to machine learning systems, as well as humans and other kinds of epistemic agents. This notion of understanding is deliberately non-anthropocentric, so that it could, at least in principle, apply to non-human, mechanical systems.[1] Then I assess whether deep learning systems in fact *do* exhibit systematic understanding, of at least some salient properties, for some real-world systems.

[1]Even for those who would prefer a different notion of understanding, hopefully providing this model will offer a starting point for alternatives to be explicated.

Roughly speaking, for an agent to possess systematic understanding of some property of a target domain, they must have an internal model of the target system, capturing some of its salient regularities, and enabling them to make adequate predictions about this property systematically from that underlying model. That is, the predictions do not arise from luck, memorization, or blind interpolation, but rather because they have internalized some *real patterns* in the target system. Following Dennett (1991a), I define a pattern in a body of data as *real* when there exists a finite, algorithmic description that (i) compresses the data more efficiently than brute-force enumeration and (ii) supports predictions of the existing or future data with a reliability that materially exceeds chance. Crucially, understanding on this account comes in degrees. At one extreme, a system that merely memorizes its training data achieves no genuine compression of the target's regularities and thus possesses no systematic understanding. At the other, a system whose internal model compactly captures the target's dependency structure and supports successful predictions across a wide range of conditions possesses a high degree of understanding. Most real systems fall somewhere between these poles. I will use this notion of systematic understanding to argue that deep learning systems possess a real, but **fractured understanding**, in a way that should make sense of the two contradictory intuitions above.

I have two distinct aims, one primarily conceptual and one primarily diagnostic. The conceptual aim is to propose a deliberately *thin*, non-anthropocentric model of *systematic understanding*, intended as a basic framework for clarifying disputes about whether deep learning systems understand their target domains. The account is offered as a proposed explication: it introduces a regimented notion of understanding that can be applied to artificial systems, and that helps to locate disagreements more precisely, even for readers who ultimately prefer a different notion. The goal is to supply a tractable framework for assessing which epistemic capacities are present in current systems, what kinds of failures those capacities exhibit, and which engineering choices predictably shape the resulting form of understanding.[2]

The diagnostic aim is to use that framework to characterize a recurrent pattern in contemporary deep learning. I argue that current systems can and often do satisfy the thin criteria for systematic understanding in restricted respects, yet that the form such understanding typically takes is not well captured by the traditional ideal of scientific understanding. In particular, learned representations are often misaligned with the target domain's natural variables, and competence is frequently distributed across locally reliable fragments rather than unified into a small stock of reusable principles. This motivates the *Fractured Understanding Hypothesis* developed later.

Note that I will focus on neural network systems more generally, rather than natural language processing or large language models (LLMs) specifically. The reason is that applying this approach of systematic understanding to LLMs would require a further,

[2]I do not claim to analyze the ordinary-language concept of understanding, nor to identify an underlying natural kind. Nor do I claim that satisfying the present criteria is sufficient for the strongest kinds of scientific understanding, especially those associated with causal-mechanistic explanation, although I hope it might provide a basis that could be adapted for such accounts.

specialized treatment. The framework here suggests that applying systematic understanding to LLMs requires answering two distinct questions in sequence: first, whether they acquire a systematic understanding of human language, and second, whether that linguistic understanding endows them with a wider understanding of the "world" (including a **world model**). These are genuinely different questions requiring different kinds of evidence and different bridge principles, and conflating them has been a source of confusion in the existing debate. By working through cases where the target system is well-defined and the bridge principles are (relatively) transparent, in geometry, algebra, game dynamics, I aim to secure the framework on firm ground before it is extended to the harder case. With that said, transformer models will provide two of the key case examples, in one case with highly structured linguistic data.

In section 2, I offer a brief definition of what I mean by a deep learning system. In section 3, I give an overview of the existing dialectic around understanding in deep learning. In section 4, I explicate a model of understanding, which I call **systematic understanding**. In section 5, I present a model of how deep learning systems model the world, with the aim of explaining the senses in which deep learning systems can or cannot achieve systematic understanding. I supplement this with four examples in section 6, each designed to pump different intuitions about the kinds of systematic understanding that current deep learning systems are well suited for developing. In section 7, I propose a **fractured understanding hypothesis**, according to which deep learning systems often achieve systematic understanding, but that this understanding is highly fragmented and rarely completely general. I conclude (section 8) with some lessons for both philosophers and artificial intelligence developers.

All computational models and experiments discussed in this paper are implemented in a set of iPython Jupyter notebooks, available at: `https://github.com/DavidFreeborn/Model_of_Understanding_Deep_Learning`. The repository is intended as a transparent companion to the paper rather than as a general-purpose software library.

2 Deep Learning Systems

Machine learning systems are algorithms that improve their performance on a task by adapting to accumulated information, rather than through explicit programming of every rule. Through some specified learning technique, the systems are exposed to data, often in the form of input–output pairs, and inductively infer a mapping from inputs to outputs that can then be applied to new, unseen cases. At its most basic level, we can think of the task of a machine learning system as learning to approximate some real-world function, as an internal function f_θ, by altering various *parameter values*, θ. Together, the function encodes all of the system's information about the target domain.

Typically, we divide the data into **training data** (examples that the learner actually *sees* during the parameter-adjustment phase) and **test data** (a previously unseen set of examples). Given training data $(x^i, y^i)_{i=1}^n$, where each $x^i \in \mathcal{X}$ is an input (often a vector in $\mathbb{R}^d$, for some dimensionality d) and each $y^i \in \mathcal{Y}$ is a desired output (a scalar, vector, or label), a machine learning system learns a function

$$f_\theta : \mathcal{X} \to \mathcal{Y}. \tag{1}$$

The aim is for f to correctly predict the outputs for new inputs (i.e. test data), suggesting that the machine is not merely memorizing the training pairs but has internalized genuine regularities (Goodfellow et al., 2016; Hastie et al., 2009; LeCun et al., 2015).

Deep learning refers to a family of machine learning methods in which the model's parameters are organized across many (N) layers, typically in a structure called a multilayer **neural network**. We call the layers between the input layer, which receives the data, and the output layer, which produces predictions, the **hidden layers**. In a feed-forward neural network[3], each layer j transforms its input through a composition of an affine transformation $W_j(x_{j-1}) + b_j$ and a nonlinear activation function, σ_j, where $W_j, b_j \in \theta$ and x_j refers to the inputs from layer j. So we can express the whole function as,

$$f_\theta(x_0) = \sigma_N\big(W_N\, \sigma_{N-1}\big(\cdots \sigma_1(W_1 x_0 + b_1)\cdots\big) + b_N\big), \tag{2}$$

where x_0 is the initial input data. The combination of depth (many layers) and scale (large numbers of parameters) allows these methods to approximate extremely complex functions.

For example, take a neural network trained to classify images as depicting a cat or not. Each input x_i might be a vectorized image (say, a 28×28 pixel image flattened into a 784-dimensional vector), and y_i are the corresponding labels, either *a cat* or *not a cat*. The neural network learns a function that assigns a probability distribution over labels to each image. We can think of the resulting function as specifying a **decision boundary** (e.g. $f_\theta = 0$), a surface defined over the input space, classifying what it thinks is or is not an image of a cat. More loosely, perhaps we might think of the function as giving us the machine learning system's model of *catness*.

What does this learned surface look like? For simplicity, let us suppose that the activation functions, σ_i, take the most common form, that of a **Rectified Linear Unit (ReLU)**[4], defined as,

$$\sigma_i(z) = \max(0, z). \tag{3}$$

This function is continuous and piecewise affine: it is linear on each side of the origin but introduces a kink (a point of non-differentiability) at zero. The effect of combining these piecewise affine activations in each layer is to partition the input space $\mathbb{R}^d$ into convex polytope regions, within each of which the function behaves as an affine transformation,

[3]For simplicity, henceforth, I will discuss feed-forward neural networks, in which information flows strictly in one direction from input to output without recurrence or cycles. These are the most common kind, but general arguments of this paper apply more broadly.

[4]This greatly simplifies the arguments. However note that the general following arguments about spline interpolation will apply for any piecewise polynomial activation functions.

$$f_\theta(x) = A_k x + b_k \text{ for } x \in \mathcal{P}_k, \tag{4}$$

where $\mathcal{P}_k$ is a polytope in the partition induced by the activation patterns of the ReLU units, and A_k, b_k are matrices and vectors determined by the weights θ. This piecewise-affine structure is a form of multivariate spline, and so this view is sometimes known as the **spline theory of neural networks** (Balestriero and Baraniuk, 2018).[5]

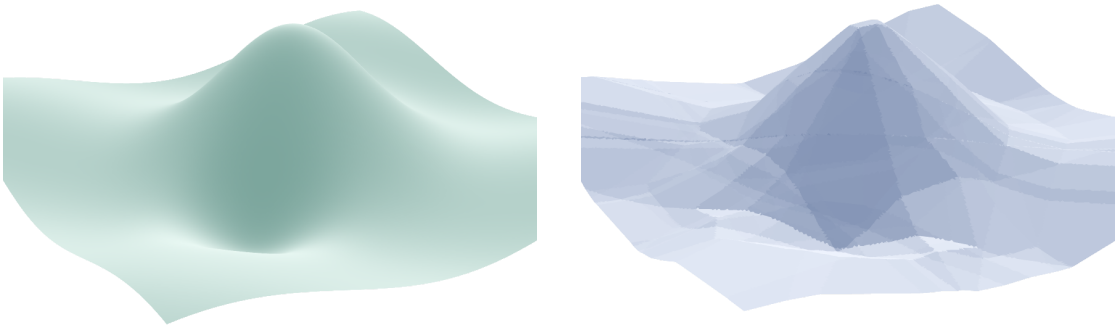
Thus, the learned surface of such a neural network consists of a complex arrangement of locally linear regions, stitched together into a continuous, but non-differentiable function. The network learns by adjusting how these linear patches are assembled to best approximate the training data. Figure 1 provides one visual example.[6] So taken literally, neural networks really are sophisticated curve fitters; all the information that the neural network learns about the world can be understood as merely parameterizing this spline-like surface.

3 Proxy Battles over Understanding

The debate over whether deep learning systems truly *understand* has been intense, and often confused. One source of confusion is that "understanding" itself is rarely defined with enough precision to support productive disagreement. In practice, machine learning researchers often sidestep the term, treating it as too vague. Yet concerns about understanding continue to motivate much of the surrounding discourse. What has happened, I suggest, is that these concerns have often been displaced onto more tractable proxy concepts that capture only part of what is at stake. Two of the most important proxy battles concern whether machine learning systems genuinely generalize or merely memorize their training data, and whether they extrapolate beyond their training regime or merely interpolate within it. Both debates are important in their own right, but both can also be read as attempts to operationalize specific aspects of the deeper question about understanding. However, as we shall see, these proxy debates have also been

[5] The term "spline" originally referred to a flexible strip of wood, metal, or plastic used to draw smooth curves through a set of points. Mathematically, a spline is defined as a piecewise-polynomial function that is smooth at the boundaries where pieces connect. But here "spline" is meant in a looser sense; for ReLU networks, the resulting map is *continuous piecewise-linear*, not a smooth function (Balestriero and Baraniuk, 2018; Raghu et al., 2017).

[6] For this, I trained a multi-layer perceptron (MLP) neural network consisting of three linear layers with 24 neurons in each hidden layer ($3 \to 24 \to 24 \to 1$) and ReLU activation functions, totaling 697 trainable parameters. The network is trained to regress an implicit scalar field of the form $F(x, y, z) = z - f(x, y)$, where $f(x, y) = 0.5 \cos x \cos y + 2.0 \exp\left(-\frac{x^2+y^2}{1.5}\right) - \exp\left(-\frac{(x-1.5)^2+(y-1.5)^2}{0.5}\right)$, which defines a smooth height-field landscape with a central peak, oscillatory structure, and an off-center crater. The training data consist of 262,144 points sampled from a regular 64^3 grid over the domain $[-3.5, 3.5]^3$, with target values given by the analytic implicit function. The model is optimized using Adam with learning rate 10^{-3} for 4,000 epochs, minimizing the Mean Squared Error between predicted and ground-truth scalar values. After training, the learned implicit surface $\hat{\Sigma} = \{(x, y, z) \mid f_\theta(x, y, z) = 0\}$ is extracted using the Marching Cubes (Lorensen and Cline, 1987) algorithm on a higher-resolution 200^3 evaluation grid. See the accompanying repository for the implementation details.



(a) Ground truth surface (b) Learned isosurface

Figure 1: Comparison between a ground truth surface (left) and the neural network's learned isosurface (right). Observe how the learned isosurface is made by stitching together piecewise-linear surfaces. The ReLU activation functions lead the network's output to be a continuous piecewise-linear function (a multivariate spline), approximating the smooth target curvature through a vast number of local planar regions.

confused by a lack of common, clearly understood terminology. Making the connections explicit will help motivate the framework developed in the next section.

3.1 Memorization and Generalization

If a machine learning system genuinely *understands* the target, by internalizing real patterns or regularities in the data, then we should expect it to make good predictions about new, relevantly similar data. This ability to extract underlying regularities from the training data and to apply them in order to make accurate predictions on previously unseen data is called **generalization**. On the other hand, **memorization** refers to the rote storage of specific training data. Memorization is often characterized by *overfitting*, in which a system learns noise in the training data, resulting in poor performance when predicting on new, unseen data (Hastie et al., 2009; Vapnik, 1998). The operational definitions vary significantly (Wei et al., 2024); however, the debate over whether a model genuinely generalizes, rather than merely memorizing, is often best read as a proxy battle about whether machine learning systems can acquire some level of genuine understanding.

Early theoretical arguments from statistical learning theory suggested that memorization might contribute significantly to the success of deep learning systems (Arpit et al., 2017; Feldman, 2020; Zhang et al., 2017). Roughly, if a deep learning system has more free parameters than there are distinct training examples, then it is theoretically possible for the model to perfectly memorize each example by assigning each one a distinct degree of freedom. More precisely, **model capacity** can be thought of as

the flexibility for a machine learning system to represent a wide variety of patterns.[7] Low-capacity (high-bias) models generalize poorly because they cannot capture the target system's full complexity; high-capacity (high-variance) models risk overfitting by tailoring themselves too closely to the finite and noisy data.

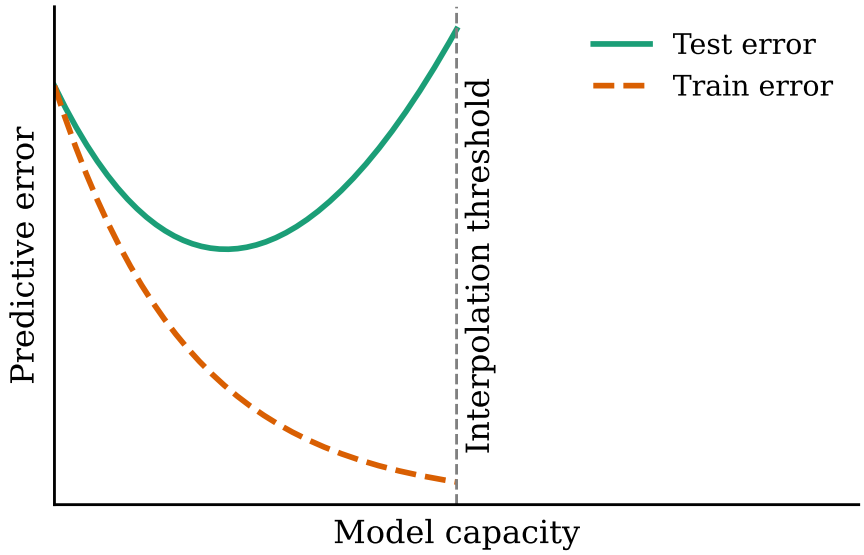
Counting arguments suggested that memorization and generalization are in tension: not every training example can be memorized uniquely without harming the system's ability to extract general patterns (Achille et al., 2020; Achille and Soatto, 2018; Hestness et al., 2017). More generally, theoretical arguments predicted that once model capacity exceeds a certain threshold relative to the dataset size, memorization should dominate over generalization. When this happens, a model's predictive performance should deteriorate when the model is deployed outside the memorized training data. This phenomenon is sometimes described as the bias-variance tradeoff, schematized in figure 2a.

However, landmark experiments shattered these expectations. Zhang et al. (2017) demonstrated that modern neural networks with millions of parameters can fit *random* labels perfectly and yet still generalize well on *real* labels, provided the data are structured. In other words, the same architecture that can *memorize* unstructured noise to zero training error simultaneously *generalizes* impressively on structured data. Further work revealed a more complex relationship between generalization and memorization. Rather than following a simple U-shaped curve in figure 2a, modern deep networks often exhibit a **double descent** phenomenon shown in figure 2b (Belkin et al., 2019; Nakkiran et al., 2021; Rocks and Mehta, 2022). As the system's capacity grows, test error first decreases, then increases around the interpolation threshold (where the model first attains zero training error), and then decreases again. The second descent suggests that supposedly over-parameterized models can re-enter a generalization-friendly regime, perhaps because the standard methods of optimization bias them towards seeking out simpler, more stable solutions. A distinct but related phenomenon is *"grokking"*, in which a machine learning system undergoing continued training long after it achieves zero training error suddenly seems to acquire improved performance on test data (Power et al., 2022). Generally, this is interpreted as a sudden shift from memorization to generalization. Once again, grokking is hypothesized to arise because contemporary learning algorithms are biased to favor simpler solutions (Arpit et al., 2017; Liu et al., 2022; Valle-Pérez et al., 2018).

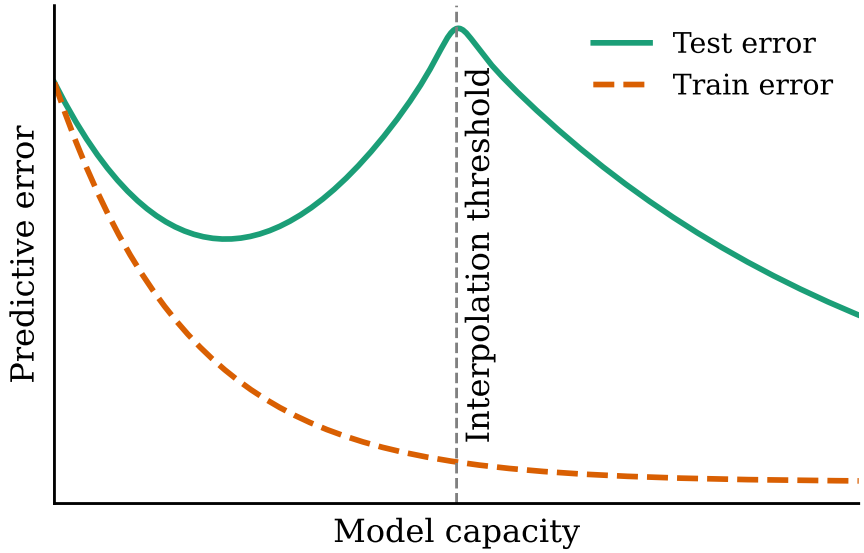
Another strand of evidence comes from attacks that extract verbatim training snippets from LLMs (Carlini et al., 2023). These studies demonstrate that, while models do indeed memorize *some* verbatim snippets, this is vanishingly rare relative to their overall behavior. Contrary to early theoretical expectations, it seems that pockets of memorization can and do coexist with broad generalization.

Thus, while debate continues (Anagnostidis et al., 2022; Recht et al., 2019), a general

[7] This can be understood through Vapnik–Chervonenkis (VC) dimension or Rademacher complexity. The *VC–dimension* of a model class is the size of the largest set of points it can shatter—that is, the largest dataset on which it can realize all possible labelings (Vapnik, 1998). Rademacher complexity measures the expected correlation between random noise and the model's outputs, thereby quantifying the model's ability to fit arbitrary labels (Bartlett and Mendelson, 2002).



(a) The bias–variance tradeoff: as deep learning system capacity increases from left to right, training error falls, but test error increases, indicating the system is overfitting due to memorizing more of the data.



(b) The double descent phenomenon: beyond the interpolation threshold, the test error continues to decrease, indicating the system is generalizing by learning real patterns in the training data.

Figure 2: Schematics showing the bias–variance tradeoff and the double descent behavior.

consensus has emerged. In current, successful deep learning models, only a small (but non-zero) fraction of model capacity is devoted to direct memorization, while many of the achievements arise due to generalization. At the very least, deep learning systems are able to identify and apply general patterns in the training data, an indication that these systems might possibly *understand* some of the regularities in their target systems. We will explore memorization more precisely in section 4.1.

3.2 Interpolation and Extrapolation

A second, closely related, proxy battle centers on whether these systems genuinely **extrapolate** or merely **interpolate**. Roughly speaking, a system interpolates when it generates successful, novel predictions from inputs within the envelope of its training data, whereas it extrapolates when it makes successful, novel predictions from inputs outside that envelope. A system that only memorizes the training data may have limited success in interpolating between those data points, but is unlikely to have *systematic* success outside the envelope of the training regime. A system that generalizes should have at least some interpolative success. It *may* succeed at extrapolation, *if* the learned generalizations extend beyond the training regime. The appeal of extrapolation as a diagnostic comes from the thought that a system which has acquired deeper understanding should sometimes succeed beyond the immediate envelope of its training data, whereas a system that has learned only local fits will often fail outside of the training regime.

A prevailing folk-theory holds that deep learning systems are generally good at interpolation but poor at extrapolation. After all, deep learning systems work by learning to fit a function to the data only in the specific domain of their training data. As such, deep learning systems can estimate intermediate values between known examples, but fail to pick up patterns that generalize outside of the training data. Marcus (2018) gives a simple example: he trained a three-layer model on the (seemingly trivial) identity function, $f(x) = x$ for a small set of even numbers. The model worked successfully on nearby even numbers but gave nonsensical results for odd numbers. Marcus takes these results to show that the model successfully interpolates to nearby even numbers, but fails at odd numbers, outside of the original training set.

Balestriero et al. (2021) challenge the conventional wisdom, defining interpolation as predictions whose input lies inside the convex hull of the training points, whereas extrapolation applies whenever the input falls outside that. They demonstrate that in high ($\gtrsim 100$) dimensional spaces, characteristic of almost all real-world deep learning problems, the probability that any new input will fall inside the convex hull becomes very small. Consequently, they argue that almost all practical high-dimensional deep learning predictions are extrapolative.

However, Chollet (2019) suggests that the relevant criterion of *interpolation* in many cases should be *interpolation on the latent manifold*. The **manifold hypothesis** posits that many kinds of natural data occupy lower-dimensional structured subspaces (latent manifolds) rather than filling the surrounding volume.[8] Deep learning networks learn to

[8]See Freeborn (2025b) for a philosophical overview of the manifold hypothesis.

unfold this manifold, turning it into a space in which local linear interpolation can suffice for generalization. As a result, most of the cases of extrapolation inside the convex hull can also be understood as interpolation on the manifold.

Both accounts are probably right: most deep learning predictions are both close to being interpolative on the manifold and extrapolative in the convex hull. It is likely that deep learning systems are poor extrapolators beyond the latent manifold of the data. They *can* learn general patterns in the data. However, often, deep learning systems fail to pick up on the right kinds of inductive regularities: those that they identify often fail to extend far beyond the envelope of the training data.

The two foregoing proxy disputes are best understood as partial attempts to operationalize constraints on genuine understanding. They arise because researchers often want to ask whether a model has learned real structure, but lack a shared and sufficiently precise account of what that would amount to. The memorization/generalization debate targets the thought that understanding must not depend essentially on sample-contingent idiosyncrasies of a particular training set. The interpolation/extrapolation debate targets the thought that understanding should involve some grip on structure that extends beyond local fit within the training regime. The model of systematic understanding developed in section 4 is intended to make these underlying concerns explicit and to show why these proxy debates capture something important, while also explaining why they are insufficient on their own.

4 A Model of Understanding

Understanding goes beyond mere capacity to predict. We shall think of it as a relation between one kind of system (the **agent**) and certain **properties** of a second system (the **target system**). This is the intuition I want to capture: to say that an agent *understands* some property of a system means that they have the capacity to make predictions about it that are not merely due to luck, and not due to rote memorization. Instead, they can systematically derive such predictions from an underlying model of the target system. Insofar as their understanding is good, then this model must track salient features of the target system, and the predictions should derive from the accurate tracking of these features.

This conception is intended to capture a single, reasonably coherent notion of understanding. Hopefully, the core intuition is familiar. A human child who can reliably classify previously unseen four-legged pets as either dogs or cats plausibly understands something about "dogness" and "catness": they have internalized regularities that support correct generalization, even if they cannot articulate them. Likewise, an athlete who can consistently catch a ball has, in effect, internalized a model of its typical trajectories, and understands something about the dynamics of this physical system (even if they lack training in physics). And a scientist who constructs an explicit model of a system and uses it to generate novel predictions and explanations likewise exhibits understanding of the system in question (even if some parts of the model are false).

Understanding in this sense comes in degrees, and an agent may understand some

properties of a target system better than others. Consider the case of planetary orbits as an example target system. These orbits have many properties; for instance their shape is not circular, but rather slightly anisotropic. In the early seventeenth century, Tycho Brahe's meticulous observations provided accurate positional data for Mars and Venus. These data supported reliable short-term prediction: Brahe could list where Mars or Venus had been, and predict where they would be by interpolation, but he lacked any principled grasp of the underlying patterns behind the numbers. So, he already possessed some understanding, but it was very limited. Johannes Kepler went a step further. By recognizing that the Sun sits at a focus of an ellipse, Kepler could derive the planetary data from a simple geometrical model. His capacity to forecast future positions was therefore no longer a matter of lucky curve-fitting but followed systematically from the structure of the model itself (Kepler, 1992; Thoren, 1990; Voelkel, 2001). I contend that Kepler had a greater *understanding* of the target system: he could predict the motion of the planets, deriving his predictions from an underlying elliptical model and three laws of motion.

Yet, Kepler's models could not explain all of the system's phenomena; the elliptical form of the orbits was treated as a basic geometrical fact, not itself explained. Newton achieved a deeper understanding by showing that elliptical orbits follow from an inverse-square force law. Yet, even Newton could not fully understand why the attraction took an inverse-square form. Gauss, in turn, developed a still deeper understanding using a further underlying model: a field emanating uniformly in all directions from a point source in three-dimensional space may naturally exhibit such an inverse-square dependence. So while Kepler had *some* understanding of some properties of the target system, Newton understood *more*, and Gauss understood even *more*. At each stage, understanding deepened by embedding previously accepted regularities within a more encompassing model. Of course, none of these models captured all features of the target system. They idealized away tidal effects, relativistic corrections, and many other phenomena. Models of a target system are generally highly idealized, providing only an approximate understanding of some subset of properties, under some conditions, based on an incomplete representation of the system.

This account of understanding is intended to capture some core insights of an ability-based picture, towards which much of the recent philosophical literature has converged (de Regt, 2017; Greco, 2014; Grimm, 2014,1; Railton, 1978; Salmon, 1984; Strevens, 2008; Woodward, 2003), especially the manipulationist or interventionist approaches, as well as the contention that genuine understanding must be free of environmental luck (Pritchard, 2010).[9] With that said, my aim is to explicate a deliberately limited and philosophically *thin* conception of understanding; for example, I will not consider causal mechanisms here. I make no claim that this is a general explication of the concept, nor that it is the best overall account. Rather, the purpose is to provide one useful starting point, to establish some common ground on which to assess and evaluate the kinds of understanding that deep learning system can, or cannot, exhibit.[10]

[9] However, see Kvanvig (2009) for an alternative view.

[10] To put it another way, I do not aim to establish a metaphysical thesis about "understanding". I

But I aim for an account of understanding that is suitable, at least in principle, for artificial intelligences. Therefore, I will avoid anthropocentric notions of understanding here, not because they are unhelpful, but they are, at least prima facie, poorly suited to machine learning systems. Thus my account will not aim to capture all of the key insights from explicitly psychological accounts of understanding (de Regt, 2017). After all, deep learning systems ostensibly lack precisely those psychological features.[11]

4.1 Memorization as a Limiting Case

At this point, it is worth explicating the term "memorization" more clearly, and to explain why it constitutes a limiting case that falls short of systematic understanding.

Following Dennett (1991a), a pattern in a body of data is **real** only if it admits a description that is shorter than brute-force enumeration, thereby supporting reliable prediction. The guiding contrast is between a lookup table, which guarantees accuracy by recording every idiosyncratic detail, and a compressed description that exploits stable regularities. Real patterns, in this sense, are precisely those that can be compressed while offering predictive power (Chaitin, 1975; Dennett, 1991a; Li and Vitányi, 2008).

Let T be a target system, and let p be a property of T that an agent system S is tasked with predicting. In standard learning settings, S is trained on a finite dataset $D = \{(x^i, y^i)\}_{i=1}^n$ generated from T under fixed bridge principles. At a first approximation, a memorizing system is one whose apparent success is essentially *data-dependent*. Its performance relies on idiosyncratic features of the particular sample D that happened to be encountered, rather than on extracting regularities that would support similar success across other datasets generated from the same target system.

This distinction can be made precise using the Minimum Description Length (MDL) framework. On an MDL approach, a learner extracts structure from D by selecting a hypothesis H that minimizes the total description length

$$L(D, H) = L(H) + L(D \mid H), \tag{5}$$

where $L(H)$ measures the complexity of the hypothesis and $L(D \mid H)$ measures the remaining description length of the data given that hypothesis (Grünwald, 2007; Rissanen, 1978). A hypothesis offers genuine compression only if it improves upon brute listing:

$$L(H) + L(D \mid H) < L(D). \tag{6}$$

Here "length" refers to the length of an optimal code (e.g. in bits) for representing an object (see Chaitin, 1975; Grünwald, 2007; Rissanen, 1978). A hypothesis with many free parameters, exceptions, or ad hoc clauses requires a longer description than a simple, structured hypothesis. Likewise, if a hypothesis captures genuine regularities in the data,

wish to propose a model of "understanding" as a foundation for the debate, and thereby to endow the reader with some understanding of this concept.

[11] With that said, the emphasis on an agent's capacity to manipulate the model echoes many features de Regt's contextual theory of scientific understanding. Likewise, this approach necessitates failing to capture some features of testimonial accounts (Hills, 2016).

fewer additional bits are needed to specify the remaining details of the dataset once the hypothesis is given. Thus MDL offers a formalization of the idea that learning consists in trading a modest increase in model complexity for a substantial reduction in the cost of describing the data.

Compression, however, is necessary but not sufficient for systematic understanding. What matters is not compression *simpliciter*, but compression that captures stable regularities of the target system across the relevant perturbation family, rather than merely exploiting statistical artifacts of a particular training sample. To identify a real pattern, in Dennett's sense, is to distinguish **signal** from **noise**: to retain those regularities that persist across datasets generated from the same target system, while discarding sample-contingent idiosyncrasies that do not support reliable prediction under appropriate changes. [12] A memorizing system fails to make this distinction. Because it incorporates noise specific to D into its representation, its success is fragile, not robust. When evaluated on a new dataset D' drawn from the same target system, where the underlying signal is unchanged but stochastic details may differ, performance typically degrades. Overfitting provides a familiar illustration of this phenomenon.

In this sense, memorizing agent systems are those that build a trivial model of the target system. Model complexity scales with the amount of data rather than with the complexity of the regularities governing p. Memorization is the limiting case in which predictive success is secured primarily by retaining training-specific information, rather than by internalizing a model that tracks target-stable structure. In information-theoretic terms, memorization is the limit of zero genuine compression: the model's effective description length scales with the size of the dataset rather than with the complexity of the regularities governing p. Such a system may perform well within a narrow regime of familiar cases, but it does not thereby attain systematic understanding of the target property.

None of this implies that memorization is useless. Lookup tables and memory-based strategies can be valuable for reliable prediction in constrained domains. The point is rather that such success is compatible with a lack of systematic understanding. Moreover, real systems often occupy intermediate positions. A single agent may combine pockets of memorization with real understanding. This hybrid picture will be central to the diagnosis of *fractured understanding* in later sections.

Of course, it is worth noting that compression itself admits of degrees and kinds. A system may achieve modest compression by encoding a few local regularities more efficiently than brute enumeration, while still falling far short of the deepest forms of compression, in which a small stock of reusable principles generates predictions across a wide range of phenomena. The distinction between these levels of compression will prove important when we consider the ideal of scientific understanding in section 4.7, and the characteristic ways in which deep learning systems fall short of it.

[12] This qualification is important because a system can achieve compression of the wrong kind. Shortcut learning, for example, compresses the training data by latching onto superficially predictive features, such as image texture or background context, that happen to be correlated with the target property in the training distribution but do not reflect the deeper structure of the target system. Such a system compresses, but it compresses noise rather than signal.

4.2 Systematic Understanding

We are now in a position to state the core idea more precisely. I propose the following characterization of **systematic understanding**.

> An **agent system** S **systematically understands** a property p of a **target system** T insofar as the following criteria are met.
>
> - S contains a subsystem $M \subset S$ that functions as an *adequate model* of some aspect of T.
> - The model M systematically tracks the property p of T without memorizing it.
> - There exist appropriate **bridge principles** connecting the terms of M to those of T.
> - The agent can use M to (approximately) derive certain properties of p of T.

Several of these terms require further explanation.

Structured Systems In practice, this notion of systematic understanding is fundamentally *structural*. What matters is that the internal model instantiated by the agent mirrors, preserves, or tracks relevant patterns, relations, or invariants in the target system, and that the agent can exploit this structure to generate predictions.

To make this more precise, I will treat both agent systems and target systems as structured systems. I leave open how best to represent such structures, whether in set-theoretic, algebraic, category-theoretic, or other terms. [13]

This is *not* intended as an ontological commitment, that real-world systems admit structural representations.[14] All I offer is a *model* of understanding, to give us a way to

[13] There is a parallel here with the free energy principle (FEP) (Friston, 2010). On the FEP picture, an *adaptive* system S is a persisting self-organizing (stochastic) dynamical system that maintains its organization over time, typically by remaining within a bounded range of states in a relatively stable environment. Such a system can be described as maintaining internal states (often interpreted as encoding approximate posterior beliefs) that parameterize a generative model of how its sensory states depend on latent variables in an external system T. In our terms, this generative model functions as an internal model $\mathcal{M}$ whose structure approximately mirrors the dependency relations linking (some variables of) T to the system's observations. Under a Markov-blanket partition (a set of *sensory* and *active* interface variables such that internal and external states are conditionally independent given the blanket), internal and external states are coupled only via those interface states (Friston, 2010,1). This partition plays a role akin to a family of bridge principles in our sense, since it fixes the interface variables through which S and T are coupled, and thereby the channel through which internal states can track external regularities. *Active inference* is the associated account of perception and action: roughly, S updates its internal states so that its sensory inputs become less surprising under its generative model (where "surprise" means the negative log probability of current sensations), and it selects actions expected to yield unsurprising, preferred, and uncertainty-reducing future sensory inputs (Friston et al., 2017; Parr et al., 2022). On this construal, paradigmatic active-inference cases are naturally interpreted as instances of *structural* understanding (though not automatically as *reductive* understanding).

[14] Some (see Ladyman and Ross 2007) do contend that the world really does consist of structured systems, in which case, one could choose interpret these claims more literally.

study the concept more precisely. Representing systems as structured entities is intended as a convenient modeling choice, so that we can talk about how an agent's model system succeeds or fails in tracking properties of the target system. Nonetheless, it is standard to describe real-world systems using structured systems (e.g. scientific theories). In the same spirit, I will treat both agent systems and target systems *as if* they were structured systems, so that relations such as tracking, derivation, and misrepresentation can be meaningfully discussed.

Different kinds of tracking relationships could bestow different kinds of systematic understanding. In this paper, I will focus on **structural understanding**. However, it is also worth considering other kinds; I suggest one other, **reductive understanding**. Loosely speaking, structural understanding requires that the model represents the right patterns and relations in the target system, whereas reductive understanding requires that the model represents a higher level approximation of the target system.[15] I will discuss these specific kinds of systematic understanding in sections 4.3 and 4.4 below.

Agent Systems The notion of an agent system is correspondingly broad. The agent may be a human, an artificial intelligence, a collective organization, an idealized Bayesian learner, or something else. In some cases, the agent system may even be embedded within the target system that it seeks to understand. What matters is not the boundaries of the agent, but whether there exists within it a subsystem capable of internalizing an adequate model of the relevant properties of the target.

This allows for cases in which understanding is distributed across multiple components. For instance, in Searle's *Chinese Room* thought experiment (Searle, 1980), an English speaker follows a gigantic rule-book that pairs Chinese inputs with Chinese outputs. We may consider the target system to be the *Chinese Language*. Searle contends that the individual in the room does not understand the Chinese language. A common response is to argue that the room-person system, taken as a whole nonetheless does understand the language. According to this response, the *agent*, in our language, would be the room-person system as a whole (Churchland and Churchland, 1990; Harnad, 1989; Lycan, 1990).[16]

[15] We could also consider stronger kinds of tracking relation, which I will not discuss here for the sake of brevity. An obvious, and potentially highly illuminating extension would be a kind of **causal understanding**, with a requirement of a causal tracking relation. Plausibly, a causal tracking relation would require invariance under interventions: for appropriate variables X and target property p, the model tracks p causally only if it represents (perhaps implicitly) interventional dependencies of the form $P(p \mid do(X{=}x))$, not merely observational correlations $P(p \mid X{=}x)$, as captured by Pearl's interventionist framework and do-calculus (Pearl, 2018).

[16] To take another example, perhaps no individual within the intelligence services of Oceania understands the mechanisms behind the war with Eastasia (Orwell, 1949). Even so, the intelligence services as a whole may possess such an understanding, even though the requisite model is distributed across various files, algorithms and administrative clerks. As such we might think of the intelligence services as a whole as the *agent*.

Models of the System Now, models are often understood not as merely structures, but as intrinsically semantic, interpretive objects.[17] For instance, perhaps a formal structure within the agent only really counts as a model relative to additional interpretive scaffolding, including semantic assignments, idealizations, and background assumptions that enable the agent to use it representationally. For this reason, I distinguish between a bare structure M and a model $\mathcal{M}$, where the latter includes whatever additional resources are required for the structure to play a representational role. However, nothing in my account, turns on this point: readers who prefer a purely syntactic conception of models may simply identify M with $\mathcal{M}$ as needed.

By an *adequate* model, I mean adequate for the purposes for which the agent will use the model. As Parker (2020) emphasizes, what matters is not solely how accurately a model mirrors reality in general, but whether it is fit for the specific inferential or practical tasks for which it is employed. A model may idealize, distort, or even misrepresent aspects of a system, and yet remain adequate-for-purpose if it supports the kinds of epistemic or practical interventions the agent requires. However, at the very least, we require that an **adequate model** must allow the agent to make sufficiently accurate (approximate) predictions of property p, in order to endow our agent with understanding of p. Furthermore, such predictions must not be arbitrary, but derive from the fact that the model represents and tracks at least some features of the target system. This requires suitable bridge principles.

Bridge Principles To construct a genuinely non-anthropocentric notion of systematic understanding, we must not demand that bridge principles rely on linguistic representations or something directly analogous to human mental states. What matters is not that the agent *interprets* its model in a reflective or semantic sense, but that there exists a stable interface through which the internal structure of the model can reliably track, and be used to derive claims about, properties of the target system. When such mappings are in place, the model's internal variables can be said to be "about" the target system in a minimal but substantive sense: changes in the target systematically covary with systematic changes in the model, and the agent might exploit this relationship to generate predictions or guide action.

I do not intend to address the wider problem of intentionality, namely, what it is for a mental state, representation, or model component to be *about* some external feature of the world. Instead, I will proceed with a modest naturalistic assumption: whatever intentionality consists in, it can be realized by physical systems (including engineered ones) in virtue of the roles their internal states play in reliable prediction, explanation, and action. Several possible accounts could suffice. First, on a broadly *informational* or *causal–covariational* view, internal states have content insofar as they carry information about external conditions: they stand in stable patterns of counterfactual dependence on features of the environment, and the agent can exploit those dependencies to guide

[17] See da Costa and French (1990); Suppes (1960); van Fraassen (1980) for semantic approaches for scientific models, and Frigg (2022); Frigg and Nguyen (2017); Morgan and Morrison (1999) for additional discussion of the representative capacities that we might need to bestow upon models.

successful prediction or action (Dretske, 1981,8). Second, on a *teleofunctional* or etiological view, the relevant informational links are not mere accidental correlations but are stabilized by proper function, where correctness and error are fixed by the role that the state was selected, learned, or trained to perform (Millikan, 1984; Neander, 1995). Third, on Dennett's *real-patterns* perspective, intentional characterizations are warranted at a coarser explanatory grain when they pick out objective patterns that support compression and reliable prediction across a range of counterfactual conditions (Dennett, 1991b). On this view, treating some internal configuration as a representation is justified when doing so materially improves our explanatory and predictive grip.

These assumptions are intentionally modest. They do not require linguistic meaning, conscious awareness, or a special metaphysics of reference. They are, however, strong enough for what follows: they license the claim that, given suitable coupling and stability conditions, components of an agent's internal model can be *about* features of a target system in a way that supports tracking, derivation, and the possibility of systematic error. This is exactly the sense of aboutness presupposed by the bridge principles.

Hence, by **bridge principles**, I mean any stable mappings, conventions, or interface relations that connect elements of the model $\mathcal{M}$ to elements of the target system T. As such, they are key to the semantic or interpretive component of understanding, fixing how the model is to be taken as being *about* the target rather than about an arbitrary dataset or abstract mathematical structure.[18] These mappings determine which variables, states, or structures in the model correspond to which properties, quantities, or configurations of the target system, and thereby underwrite the model's representational role. These need not take the form of explicit laws, identities, or semantic interpretations. For instance, in nonhuman agent systems, they may include procedures, tokenization or featurization schemes, conventions, evaluation protocols, or other engineering choices which serve to couple internal features of the model, $\mathcal{M}$, to features of the target system, T. In the context of machine learning systems bridge principles typically include the preprocessing steps that map target system states to data inputs, as well as the decoding rules that map internal weights or output scores to predictions, and the fixed routines by which those predictions are interpreted as claims about the target system. I will offer a more explicit unpacking of bridge principles in the deep learning case in section 5.

This raises a question about the *boundaries* of the agent system. Bridge principles are not optional interpretive commentary supplied by an external observer, but an essential part of the agent system itself. When this framework is applied to deep learning systems (section 5), the natural locus of understanding is therefore often not the trained network in isolation, but the *deployed computational pipeline* that couples the network to the target system and renders its internal states and outputs as determinate claims about p. As we will see, the deep learning agent system typically includes not only the neural network, but also rules of preprocessing, tokenization, sensor transduction, decoding, and other operations.

[18]Recent work on large language models suggests that representational aboutness may often be grounded in relatively stable regions of high-dimensional activation space, rather than in discrete symbolic vehicles (Ball et al., 2025; Shea, 2007).

As such, I will treat bridge principles as part of the agent system only when they are (i) *fixed at deployment*, rather than selected post hoc to rationalize success, (ii) *executed automatically in normal operation*, rather than supplied ad hoc by an external evaluator, and (iii) *available as part of the system's competence*, in the sense that they participate in its closed-loop capacity to map target situations to task-relevant outputs. A further, stronger requirement is *functional integration*: the bridging routines must be stably coupled to the learned component so that together they constitute a single, reliable end-to-end competence, rather than a mere attachment whose contribution is more appropriately credited to an external user. Without such constraints, claims about understanding would risk trivialization by tacitly importing additional and essential structure into the system.

Derivation The derivation condition is intended to exclude cases in which predictive success is merely read off by an external evaluator, rather than generated by a stable competence of the agent system itself. Accordingly, when I say that S can use M to (approximately) derive properties of p, I mean that there exists a *fixed, agent-available procedure* by which S maps situations of the target system (as presented under the bridge principles) into outputs that constitute claims about p. For machine learning systems, this procedure may include the trained forward pass together with the *fixed* encoding, decoding, and evaluation conventions that are part of the deployed pipeline. By contrast, *post hoc* scientific analyses performed only by the investigator (for example, interpretability probes trained after the fact, or external measurements that the deployed system does not itself run) may provide evidence that M carries certain information, but they do not, by themselves, constitute the system's derivational competence unless those analyses are genuinely integrated into the agent system as deployed.

Non-Memorization Following section 4.1, we can operationalize non-memorization in several ways. One option is a **robustness test** of whether a system is merely memorizing, closely related to the tests performed by machine learning practitioners (Geirhos et al., 2020; Vapnik, 1998). Fix a target system T, a property of interest p, and bridge principles that determine how data are generated and how predictions are evaluated. Consider a family of *structure-preserving perturbations* Π on the configurations of the target system Ω_T, where each $\pi \in \Pi$ is designed to vary features that are *irrelevant* to p while preserving the salient structure that determines p.[19] Let $\widehat{p}(S, \omega)$ denote the system's predicted value for p on a target situation $\omega \in \Omega_T$. Then a necessary condition for non-memorization is some degree of *stability*:

$$\widehat{p}(S, \omega) \approx \widehat{p}(S, \pi(\omega)) \qquad \text{for all } \omega \text{ in the intended domain and for all } \pi \in \Pi. \tag{7}$$

[19] I contend that what counts as "structure-preserving" is necessarily interest-relative and must be specified by the bridge principles. In image classification, for example, Π might include changes in background or lighting; in a physical prediction task it might include perturbations that preserve the relevant conserved quantities; in a symbolic task it might include relabellings or symmetries that preserve the underlying algebraic relation.

Intuitively, a system that merely memorizes training-specific idiosyncrasies will tend to be fragile under such perturbations, because the idiosyncrasies do not persist across Π-variations. By contrast, a system that tracks a target-stable regularity can remain reliable across the perturbation family. In other words, a successful non-memorizing system should keep succeeding at predicting property p when we *change features that should not matter to* p. That is, if we intervene on the target situation or its presentation (for example, by altering irrelevant background factors, or by using a different but equivalent encoding or measurement setup) while holding fixed the underlying structure that makes p true of the target, then the end-to-end pipeline should still make approximately correct predictions about p.

We might supplement this robustness test with a **compression proxy**: holding the bridge principles fixed, a non-memorizing strategy is one whose effective description length (as captured by a suitable model class) does not scale linearly with the number of distinct training examples in the way a lookup strategy does. After all, robust structure-tracking typically permits a more compact description than sample-contingent encoding.

The robustness test and the compression proxy are complementary operationalizations of the same underlying requirement. A system that achieves genuine compression of target-stable regularities will tend to be robust under structure-preserving perturbations, because the compressed representation discards sample-contingent idiosyncrasies that do not persist across Π-variations. Conversely, a system that is robust across the perturbation family Π has likely achieved some compression, since retaining all training-specific details would leave it fragile under perturbation. Together, these conditions make precise one sense in which understanding is a matter of degree: the greater the compression of target-stable structure, and the wider the family of perturbations across which predictions remain stable, the greater the degree of systematic understanding. On this picture, memorization and full systematic understanding are the two poles of a continuum, and most real systems occupy intermediate positions.

4.3 Structural Understanding

Structural understanding is one kind of systematic understanding, drawing from the contemporary approach to scientific structuralism (see Bokulich and Bokulich, 2011; Dewar, 2022). We specify the tracking relation as follows:

- there exists a structure-preserving mapping between relevant parts of $\mathcal{M}$ and T, such that corresponding elements track the property p;

The structure-preserving relation need not be an isomorphism. In most cases it will be a homomorphism or other partial correspondence that preserves only those relations relevant to the property of interest.

As a simple example, let the agent system S be a physics student equipped with a pen and paper and basic mathematical tools. Let the target system T be a sample of unstable, decaying Barium-137m nuclei. Let the property of interest be the number of

nuclei as a function of time,

$$p \;:=\; \big\{\, N_T(t_T) \in \mathbb{R}_{\geq 0} \;\big|\; t_T \in \mathbb{R}_{\geq 0} \big\}, \tag{8}$$

where $N_T(t_T)$ denotes the number of undecayed nuclei at laboratory time t_T. To represent this system, the student employs an exponential decay model,

$$\frac{dN_M}{dt_M} \;=\; -\lambda_M \, N_M(t_M), \tag{9}$$

where N_M and t_M are model variables and λ_M is a model parameter. Bridge principles identify t_M with laboratory time t_T, interpret $N_M(t)$ as the expected number of undecayed nuclei at time t, and fix $\lambda_M = \lambda_T$ by an independent half-life measurement or by reference data in nuclear-data tables.

This defines a model structure

$$M = \big\langle \mathbb{R}_{\geq 0},\, \tfrac{d}{dt_M},\, \lambda_M \big\rangle, \tag{10}$$

interpreted with the usual rules of calculus and dimensional assignments, together with a structure-preserving mapping

$$h :\; N_T(\cdot) \;\longmapsto\; N_M(\cdot), \tag{11}$$

where h maps target trajectories to model trajectories under the identification $t_M = t_T$ and the interpretation of N_M as an expectation value.
Because h preserves the additive structure of the real numbers and commutes with the derivative operator under this identification, it is a homomorphism between the relevant dynamical substructure of T (its mean-field decay law) and that of $\mathcal{M}$. In this sense, the same exponential pattern is instantiated in both the model and the target system.

The model is certainly idealized: the real decays are a probabilistic process. But if the model is successful, then our student can not only approximately *predict* the number of Barium-137m nuclei over time. They also **understand** something about this property, that it arises from a decay process in which the rate of decay is proportional to the number of remaining nuclei. A deeper understanding might in turn derive the equations of this decay process from the nuclear structure of the particles.

4.4 Reductive Understanding

Let us consider an alternative kind of systematic understanding, **reductive understanding**. This draws from the Nagelian-Schaffnerian tradition (see Dizadji-Bahmani et al., 2010; Nagel, 1961; Schaffner, 1967). While structural understanding depends upon a homomorphism between model-structure and target-structure, reductive understanding is achieved when the agent possesses a model that *derives* the relevant property of the target system from a more fundamental theoretical basis.

- There is a reduction relation from part of the target system T and the model $\mathcal{M}$, such that the properties of $\mathcal{M}$ can be derived from T, and we can approximately predict p from $\mathcal{M}$.

For present purposes, we do not need to specify this more fully. One possibility, following the generalized Nagel-Schaffner model, proposes three criteria for a reduction relation between M and T: strong analogy, derivability, and connectability.

Perhaps both structural and reductive understanding may both be useful and distinct species of systematic understanding. Structural understanding is based on the requirement that the agent's model must track certain structural features of the target system. Reductive understanding is based on the requirement that the agent's model must be a reduction of the target system. It may be that the requirements for reductive understanding are as *least as strong* as the requirements for structural understanding. That is, if the target system can be reduced to the model, then there must also be some structural homomorphism between the target system and the model.[20] But it seems possible that we could sometimes have structural models without there being any **straightforward** reduction[21]. These could include empirical laws that successfully track structural regularities in the target system.[22]. Thus, I leave it open as to whether the requirements for reductive understanding are strictly stronger than the requirements for structural understanding.

4.5 Causal Understanding

Finally, let us consider one further kind of systematic understanding, **causal understanding**, in which the relevant tracking relation preserves not merely structural patterns but the target system's *causal* dependency structure. As such, the model's variables correspond to features of the target system in a way that supports stable predictions under causal interventions, not merely under passive observation. Pearl's interventionist framework and do-calculus provide one natural way to make this precise (Pearl, 2018). In the Pearlian approach, the model tracks p causally only if it represents interventional dependencies of the form $P(p \mid do(X{=}x))$, rather than merely observational correlations $P(p \mid X{=}x)$ (Pearl, 2018). Such an account might offer a richer conception of understanding than the structural variety developed here, and would connect naturally to recent work on causal discovery and causal representation learning (Peters et al., 2017; Schölkopf et al., 2021). I will not develop causal understanding further in this paper, but note it as a promising direction for extending the framework.

4.6 Tacit and Symbolic Understanding

We need to consider another distinction between kinds of systematic understanding. The examples I have discussed above were **symbolic**, expressed (somewhat) explicitly

[20] See Wallace (2022) for a similar idea.

[21] I use the term "*straightforward*" reduction judiciously. Perhaps, if the requirements for reductive understanding are sufficiently loose, any structural understanding could entail some kind of reductive understanding too.

[22] Possible examples, all contentious, include cases of Zipf's law (Ferrer-i Cancho and Solé, 2003; Mitzenmacher, 2003; Piantadosi, 2014; Zipf, 1949), the Tully-Fisher relations relating galaxy luminosities and rotational velocities (Said, 2024) or effective chiral Lagrangians in low-energy Quantum Chromodynamics (Georgi, 1993).

with mathematical terminology. But I suspect that most real understanding in human beings is not of this kind, but is rather more **tacit**, without the agent knowing how to express this in explicit symbolic form[23]. This symbolic/tacit dichotomy does not perfectly map onto the structural/reductive distinction. However, *generally speaking*, reductive understanding might be relatively more likely to be symbolic compared to structural understanding.

This symbolic–tacit dichotomy echoes Polanyi's distinction between tacit and explicit knowledge (Polanyi, 1958)[24]. Polanyi's *tacit* component of personal knowledge encompasses precisely those sub-propositional skills and pattern recognitions. Let us apply this distinction not just to *knowledge*, but also structural *understanding*.

Consider a physicist catching a ball.[25] Even the most committed theorist is unlikely to explicitly model the ball's trajectory mathematically, for example by using Newton's equations of motion. Yet, they can adapt to subtleties in the ball's motion in a fraction of a second. The physicist's brain somewhere stores an implicit *mental model* of the ball's trajectory, even if it is not an explicit one that they can symbolically describe. This mental model has clearly captured certain regularities about how the ball moves through space. And so the physicist has at least *some* tacit, structural understanding of the ball's motion.

Alternatively, consider the "waggle-dance" communication of honey-bees. When a successful forager returns, she traces a figure-of-eight whose angle and duration encode the displacement vector to a nectar source. Recruits observe multiple dances, effectively average the encoded vectors, and then navigate to the advertised location. At the level of the colony, this implements vector integration in a two-dimensional metric space. Yet no single bee possesses the trigonometry capacity for such a computation. The spatial model is distributed across many individuals and implemented in sensorimotor routines rather than in explicit, symbolic form (Seeley, 1995; von Frisch, 1967).

Finally, consider a chess grandmaster. Undoubtedly, they have a high-level understanding of the chess game, its rules and strategies. Yet they cannot rely on an explicit, symbolic traversal of an exponential game tree, but rather chess grandmasters excel due to superior recognition of structural patterns in the game de Groot (1965) and Gobet and Lane (2005).[26] Their skill reflects the internalization of a highly sophisticated model of the game, one that guides judgment and action without being fully expressible in symbolic form.

[23]Indeed, the ability to mentally, qualitatively manipulate a model without having to carry out exact mathematical calculations is central to the account of understanding developed by (de Regt, 2017). I contend that at least some kinds of understanding are non-symbolic in this sense.

[24]Likewise, Ryle's distinction between *knowing-that* and *knowing-how* (Ryle, 1949) marks the difference between possessing a catalog of explicit propositions (symbolic understanding) and possessing the trained disposition to act appropriately in concrete situations (implicit understanding).

[25]Observational evidence suggests that some, though not all, physicists are capable of this.

[26]Indeed, even Chess grandmasters typically only think a few moves ahead explicitly, during the midgame. This can increase during early and late game phases.

4.7 The Ideal of Scientific Understanding

We might identify an epistemically privileged **Ideal of Scientific Understanding**, which has often historically been favored by both philosophers and scientists (although I do not claim that all or most scientific models meet this ideal). According to this ideal, understandings should be both **reductive**, **symbolic** and **unifying**. There are good reasons for this. Symbolic representations are communicable, portable artifacts that can be inspected, critiqued, built upon and improved by other investigators, whereas an individual's tacit grasp of a phenomenon is largely incommunicable. They can also be very directly manipulated through logical reasoning. Reductive models may offer especially suitable candidates for partial realist interpretations. In particular they may facilitate selective realism about the essential parts of the model (likely to survive under intertheoretic reduction) (Freeborn, 2025a; Psillos, 1999; Worrall, 1989).

Moreover reductive models may offer especially information-efficient, compressed representations of systems, in which a comparatively small stock of principles and derivational patterns can generate descriptions of a wide range of phenomena (Freeborn, 2025a; Friedman, 1974; Grünwald, 2007; Kitcher, 1981; Nagel, 1961). Such compression is sometimes considered an indicator that the model may have latched on to genuine structure in the world.

The unification ideal connects to the compression framework already developed in section 4.1.On Kitcher's view, science advances understanding by deriving descriptions of many phenomena from the same patterns of reasoning, used again and again, thereby reducing the number of independent facts we must accept as brute. A unifying explanation does not merely represent a phenomenon symbolically, nor merely derive it from a more fundamental basis; it also brings many apparently distinct cases under a comparatively small stock of reusable principles, derivational patterns, or inferential templates. For instance, Newton's laws unify the motion of earthly cannonballs, and distant planets under a single framework. A maximally fragmented model, by contrast, may predict each phenomenon correctly without any of its components lending support to the others. Building on this Votsis (2015) defines a unified hypothesis as one whose parts are *confirmationally connected*, so that evidence for one part lends support to the others. A mere conjunction of independent claims, each supported only by its own datum, achieves no such connection, and is instead "monstrous". At least ideally, science offers unified theories, rather than monstrous patchwork models whose separate responses to different inputs are confirmationally disconnected.

The ideal of scientific understanding is plausibly strongest where these features coincide: where a representation is symbolic, reductive, and unifying. Nonetheless, it may be that *much* (perhaps *most*) of our actual understanding of real-world phenomena is most naturally thought of as tacit, structural and non-straightforwardly reductive. I suspect that, whatever implicit manipulations a physicist performs when they model a ball flying towards them, cannot be directly related to Newtonian mechanics through a reduction relation. Rather, their brain has latched onto, and modeled a mosaic of vague and overlapping structural regularities. Despite the advantages of symbolic, reductive and unified understanding, we should admit tacit, structural understanding as a genuine

form of understanding.[27]

In section 7, I will argue that contemporary deep learning systems often appear to fall short of this ideal, not necessarily because they fail to track any real regularities, but because the regularities they track are not always organized into a small set of globally coherent and reusable principles. The kinds of understanding that deep learning systems offer are typically symbolic (but using an unnatural symbolic representation), not clearly reductive, and not very unifying.

5 Systematic Understanding in Deep Learning Systems

We are now in a position to assess whether, and in what sense, deep learning systems could ever satisfy the criteria for systematic understanding. For succinctness, we will focus on structural forms of understanding and on feed-forward ReLU neural network structures.

At first glance, it might seem that a neural network is precisely the type of system that is ill-suited for systematic understanding. After all, we have seen (section 2), that they can be understood simply as curve fitters or **spline interpolators**. The system learns merely by systematically tweaking its vast array of parameters until the function approximately fits the data from the target system. But let us keep an open mind for now. Insofar as this learned function approximates real regularities in the target system, then the deep learning system may have some systematic understanding of some properties in the target system. For example, if the learned function has some homomorphism to some features of the target system, then it may have a **structural understanding**; if it can reduce to a real function in the target system, then it may have some **reductive understanding**, assuming that the other criteria outlined in section 4.2 are satisfied.

Let the target system T be characterized, relative to some property of interest p, by a structured domain

$$\langle X_T, R_T \rangle, \tag{12}$$

where X_T is a space of relevant states, configurations, or conditions of the target system, and R_T is a family of relations on X_T that determine how variations in those states bear on p.[28] These relations may encode features like geometrical adjacency, algebraic composition, temporal ordering, or invariants that are salient for predicting or deriving the property in question.

[27] True understanding should entail greater generalizability outside of the training data, but this is always limited. The physicist's heuristic model will succeed only in a limited domain. They may manage to catch a ball thrown slightly slower or faster, or at a different angle. But they will probably fail if asked to catch a ball in a different gravitational regime, such as on the moon. Newton's Laws generalize better, although of course they too would fail outside of a more general domain, for instance if the ball approaches the speed of light.

[28] In typical supervised learning setups, p is operationalized as an input–output relation (or conditional distribution) that generates labeled data. Nothing in what follows requires that p_T be metaphysically fundamental; it may itself be an idealized or interest-relative property.

We will need to treat S as the full computational pipeline, which couples the target system to inputs and interprets outputs as claims about T. We can schematize this as

$$\Omega_T \xrightarrow{\iota} \mathcal{X} \xrightarrow{f_\theta} \mathcal{Y} \xrightarrow{\delta} \widehat{\mathcal{Y}}, \tag{13}$$

Here Ω_T is the space of *target system situations*, for example configurations of a physical system, sequences of moves in a game, or ordered lists of integers in a mathematical task. The map $\iota : \Omega_T \to \mathcal{X}$ is an *encoding or measurement interface*, which converts concrete target states into inputs the model can process. This includes sensor transduction (e.g. converting light into pixel values), featurization (e.g. extracting numerical descriptors from raw data), and tokenization (e.g. mapping words, board positions, or integers to discrete symbols). So, encoding is the point at which the target system is brought into contact with the learning architecture. The map $\delta : \mathcal{Y} \to \widehat{\mathcal{Y}}$ is a *decoding or decision rule* that turns these raw outputs into interpreted predictions or actions. Examples include selecting the highest-scoring class (argmax classification), sampling from a probability distribution, extracting a geometric object from a learned scalar field, or issuing a control signal. The space $\widehat{\mathcal{Y}}$ consists of outputs as they are evaluated and acted upon.

The learned function f_θ is specified by a neural network architecture and learned parameters, and captures all of the agent system's *learned* information about T (although of course some information will also be hardwired through the architecture or other inductive priors). Its outputs $\mathcal{Y}$ are typically uninterpreted numerical objects, such as vectors of logits, probability scores, or real-valued fields. The *bridge principles*, including ι and δ, allow subsets of the input space $\mathcal{X}$ and output space $\mathcal{Y}$ to be interpreted as representing elements of X_T, and thus allow f_θ to be *about* the target system T, rather than as merely implementing an abstract transformation between datasets.[29]

In order to specify the syntactic component of the model, M, it is helpful to distinguish between the neural network's *internal representational map* and a *readout*. Concretely, we may factor the learned function $f_\theta : \mathcal{X} \to \mathcal{Y}$ as

$$f_\theta = g_\theta \circ r_\theta, \tag{14}$$

where $r_\theta : \mathcal{X} \to \mathcal{Z}$ maps inputs into an internal *latent space* $\mathcal{Z}$, and $g_\theta : \mathcal{Z} \to \mathcal{Y}$ maps those internal representations to the system's raw outputs. The latent space $\mathcal{Z}$ is specified by the neural network's internal activation patterns, such as the activations of intermediate layers. These internal states are not directly observable at the level of inputs or outputs, but here they play an indispensable explanatory role.[30] As such, we can write the whole process as,

[29] Recent work in philosophy of science has argued that, in some scientific contexts, machine-learned models can legitimately play theory-like roles, being both inferred from data and used to generate and test data-relevant predictions (Norelli et al., ming).

[30] This distinction is useful for two reasons. First, in most deep learning systems, the structure that supports generalization does not reside in the raw input space $\mathcal{X}$ itself, but in how inputs are mapped into and organized within the latent space $\mathcal{Z}$. Very different inputs may be mapped to nearby or systematically related points in $\mathcal{Z}$ if they are treated by the system as equivalent or similar for the purposes of the task. Second, the readout g_θ typically depends only on limited aspects of this internal organization, extracting task-relevant information while ignoring other variation present in the latent space.

$$\Omega_T \xrightarrow{\iota} \mathcal{X} \xrightarrow{r_\theta} \mathcal{Z} \xrightarrow{g_\theta} \mathcal{Y} \xrightarrow{\delta} \widehat{\mathcal{Y}}, \tag{15}$$

Let $X_M \subseteq \mathcal{Z}$ be the subset of latent states that the system actually occupies across its domain of application, and let R_M be the family of relations on X_M induced by the geometry and organization of $\mathcal{Z}$ together with the learned dynamics of the network. As we have already seen, in the case of feed-forward neural networks with ReLU activations, the learned function partitions X_M into a finite collection of convex polytopal regions $\{\mathcal{P}_k\}$, within each of which f_θ is affine. The internal model component $M \subset S$ can therefore be treated as a structured object,

$$\langle X_M, R_M \rangle, \tag{16}$$

where R_M includes relations such as adjacency between regions, linear dependence relations within regions, and continuity constraints across region boundaries. In this sense, the model instantiated by the network is a continuous, piecewise-linear structure, often described as a multivariate spline. On its own, M is merely a piece of internal machinery. However, when coupled with the encoding and decoding interfaces ι and δ, and with background task conventions that fix how internal states correspond to target-system properties, we can view $\mathcal{M}$ as a model of the target system T. Systematic understanding, on our account, requires that $\mathcal{M}$ stand in an appropriate relation to the relevant structure $\langle X_T, R_T \rangle$. For example, in the case of structural understanding, this should be a structure-preserving relation; in the case of reductive understanding it should be a reduction relation.

Seen in this light, spline interpolation is not inherently inimical to understanding. In principle, the models we have just described can track real patterns in the target system's data. Indeed, neural networks are extremely well suited for approximating a wide variety of functions. According to the **Universal Approximation Theorem**, a feed-forward neural network with a single hidden layer containing a finite number of neurons and a non-polynomial activation function σ can approximate any continuous function on a compact subset of $\mathbb{R}^d$, to arbitrary precision, given sufficiently many parameters (Cybenko, 1989). Real neural networks are finite, but still show impressive expressive power. Deep learning systems with many such hidden layers increase the expressive capacity by composing many such approximations, yielding highly flexible families of piecewise-defined functions.

Yet, this raises an obvious worry. Most real-world systems are probably not governed by piecewise affine functions. It seems likely that these learning systems could learn to approximate the functions that describe such systems, at least in the target data, without approximating the true underlying regularities of the system. In that case, the learned model may succeed at prediction while remaining epistemically shallow. A memorizing system would be the extreme case of this. It may fit the training data by allocating idiosyncratic representational states (or, in ReLU terms, idiosyncratic affine regions) to specific inputs in the training data, while failing to preserve the relevant relations R_T under systematic variation.

Yet this worry is not decisive. First, there is no requirement that a successful representational map must mirror the target's underlying dynamics in its own native mathematical idiom. What matters for systematic understanding is whether some internal structure in the agent systematically approximately tracks a structure in the target. Crucially, a piecewise affine function can, in principle, encode and preserve non-piecewise regularities via a change of variables, or by approximating a smooth dependence closely enough on the relevant domain that the salient invariances are preserved. The key question, then, is not whether the learned function is literally piecewise affine, but whether its internal organization implements a stable, compressive, and generalizable dependence on the target property of interest.

Second, the depth of the concern depends on what we take approximating the true regularities to amount to. In many scientific contexts, we do routinely accept models whose functional form is non-reductive, piecemeal, or otherwise mathematically alien to the underlying microdynamics, so long as they capture the right dependence structure at the level of description that matters. Likewise, a network can be epistemically shallow if its fit is achieved by a patchwork of idiosyncratic regions that fail to extend beyond the training distribution, but it can also be epistemically deep if its learned representation induces a small set of robust relations that continue to govern its behavior under systematic variation. From the above account, we cannot rule out the possibility of deep learning systems exhibiting genuine structural understanding, even of highly complex target systems.

To get a sense of whether and when this kind of systematic understanding can arise in practice, we will next look at some simple examples. In each case, we will ask what a deep learning model does, and does not, learn about some salient properties of a target system. Each example is meant to be simple enough to be clearly interpretable. In each case, a machine learning agent system will learn to approximate a function through a spline-like interpolation. We will analyze to what extent this kind of curve-fitting endows our agent system with systematic understanding of the target system.

6 Illustrative Examples

6.1 The Genus of a Torus

Our first example demonstrates that structural understanding of a global property can emerge from purely local supervision, without any explicit encoding of the target property. We consider a relatively trivial case: a neural network trained to learn an implicit representation of a toroidal surface embedded in three-dimensional space. We shall see that, by training the neural network on individual data points from the torus surface, the network gains a "structural understanding" of the genus-one topology of the torus.

The target system T is a continuous scalar field on $\mathbb{R}^3$, defined by the function,

$$F(x, y, z) = \left(R - \sqrt{x^2 + y^2}\right)^2 + z^2 - r^2. \tag{17}$$

Let $D = [-3.5, 3.5]^3 \subset \mathbb{R}^3$. We define the target surface $\Sigma := \{x \in D : F(x) = 0\}$.

This is a toroidal surface of major radius $R = 2.0$ and minor radius $r = 0.7$. The training data are obtained by uniformly sampling data from D and assigning each point its scalar value under $F(x, y, z)$, i.e. $(x^i, y^i, z^i, F^i)_{i=1}^n$.

The agent system consists of a trained three-layer feed-forward network, together with a fixed post-processing pipeline that extracts an isosurface from the learned scalar field and computes topological invariants of that surface. The neural network is trained to minimize the mean squared error between the predicted and true scalar field values.[31]

This architecture defines a function $f_\theta : \mathbb{R}^3 \to \mathbb{R}$ parameterized by weights θ. The model $\mathcal{M}$ consists of this learned function, together with the fixed reconstruction and measurement procedures that extract a surface from f_θ and compute its topological properties. The system's goal is to learn an approximation of the function F, from which a learned surface, $\hat{\Sigma} := \{x \in D : f_\theta(x) = 0\}$ can be extracted. The property of interest, p, is the *genus* of the target surface i.e. the property that the surface $F = 0$ is a connected, orientable surface of genus one. That is, the torus forms a single, continuous surface, with a consistent notion of inside and outside, and exactly one hole running through it (see figure 3).[32]

The bridge principles are straightforward in this setting. Inputs in $\mathcal{X} = \mathbb{R}^3$ are interpreted as spatial locations in the domain D, and the real-valued output is interpreted as an estimate of the underlying scalar field at that location.[33] It is natural to treat Σ and $\hat{\Sigma}$ as topological spaces with the subspace topology inherited from $\mathbb{R}^3$, and to treat the relevant structure-preserving map as a homeomorphism.[34] Concretely, the model tracks the structural property p insofar as the learned surface $\hat{\Sigma}$ is topologically equivalent to Σ, for example insofar as there exists a homeomorphism[35]

[31]The network is a fully connected feed-forward neural network with ReLU activations (as defined in equation 2), specifically, a 3-layer multilayer perceptron with architecture `Linear(3→64) → ReLU → Linear(64→64) → ReLU → Linear(64→1)`. The model has a total of 4,481 learnable parameters. It is trained to regress the scalar field $F(x, y, z)$ whose zero-level set defines the torus, using 262,144 points sampled on a 64^3 grid over D. The loss function is mean squared error (MSE), optimized using Adam with a learning rate of 10^{-3} for 5000 epochs. See the accompanying repository for the implementation details.

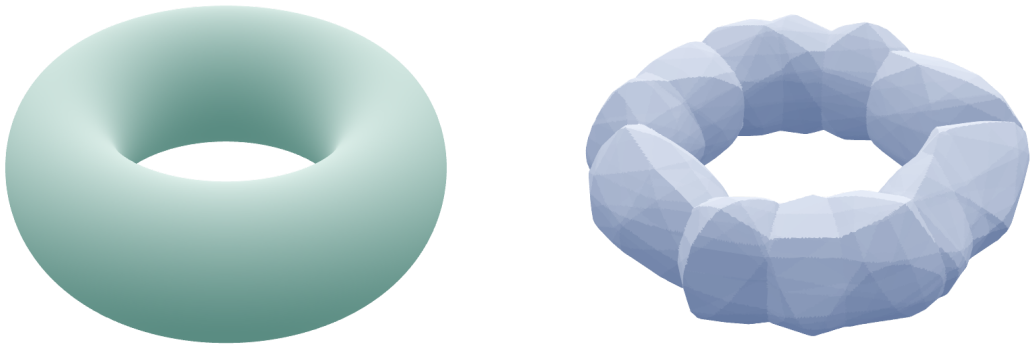
[32]Note that I extract the learned surface using the marching cubes algorithm (Lorensen and Cline, 1987) on a fixed grid. The resulting triangulation is therefore an artifact of the extraction procedure rather than a feature of the learned model itself. What matters for present purposes is that the reconstructed surface is globally coherent and piecewise planar, allowing its large-scale geometric and topological structure to be inspected. In this sense, marching cubes functions as a tool used by the human investigator, rather than as part of the agent system, to test hypotheses about the topology of the learned surface. The agent system's internal model, understood as a learned scalar field, exists independently of this visualization procedure.

[33]We might specify further bridge principles, mapping from the scalar field to a geometric object: the 0-level set is interpreted as a surface. Furthermore, the agent system derives properties of p by applying a fixed extraction-and-measurement procedure (marching cubes followed by Euler characteristic, hence genus) to the learned field. On this construal, the derivation step is performed by S as a whole, namely the trained network taken together with this fixed post-processing pipeline.

[34]Here, *homeomorphic* means that there exists a continuous bijection between the reconstructed surface and the standard torus whose inverse is also continuous; equivalently, the two surfaces have the same topology (in particular, the same pattern of connectedness and holes), even if their geometries differ.

[35]Note that, in practice we do not construct h explicitly. Instead, we test the intended structural

$$h : \Sigma \to \hat{\Sigma}. \tag{18}$$



(a) Ground truth surface (b) Learned isosurface

Figure 3: Comparison between the ground truth torus surface (left) and the neural network's learned isosurface (right). Observe how the learned isosurface is made by stitching together piecewise-linear surfaces. Nonetheless, it forms a coherent genus-one surface.

The trained model produces a surface with the correct **topological genus** (1), capturing the defining property of a torus: the presence of a single hole, and hence is homeomorphic to a torus.[36] In this sense, the learned model reproduces the salient topological structure of the target system, even though the training signal contains no explicit topological supervision. Consequently, according to our criteria, the network exhibits a form of structural understanding of the topological property $p = \textit{genus} = 1$. This is not a trivial accomplishment, as no global topological information about the genus was supplied to the model during training. The model was trained entirely on local scalar values, and yet the learned function globally reconstructs the correct topological structure (figure 3).

This case illustrates several key points. First, structural understanding does not require explicit symbolic encoding of the target property. The network lacks any concept of genus, holes, or topology in the formal sense, and yet it implicitly represents those features through the structure of the learned function. In this sense, it may be somewhat analogous to cases of tacit understanding in humans. Second, this understanding emerges from local supervision alone: the system infers global structure from local

agreement by comparing topological invariants of the reconstructed surface.

[36] As a check, let $\hat{T}_h$ be the triangulated 0-isosurface extracted from f_θ by marching cubes on an h-spaced grid. In our reconstruction $\hat{T}_h$ is a single closed connected surface with Euler characteristic $\chi = 0$, hence genus $g = (2 - \chi)/2 = 1$.

data. Finally, the ReLU network's piecewise-linear function class is expressive enough to model a globally nonlinear manifold (a toroidal surface) by stitching together many locally affine patches, a realization of the spline interpretation discussed in section 5.

Thus, the neural network achieves a form of structural understanding of the torus: it creates a model that tracks and reproduces a salient topological property of the target system through its internal functional representation, without any explicit symbolic manipulation. The example illustrates that under appropriate conditions, deep learning systems can internalize structural regularities like topological genus.

6.2 Kepler's Elliptical Orbits

Our second example concerns a different kind of global structure: not a topological invariant of a surface, but *geometrical* regularities in a dynamical system. Unlike our previous example, these geometric quantities will only be learned *approximately*. Recall, in section 4 I argued that Kepler achieved a greater understanding of the planetary orbits by realizing he could derive Brahe's descriptive ephemerides with a compact geometric model, in which the Sun occupies a focus of an ellipse (Kepler, 1992). Instead, let us suppose that, in a moment of divine insight, an inventor (perhaps Wilhelm Schickard) presented Emperor Rudolf II with the *Denkrechnuhr*, a mechanical neural network, to tackle the problem of planetary orbits.[37] Could such a system have achieved the same understanding as Kepler?

The target system T is the orbit of Mars relative to the Earth and Sun. The properties of interest, p, include the basic observational regularities about the orbits (e.g. the synodic periodicity, and the profile of apparent angular speed), as well as the property that Mars's heliocentric orbit is well-approximated by an ellipse with eccentricity e, one of Kepler's notable findings being that the orbit was non-circular, $e \neq 0$. We will take the Denkrechnuhr to be the agent system, S. Concretely, S consists of (i) a fixed neural network architecture, (ii) the learned parameter values obtained by training on the historical observations, and (iii) a fixed procedure for producing predictions and evaluating error. At prediction time, S receives a date (represented as the number of days since the first observation) and outputs a predicted sky position for Mars in ecliptic coordinates (longitude and latitude). We will take a subset of Tycho Brahe's observations for Mars as the dataset (Dreyer, 1913; Thoren, 1990).[38] The system receives a date (represented as days since the first observation) and predicts the ecliptic longitude and latitude for Mars.

We can consider two versions of the Denkrechnuhr, differing in model and bridge

[37] Historically, Schickard described a calculating clock (or *Rechenuhr*) in letters to Kepler in 1623–1624 (see Kistermann 1985; Seck 2005).

[38] These consist of 12 recorded oppositions. Each observation provides Mars's geocentric ecliptic longitude and latitude at a date $(\text{day}, \text{month}, \text{year})$. In addition, the dataset includes the mean ecliptic longitude of the Sun at those epochs. A second table provides 10 triangulation-style constraints, giving Earth heliocentric longitude and Mars geocentric longitude at additional dates. We represent each time-stamp by the number of days elapsed since the earliest observation, $t \in \mathbb{R}_{\geq 0}$, and represent angles in radians.

principles.[39] Let us call them the Baseline version, S_B, and the Keplerian version, S_K. In the baseline version, the output is produced by directly regressing angles on time. The learned model $\mathcal{M}_B$ consists of two continuous, piecewise-linear, *spline-like* maps from time to longitude and latitude, treating the observed angular trajectories as primitive objects to be fit. The bridge principles consist of the mapping from dates to temporal inputs, and the interpretation of the model's outputs as sky angles measured in fixed coordinates. These mappings fix how the learned curves are to be taken as representing observational regularities, without directly positing any underlying spatial orbit structure. As such, the Baseline version treats Mars's observed sky-position as a time-indexed curve to be learned directly. We can think of it as drawing a smooth path through the recorded points on two graphs: one graph for longitude-versus-time and one for latitude-versus-time. Training amounts to adjusting the shape of those two curves until, at the recorded dates, the curves pass close to Brahe's measurements.

In the Keplerian version, the prediction procedure maps dates to angles *indirectly* by first computing where Earth and Mars would lie under a simple orbital model with a small number of adjustable parameters, and then converting that direction into the predicted longitude and latitude. The model, $\mathcal{M}_K$, consists of a geometric representation of heliocentric motion, parameterized by orbital elements such as period, eccentricity, and orientation, together with deterministic rules that derive predicted sky angles from these spatial variables. As such, the learned spline encodes the set of global orbital parameters that pick out a specific Keplerian orbit (an ellipse with a particular eccentricity, orientation, and period). The system then computes the planet's position along that orbit over time and derives the corresponding sky angles by fixed geometric relations. Consequently, the observed angle–time curves are treated as consequences of the underlying geometric model. The bridge principles consist of the temporal mapping from dates to model times, the geometric mapping from internal spatial variables to heliocentric positions, and the coordinate transformations that map those positions to observable angular quantities. These mappings ensure that the observed angular trajectories are treated as consequences of an underlying spatial model rather than as primitive curves.

Necessarily, the Keplerian version bakes in some of Kepler's own assumptions: that planets move around the Sun on simple geometrical paths, and that the apparent motion of Mars arises from the *relative motion* of Mars and Earth. Historically, Kepler's decisive move was to abandon ad hoc combinations of circles and epicycles and to search for a single simple curve that could generate the observations (Voelkel, 2001). In modern terms, his hypothesis was that the orbit is a *conic section*, and in the bound case an *ellipse*, with the Sun at one focus (Kepler, 1992).[40] Instead of treating the angles as

[39] See the accompanying repository for the implementation details.

[40] Kepler's assumptions function as strong inductive biases here. The model represents Mars's heliocentric path as an ellipse described by a small number of adjustable quantities (including its eccentricity, overall scale, period, and orientation). It then uses Kepler's area law (equal areas in equal times) to determine where Mars lies on that ellipse at each date. It treats Earth's motion in a simplified way and subtracts Earth's heliocentric position from Mars's to obtain the Earth-to-Mars direction. Finally, it converts that direction into the predicted ecliptic longitude and latitude that correspond to the observational data.

curves to be drawn directly, the Keplerian version treats them as consequences of an underlying spatial motion.

Both systems achieve a degree of predictive success. The Baseline system accurately reproduces the observed positions at the training dates and interpolates between them. It captures several genuine regularities of the target system, including the synodic period of Mars relative to Earth and the qualitative profile of apparent angular speed, including retrograde loops. In this sense, the Baseline model tracks real patterns in the data and so satisfies a very minimal condition for systematic understanding. What the baseline does *not* provide on its own is a bridge from these observational quantities to heliocentric orbital elements like eccentricity or semi-major axis; extracting those would require imposing an additional spatial orbit model, which is precisely what the Keplerian version builds in from the start.

The Keplerian system can track additional geometric properties of the orbits. In our results, it learns a Mars eccentricity value $e_M \approx 0.096$ and a period $P_M \approx 687.05$ days, both close to modern estimates (Mars has eccentricity ≈ 0.093 and a sidereal period of ≈ 687 days) (NASA Jet Propulsion Laboratory, 2019). So, once the hypothesis class is restricted to Keplerian conic-section motion, the model can approximately track a global geometric property of the target system, namely non-circularity as quantified by e_M. What the model most clearly tracks here is a stable, nonzero eccentricity consistent with Kepler's elliptical hypothesis, rather than a fully faithful reconstruction of all geometric and dynamical details. So the system does build a rather accurate model of the orbital system, when we allow it to make certain assumptions, similar to those of Kepler. A more general system might be able to optimize to select conic sections from a wider space of models, starting with fewer initial assumptions.

The Keplerian system achieves a richer homomorphism between a low-dimensional geometric model and the observed data, mediated by bridge principles that connect spatial positions to angular observations. As such, it more closely resembles Kepler's own achievement: not merely fitting the data, but showing how a wide range of observed phenomena arise from a single, simple underlying structure. At the same time, an important caveat applies. It is also plausible to regard the Keplerian system as exhibiting a genuine, though limited, form of reductive understanding. The target of reduction here is not planetary dynamics in full generality, but the structured body of observational regularities encoded in the angular ephemerides themselves. In the Keplerian model, those regularities are instead derived from a lower-dimensional spatial-geometric model of relative motion, together with explicit bridge principles linking heliocentric positions to observed angles.

It is perhaps unsurprising that the Keplerian system needs some greater input assumptions in order to achieve a greater understanding of the target system. One might reasonably question how much epistemic credit should be assigned to the learned parameters, as opposed to the hand-built scaffold in which learning occurs. Crucially, the Keplerian system's success depends on inductive biases that arguably encode some of Kepler's insight a priori. The restriction to conic sections, the assumption of heliocentric motion, and the use of area-preserving dynamics imposes a narrow hypothesis space for

the deep learning model. From one perspective, this weakens the epistemic significance of the result: the system does not discover ellipticity from a neutral starting point, but rather selects parameter values within a family already known to be appropriate. From another perspective, however, this mirrors scientific practice. Kepler himself did not search the space of all possible curves, but worked within a constrained space of geometrically motivated hypotheses. On this interpretation, the Keplerian system exhibits a genuine, though conditional, form of structural understanding: conditional on the correctness of the imposed modeling assumptions, it successfully tracks a real and explanatorily salient property of the target system.

Ultimately, it is debatable as to the degree to which the Keplerian system is really a case of understanding in deep learning systems, as opposed to understanding in extended systems with inbuilt structures and biases (although it does fit the guidelines I suggest in section 4.2). Nonetheless, it is plausible that a more powerful deep learning model could feasibly operate without such stringent structures and biases. I leave this as an open question. Regardless, the Keplerian system demonstrates how the choices of representation and bridge principles can strongly influence the degree of understanding that any given system can develop.

6.3 Modulo Addition

Our third example provides the clearest illustration of a system transitioning from memorization to genuine systematic understanding. We study a transformer-architecture neural network trained to perform *modular addition*, closely following a model introduced by Power et al. (2022) and analyzed by Nanda et al. (2023). We use this setup for a different purpose. Modular addition serves as a particularly clean test case for the idea that a system can move from merely *memorizing* observed input-output pairs to internalizing a *systematic* representation of the target structure. In the first possibility, the system internalizes a distributed lookup table, storing enough information about the particular training pairs it has seen to output the right labels for those pairs, without encoding any general rule. In the second possibility, the system internalizes a procedure that effectively computes the remainder of $a + b$ for *any* input pair, in other words it builds a *model* of modulo addition. These two possibilities come apart sharply once we withhold a substantial fraction of the m^2 possible pairs for testing. A memorizing solution has no particular reason to succeed on withheld pairs, since those pairs are not "near" the training pairs in a straightforward sense (unless there is some kind of internalization of the modular nature of the task). By contrast, a genuinely rule-like solution should succeed uniformly across all pairs, because the same procedure applies everywhere.

The target system is the finite abelian group $\mathbb{Z}_m \times \mathbb{Z}_m$, where $\mathbb{Z}_m = 0, 1, \ldots, m-1$ denotes the integers modulo m, equipped with componentwise addition modulo m. We choose to fix $m = 113$.[41] In modular addition, we add two numbers as usual, and

[41] We choose m to be prime because then $\mathbb{Z}_m$ has especially clean algebraic structure: every non-zero element has a multiplicative inverse, and (equivalently) $\mathbb{Z}_m$ forms a field. This avoids certain extra

then keep only the remainder after dividing by m. For example, modulo 113 we have $(110+5) \bmod 113 = 115 \bmod 113 = 2$. We train on a random fraction of all m^2 pairs (in our main training run, 60%), and we test on the remaining held-out pairs. The property of interest is which output, y, is correct for each input pair of integers (a, b).

The agent system S is a trained network, together with the fixed architectural choices, training procedure, and decoding conventions that define its end-to-end input–output behavior. Following the spirit of Nanda et al., we use a small transformer architecture, trained with gradient descent and weight decay.[42] The model, $\mathcal{M}$, is the learned input–output function implemented by the network, understood as a score surface over the discrete grid of inputs, together with the internal representations that support that mapping. The bridge principles include the representation of each input integer by a discrete token supplied to the network, the fixed association between output positions and the possible remainders $0, 1, \ldots, m-1$, and a decoding rule that selects the output with the highest score as the system's answer. Taken together, these mappings fix how the network's internal activity is to be understood as implementing addition modulo m rather than some other input–output transformation.

Upon training and testing, we see a clear case of the *grokking* phenomenon described in section 3.1. As shown in figure 4, the system attains near-perfect *training* accuracy quite quickly, while *test* accuracy remains low for a long time, and then, after many further training steps, test accuracy rises sharply to near-perfect values. Following the analysis of Nanda et al. (2023), it seems that the training initially settles into a solution that fits the training pairs by a form of partial memorization, but which fails to generalize to new, non-memorized pairs. However, later the system seems to internalize a (highly compressed) generalizing model of modulo addition, one which also succeeds at new test pairs.

We gain further insight by analyzing the structure of the system's learned surface, as with our previous two examples. Suppose that the system receives an input, a, b; then let $S_y(a, b)$ be the model's score assigned to each possible answer y. A higher score means the model weighs y more highly as an answer. For example, if our system has succeeded in learning additional modulo 113, we would expect $S_2(110, 5)$ to be high, and $S_y(110, 5)$ to be low for every $y \neq 2$.[43] Thus, for each fixed output class y, the assignment

symmetries and degeneracies that can appear for composite moduli (where $\mathbb{Z}_m$ has zero divisors).

[42]The model takes a two-token input sequence $[a, b]$, embeds each token into $\mathbb{R}^{128}$, adds learned positional embeddings, applies a stack of transformer encoder blocks (in our implementation, two layers with 4 attention heads and a 512-dimensional feed-forward sublayer), and then produces logits over the p possible outputs from the representation at the second token position. Our implementation is intentionally minimal. In particular, we do not include an explicit "=" token as in Nanda et al. (2023), and we use a standard PyTorch encoder block. These differences are not philosophically important here, since our aim is not a fine-grained mechanistic reconstruction of the internal circuit, but rather a demonstration that the learned input-output map can become both compact and systematically correct. We train with cross-entropy loss on the training pairs, and we evaluate accuracy on *all* held-out pairs. See the accompanying repository for the implementation details.

[43]Many neural-network classifiers convert these scores into probabilities by applying a *softmax* transformation. Nothing in the present analysis depends on that step. What matters is that higher scores indicate that the model "prefers" that answer, and the model's predicted answer is the y with the highest

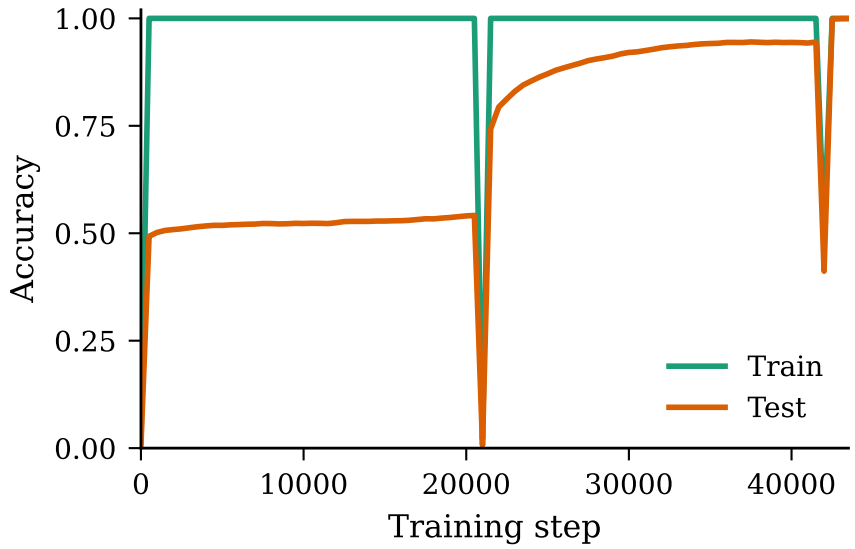


Figure 4: Training and test accuracy as a function of optimization step in the modular-addition experiment. Training accuracy rises rapidly to near-perfect performance on the seen input pairs, while test accuracy remains low for an extended period. After a period of continued training, the model *groks* the target system, with test accuracy suddenly increases sharply, nearing the training accuracy. The sharp spikes at around steps 22,000 and 42,000 are consistent with the "slingshot" grokking mechanism identified by Thilak et al. (2022).

$(a, b) \mapsto S(a, b, y)$ is simply a real-valued function on the finite grid $\mathbb{Z}_m \times \mathbb{Z}_m$. We can therefore view it as a numerical table with $m \times m$ entries, one entry for each possible input pair.

Any such function on a finite grid admits a discrete Fourier decomposition.[44] This gives a clear way to extract the structure of the learned dependence on (a, b). Roughly, if the model's score table is complicated in an idiosyncratic, example-by-example way (as one expects from a memorizing lookup strategy), then its Fourier representation tends to be spread broadly across many frequencies. By contrast, if the score table is governed by a simple underlying regularity, then its Fourier representation tends to be *concentrated* on a small, geometrically meaningful subset of frequencies. In particular, the rule should depend only on the sum $a+b$ modulo m.[45] Empirically, this is exactly what we observe.

score.

[44]Concretely, one expands in the characters $\exp(2\pi i(k_a a + k_b b)/m)$, where $(k_a, k_b) \in \mathbb{Z}_m^2$. The discrete Fourier transform returns the coefficients of this expansion.

[45]This imposes a strong symmetry: inputs with the same value of $a + b \bmod m$ lie on diagonals of the (a, b) grid, and the correct answer is constant along each such diagonal. When a learned function is primarily "a function of $a + b$", this symmetry sharply constrains which Fourier components can carry substantial weight, and one expects the Fourier magnitude to concentrate on a low-dimensional locus rather than being diffuse across the plane.

Figure 5 visualizes the average Fourier magnitude (averaged over output classes y), and it exhibits a pronounced concentration pattern rather than a high-entropy spectrum. This provides further, strong evidence that the system really is internalizing a model of modular addition.[46]

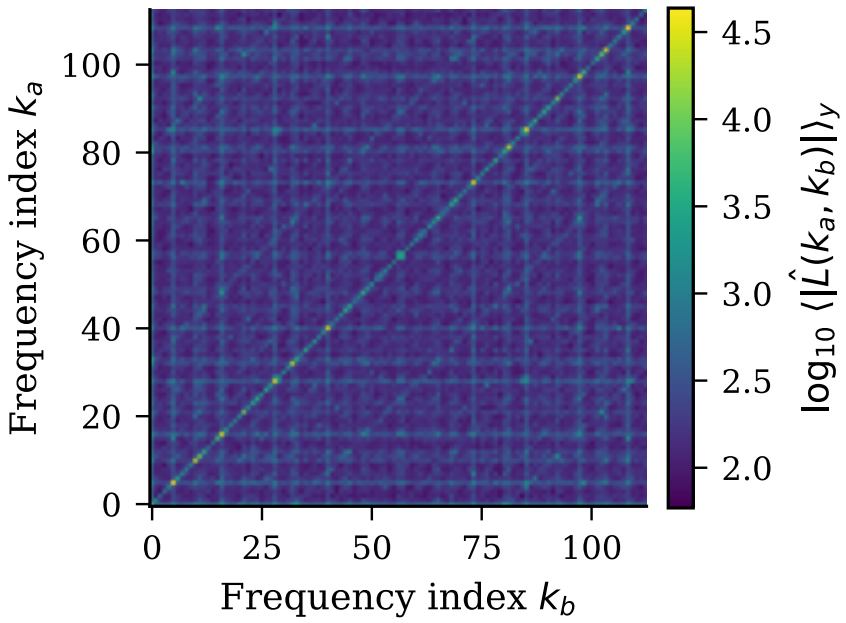


Figure 5: Average magnitude of the 2D discrete Fourier transform of the system's logits over the (a, b) input grid (averaged over output classes). The learned input-output map exhibits strong concentration in Fourier space, consistent with a compact, structured dependence on the inputs.

The evidence strongly suggests that the model $\mathcal{M}$ does not achieve success in an arbitrary way.[47] The patterns of the learned score surface are organized around the same symmetry that organizes the target rule, namely dependence on the sum $a + b \bmod m$. There is a structure-preserving alignment between the organization of the model's internal map from inputs to preferred outputs and the organization of the target rule itself. The model is not merely correct, it is correct *for the right kind of reason*, namely because its internal representation tracks a genuine regularity of T rather than a

[46] Nanda et al. (2023) find that their grokked solution can be described as a "Fourier multiplication" algorithm that converts addition into composition of rotations, implemented by a small circuit inside the transformer.

[47] However, It is worth noting what enabled this transition. The training included weight decay, a form of regularization that penalizes large parameter values and thereby biases the system toward simpler solutions. Without it, the system remains in the memorization regime indefinitely (Power et al., 2022). Additionally, the target domain possesses a genuine simple regularity, the group-theoretic structure of modular addition, which falls within what the architecture can represent.

patchwork of training-specific contingencies.[48] This seems to be an especially clear-cut case of the system acquiring structural understanding.

6.4 The Game Othello

Our final example occupies a middle ground between the highly restricted modular-arithmetic task of section 6.3 and the rich, open-ended domains usually associated with natural language, and demonstrates that a system trained on sequential data can learn a hidden internal model of a complex state space. The system is trained on the board game Othello, developed and analyzed by Li et al. (2023b). We shall see that an agent system trained on data about game moves can nonetheless learn a hidden internal model of the game rules and board state. As Othello noted of Iago, this system "*doubtless Sees and knows more, much more, than he unfolds*" (Shakespeare, 1997, Act III, Scene III).

So our target system T will be the Othello game dynamics (see figure 6), understood as a deterministic state-update process over board configurations, together with a legality constraint. Othello is played on an 8×8 square grid. Players alternate placing discs of their color on empty squares. A move is legal only if the placed disc *outflanks* (sandwiches) one or more contiguous lines of the opponent's discs in any of the eight directions; those outflanked discs are then flipped. The game begins from a fixed four-disc configuration in the center, and ends when no legal moves remain. Although the rules are simple to state, the game tree is large. Moreover, the legality of a move depends on a global property of the game state, namely the configuration of discs across the entire board, rather than on any simple local feature of the immediately preceding move.

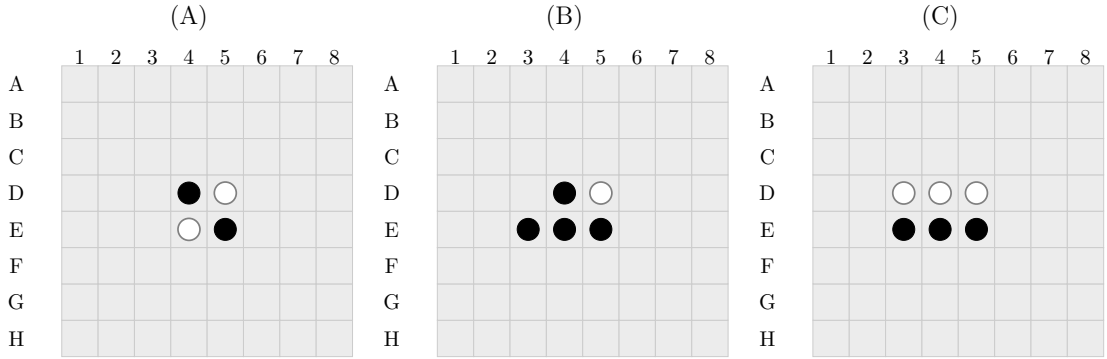


Figure 6: A visual demonstration of the Othello rules. From left to right. (A) The board is initialized with four discs placed at the center. (B) Black moves first and must outflank one or more opponent discs. (C) The opponent repeats this process. The game ends when no legal moves remain.

[48] Nonetheless, the bridge principles that make the task well-defined (the tokenization of residues, the fixed decoding rule, the fixed modulus m) are doing substantial work, and nothing here suggests that the system has a general grasp of arithmetic beyond this carefully delimited setting.

Formally, we may represent the board state after t moves as a function

$$B_t : \mathcal{X} \to \text{black}, \text{white}, \text{empty}, \tag{19}$$

where $\mathcal{X}$ is the set of board squares, and where (B_{t+1}, m_{t+1}) is determined from (B_t, m_t) by the rules of play (with m_t the side to move at time t). One property of interest p_l is the set of legal next moves. We will also focus on another particular property of interest, B_t, the state of the board at a given time, t. That is, we will ask whether the agent system's internal model contains a representation of the board state.

Li et al. (2023b) train a General Purpose Transformer (GPT)-style language model, Othello-GPT, which we will take to be our agent system, together with a fixed inference procedure for next-move prediction. The model, $\mathcal{M}$ involves the internal activation patterns of the trained network, together with the pathways by which those activations influence next-move predictions.[49]

At each step, the system receives the sequence of "move tokens" corresponding to the moves played so far, and it produces as output a probability distribution over possible next move tokens. These move tokens are discrete symbols drawn from a fixed vocabulary. Each token corresponds to one of the 64 squares on the 8×8 board, and represents the act of placing a piece on that square. A sequence of move tokens therefore represents a sequence of moves in a game, and, together with the fixed rules of Othello, determines a unique board configuration at each time step. The bridge principles include the convention that maps each move token to a specific square on the Othello board, and a fixed update rule that determines how a sequence of such moves generates a board configuration.

As such, the system's sole training objective is next-token prediction: given a partial game transcript, it must predict the next move made by a (synthetic or human) player. Crucially, the agent system is never given the rules of Othello and never shown the board state. All structural information about the game must therefore be inferred from statistical regularities in the move sequences themselves. Nonetheless, the trained model predicts legal moves with very high accuracy on held-out games. At a behavioral level, this already goes beyond straightforward memorization. Li et al. (2023b) show that performance remains high even when large portions of the game tree are systematically removed from the training distribution, ruling out the hypothesis that the model simply recalls previously seen continuations. Instead, it must have internalized some regularities in the game rules. This can be tested further with probes and interventions.[50]

[49] See the accompanying repository an implementation of a small-scale replication.

[50] It is important to distinguish two claims that are easily conflated. First, *decodability* results show that some information about B_t is present in internal activations, in the sense that an auxiliary predictor can recover it. Second, *causal-use* results (for example, activation interventions that systematically shift predicted moves in line with counterfactual legality constraints) provide evidence that the model's behavior depends on that internal information. Probes primarily support the first claim; interventions primarily support the second. Note that the probe and intervention analyses are epistemic tools used by the human (not the agent system) to test hypotheses about the network's internal structure. The agent system's model exists independently of them, but is illegible to humans. These probes and interventions are not part of the agent system's deployed pipeline, and are not needed for the agent system's understanding.

A probe is an auxiliary model trained *post hoc*, after the main agent system has been trained and frozen. They are not part of the original agent system and do not affect the system's behavior. The probes receive as input a hidden activation vector $h(t)$ (representing the weights for a particular inner layer and time step t) from the Othello-GPT network and are trained to predict the underlying game state, here the board configuration B_t. Concretely, probes are trained on pairs of the form $(h(t), B_t)$. Probes therefore provide a direct test of whether the internal states of the model provide sufficient information to reconstruct the state of the board. Li et al. (2023b) demonstrate that nonlinear probes can predict the state of the board from this information with low error rates.[51] The natural conclusion is that the Othello-GPT system learns an internalized, accurate-but-imperfect model of the board state.[52]

Second, Li et al. (2023b) test whether the decodable board information plays a causal role in next-move prediction. The basic idea is to take an internal activation state, decode a baseline board B, and then *modify* the activation so that the probe instead decodes a counterfactual board B', differing from B at a chosen square. One then asks whether the model's predicted legal moves shift in the manner one would expect *if* the model were consulting the internal board representation in order to generate its output distribution. If changing only the internal state (without changing the input transcript) yields systematically different predicted moves aligned with the *counterfactual* legality constraints, that is strong evidence that the model computes and uses an internal representation that functions as a proxy for the board state. They test deliberately unnatural board states that are far from the training data distribution, including states that cannot arise from any legal sequence of moves. They find that intervening on the internal representation produces predictions aligned with the legal moves of the intervened board even in these stringent settings (Li et al., 2023b). This is difficult to reconcile with a picture on which the model merely memorizes a catalogue of local transcript correlations. Once again the natural conclusion is that the model has learned a model that represents the board state.

Thus, there is strong evidence that our agent system has learned an *adequate model component* $M \subset S$ that encodes information sufficient to reconstruct the board state. Second, there is a plausible tracking relation: the internal variable covaries with, and is

[51] In particular, for models trained on synthetic game data, a simple two-layer nonlinear multilayer perceptron (MLP) probe recovers the color or emptiness of individual board squares with error rates as low as roughly 2 percent in mid network layers, corresponding to correct reconstruction of over 98 percent of the board state. Importantly, analogous probes trained on a randomly initialized network fail, indicating that the success of the nonlinear probes reflects learned structure rather than hardwired properties of the architecture.

[52] A subsequent analysis by Nanda (2023) takes this further. From the model's internal perspective, it is often more natural to represent each square not as "black versus white", but as "*mine* versus *theirs*". Once one makes this change of variables, much of the representation becomes linearly accessible! They show that, once the representation is viewed in an appropriate coordinate system, even *linear* probes can accurately recover the board state. Linearity is significant here because a linear probe can only succeed if the relevant information is already organized in the model's internal space as a separable variable, rather than being diffusely encoded across complex nonlinear interactions (see Alain and Bengio, 2016; Bricken et al., 2023; Cunningham et al., 2023; Elhage et al., 2021b,2; Hewitt and Manning, 2019).

causally implicated in, the model's next-move distribution in a manner aligned with the target's legality constraints. Third, bridge principles connect move tokens to board updates and connect internal activations to decoded board variables (whether in black/white or mine/theirs form). Fourth, the system uses this internal structure in its predictions. By our previous definitions, it seems reasonable to conclude that the system has achieved a systematic understanding of the Othello game and the state of the board.

7 The Fractured Understanding Hypothesis

The four preceding examples each show cases in which deep learning systems *can* acquire a degree of systematic understanding. In the torus case, the system learned exact, discrete topological properties of the target system, despite being trained only on local scalar data and despite implementing its hypothesis as a piecewise linear spline. In the Keplerian case, it learned to approximate continuous geometric properties, such as orbital eccentricity. In the modular-addition case, the transformer eventually internalizes a symmetry-respecting rule that generalizes correctly across the target domain. In the Othello case, the system learns an internal state variable that tracks the evolving board configuration and is used to guide the next-move predictions.

However, the forms of understanding that deep learning systems typically acquire rarely resemble the **ideal of scientific understanding** proposed in section 4.7. In general, they are not straightforwardly reductive, they are symbolic but use unnatural symbolic representations, and they are only weakly unified.

7.1 Non-Reductive Understanding

As spline-based function approximators, such systems are optimized to fit observed input–output relations as flexibly as possible, but these spline-like surfaces may be unsuitable in practice for capturing the salient structures of many target systems. This makes them extremely effective at capturing *local* regularities, but comparatively ill-suited to discovering global regularities. Often, the target system may be well suited for a simple description in one coordinate system, or one hypothesis class, but the deep learning system can only represent a large patchwork of local facets, which fail to latch onto the target system's more general structures. In that case the learned map can be accurate on the training data while failing to preserve the right structure under the perturbations that matter, because small changes can move inputs across region boundaries of the learned surface. As such, the understanding is usually **structural** but rarely **reductive**.

Recall that deep learning systems are trained to minimize some loss quantity over samples drawn from a particular data-generating process. The training objective is rarely to discover a generalizable, reductive model of the underlying target structure T; it is to achieve low error on the observed input–output behavior induced by the bridge principles. Consequently, the easiest route to low loss is often to capture *whatever* dependence structure is stable *in the training distribution*, even if that dependence does

not coincide with the intended property p of the target system. As a result, even when the system achieves impressive generalization, the resulting internal model $\mathcal{M}$ need not be reductive, unified, or cleanly factorized into reusable variables.

This matches more general observations of deep learning systems. Geirhos et al. (2020) discover that many apparent successes of deep neural networks can be traced to *shortcut learning*: the acquisition of decision rules that perform well when training and test data are drawn from the same underlying distribution, yet fail to transfer under modest distribution shifts or more demanding test conditions. Shortcut solutions are not mere memorization, but they often encode *regularities* that do not fully generalize. Deep learning systems are sensitive to superficially stable cues in the distribution of the training data but may miss deeper regularities in the target system.

Arithmetic in large language models might offer a further example. Contemporary LLMs succeed at many basic arithmetic tasks, such as addition and multiplication, but are increasingly likely to fail as numbers become larger (Saxton et al., 2019). In a mechanistic study, Nikankin et al. (2025) find that transformer models are neither memorizing arithmetic patterns, nor learning the true underlying rules. Rather (perhaps like many students), they learn a *bag of heuristics*.[53] The upshot is that they acquire a partial understanding: the system has latched onto genuine patterns in arithmetic, but these regularities do not fully generalize.

7.2 Symbolic Misalignment

Likewise, taken literally, a trained neural network does implement a symbolic object, namely a function defined by a finite parameter vector. But the symbols it employs, the coordinates of a high-dimensional activation space and the boundaries of affine regions, do not usually correspond to the natural symbolic structure of the target domain. Instead, the neural network systems usually seem to learn a patchwork of locally reliable regularities, stitched together into a large continuous function (or, in discrete settings, a large score table).

Recall that, in the Kepler *baseline* system (section 6.2), regressing longitude and latitude directly on time encourages precisely such a patchwork fit to the observed data: it captures real periodicities, but it cannot build a deeper general orbital model. By contrast, the Keplerian variant achieves a more robust kind of tracking mainly because we *change the representational interface*: we forced the learned degrees of freedom to live in orbital-element space, so that the spline is fitting a small number of global parameters rather than stitching together an angle-time curve. But this involved hardwiring inductive biases into the system.

An analogous pattern appears in modular addition (section 6.3): before grokking, the model can fit the training pairs using a diffuse, example-specific surface over (a, b) that fails to extend uniformly to withheld pairs, and only later does training discover a representation aligned with the group structure, at which point generalization becomes

[53] Their experimental setup itself makes vivid one reason performance need not scale smoothly: each model tokenizes numbers as single tokens only up to a finite limit, and their analysis focuses on operand and result ranges within that regime.

global. Even the Othello case (section 6.4) reflects a milder version of the same phenomenon: the internal state is present, but in a coordinate system (mine/theirs) that better matches the task's invariances than our initial black/white description, which is why some structure is initially "hidden" until we adopt the right variables. The general lesson is that spline-like function classes are expressive enough to *approximate* many regularities, but they do not automatically discover the right variables in which those regularities become simple; when they do not, understanding can remain locally adequate yet globally brittle.

7.3 Weak Unification and Representational Fragmentation

In section 4.7, I argued that the ideal of scientific understanding is not only reductive and symbolic, but also unifying: it organizes many phenomena under a compact set of reusable principles. Deep learning systems often fall short of this ideal. Their internal successes may be real, but they are frequently not organized into a small stock of globally reusable rules.

Recall the distinction between *organic* and *monstrous* theories (Votsis, 2015). A unified hypothesis is one whose parts are *confirmationally connected*: evidence for one part lends support to the others, because they are bound together by shared structure. A "monstrous" theory, by contrast, is a mere conjunction of independent claims, each supported only by its own datum and confirming nothing about the rest. This notion of monstrosity gives us a precise way to characterize the patchwork understanding that deep learning systems often exhibit. A system whose separate responses to different inputs are only weakly confirmationally connected, each supported mainly by its own region of the training data, is monstrous in something like Votsis's sense, even if many individual responses happen to be correct. The modular-addition system *before* grokking is a particularly clear example: it can fit individual input–output pairs without yet organizing them under a single general rule. After grokking, the same responses become confirmationally connected, because the same algorithmic structure governs all of them.

This vocabulary speaks directly to a recent debate about whether artificial neural networks are amenable to unifying explanation. Prasetya (2022) argues that, on Kitcher's unificationist account, neural networks resist genuine explanation, because each trained network requires its own enormously complex derivation rather than instantiating a small number of reusable reasoning patterns. Erasmus and Brunet (2022) reply that this confuses being maximally unifying with being explanatory at all: even if neural-network derivations are less unifying than Newtonian mechanics, they do not thereby cease to be explanatory. Both sides capture something important. Prasetya is right that a typical trained network, with millions of individually tuned parameters, does not implement a small stock of reusable principles. However, Erasmus and Brunet are also right that some networks *do* discover simple, general internal structures that apply uniformly across their whole domain.

Related concerns also arise in the machine learning literature. Bengio et al. (2013) argue that the success of machine learning depends crucially on learning the right internal representations of target systems, in particular representations that disentangle

the underlying explanatory factors of the data, rather than operating directly on raw inputs. Learned representations often capture only those aspects of structure that are useful for the training objective, leaving other salient factors unrepresented or only implicitly encoded. Similarly, Kumar et al. (2025) argue that the objective-driven training of deep learning systems can yield internal representations in which the system's abilities are split into disconnected, redundant fragments, which then become entangled with other fragments in ways that hinder robustness and independent adaptation. Even in cases when output behavior is excellent, the internal strategy may therefore resemble an unruly patchwork rather than a unified, factored decomposition of the target structure. They call this the *Fractured Entangled Representation* hypothesis.

More broadly, a growing philosophical literature has begun to address these questions directly. Sullivan (2022) identifies "link uncertainty"(i.e. the evidential gap between model and target) as key to determining whether machine learning supports genuine understanding. Tamir and Shech (2023) propose indicators of machine understanding that naturally come apart across different systems and tasks. Beckmann and Queloz (2025) offer a tiered framework distinguishing levels of machine understanding, and observe that large language models rely on "parallel, heterogeneous mechanisms": different tasks within a single model are handled by different internal processes, some understanding-like and others not. These contributions characterize *what kinds* of understanding deep learning systems exhibit. The question that remains is *why* fragmented understanding is so common, and when, if ever, it gives way to genuine unification.

7.4 Why Fractured Understanding Is Common

Why is fractured understanding so common? The answer is partly structural. Deep learning hypothesis classes are highly permissive: they admit both relatively unified and highly patchwork solutions. Yet patchwork solutions are often easier to find and easier to preserve.

First, they are **available**. A sufficiently expressive model can fit training data in ways that need not cohere into a single globally reusable scheme. As Zhang et al. (2017) famously showed, the same architecture that generalizes impressively on structured data can also memorize pure noise to zero training error. The capacity that enables genuine pattern extraction therefore also enables rote memorization, shortcut dependence, and many intermediate forms of only partial or local adequacy. Such permissive hypothesis classes leave ample room for patchwork solutions.

Second, patchwork solutions are often more **discoverable**. Gradient-based optimization proceeds by local improvement, and so may settle on strategies that are adequate across many cases before it discovers a more globally unified organization. This helps explain the prevalence of *shortcut learning*: models often latch onto features that are predictively useful in the training distribution, even when those features do not reflect the deeper structure of the task (Geirhos et al., 2020). Such strategies are not mere memorization, but neither do they amount to a robust grasp of the target system's more general regularities.

Third, and critically, patchwork solutions are often **stable**. Once a locally adequate

solution drives training loss sufficiently low, the objective typically supplies little pressure toward further reorganization. From the perspective of the loss function, a patchwork solution that performs well enough may be just as acceptable as a more unified one. Moving toward unification therefore usually requires some additional pressure toward simplicity, compression, or architectural constraint that the basic training objective does not itself provide.

This helps explain why Prasetya's observation is often empirically apt: many trained networks do not implement a small stock of globally reusable principles. Erasmus and Brunet are right, however, that this is not universal. Some systems do achieve more unified internal organization, but typically only when specific enabling conditions are in place.

The modular-addition case from section 6.3 illustrates the point. In that example, unification emerges when training includes sustained pressure toward compression, via weight decay, and when the target domain contains a simple discoverable regularity that the architecture can represent. Where such conditions are absent, as the LLM arithmetic case discussed above illustrates, fragmentation is more likely to persist. Conversely, some architectures can reduce the scope for patchwork by building stronger structural constraints directly into the model, for instance by encoding symmetries or conservation principles. The claim, then, is not that deep learning systems can never achieve unification, but that permissive hypothesis classes create a standing bias toward fractured understanding unless that bias is actively overcome.

7.5 Statement of the Hypothesis

These considerations motivate a **Fractured Understanding Hypothesis** (FUH). Deep learning systems are well-suited for acquiring *some*, *real* systematic understanding. But the understanding they acquire characteristically falls short of the unifying ideal, and does so in specific, predictable ways. The understanding often exhibits the following characteristic forms of fragmentation:

- **Symbolic Misalignment.** A trained network implements a precise symbolic function, but the internal "coordinates" in which it carries out the computation, such as activations, learned features, and affine-region boundaries, typically do not coincide with the target domain's natural variables, nor with variables that are easily legible to humans.

- **Non-reducibility.** Instead of representing the phenomenon by a small set of general principles that unify many cases, the model often implements a patchwork of locally effective mappings. Even when each component tracks a real regularity, the overall representation need not amount to a small set of reusable principles that generate the behavior across the whole domain.

- **Local fracturing.** Because the learned map is assembled from locally reliable pieces, its competence can depend on remaining within the same region of its internal partition of input space. Small, seemingly innocuous changes that push

an input across such a boundary can trigger qualitatively different behavior, even when the corresponding change in the target system preserves the salient structure we care about.

- **Proxy-dependent fragmentation.** Training often rewards any dependence that is predictively useful in the training distribution. As a result, the model may successfully identify regularities that exist only in the training data, while missing deeper regularities in the target system.

- **Representational fragmentation.** Even when the model tracks the intended property, it may not represent the underlying factors as separable components. Instead, they can be distributed across many units and mixed together, so that the model cannot easily be decomposed into parts that can be cleanly recombined or reused.

These five characteristics are predictable consequences of the structural forces identified above. Symbolic misalignment arises because optimization finds whatever internal coordinates minimize loss, not the target's natural variables. Non-reducibility arises because patchwork is a stable resting point once predictive adequacy is achieved. Local fracturing arises from the piecewise nature of the learned function. Proxy-dependent fragmentation arises because training rewards any predictively useful dependence in the training distribution. Representational fragmentation arises because there is no exogenous pressure to factorize internal representations cleanly. In Votsis's terms, the resulting model may be at least partly monstrous: its components are only weakly confirmationally connected, each locally adequate but collectively lacking the organic unity of a genuinely unified model.

None of this undermines the claim that deep learning can yield genuine systematic understanding. Very plausibly, a deep learning system can track properties of interest under certain kinds of variation while failing under others, because different regions of the input space, or different kinds of perturbation, are governed by different local regularities. The result is that models often look like a mosaic or patchwork of partially overlapping structures, each reliable within a particular domain. As such, the learned model will often achieve generalization in some regimes and not in others. This entails a partial understanding, in which some real, often local, regularities in the target system are learned. This may be one cause of the "jagged frontier" pattern introduced in section 1.[54]

Of course, as we have discussed, human understanding may often be very similar to this. As I stressed in section 4.7, much, perhaps most, human understanding is itself tacit, non-reductive, and probably piecemeal. When humans walk, catch a ball, navigate a landscape, play chess, or learn to use language, we typically do not deploy explicit symbolic models or derive predictions from a small set of general principles. I

[54] Although more fundamentally, given their difference in architecture and training regimes from humans and our evolutionary environment, it is unsurprising that artificial neural networks would have greatly different capabilities from us.

contend that these *are* real cases of human understanding, despite falling short of the ideal of scientific understanding. In many ways, the systematic understanding that deep learning systems acquire is, at least superficially, analogous to these cases of human understanding: it is usually non-reductive and technically symbolic, albeit in a non-natural way.

Taken together, these features help explain why the same system can appear, from one perspective, to display impressive understanding, and from another, to be epistemically shallow. The understanding it achieves is often real, but often partial: it tracks genuine regularities, often with remarkable precision, yet does so in a way that is locally constrained, distribution-sensitive, and poorly aligned with the explanatory structure of the target system.

With this being said, I do not claim that deep learning systems necessarily only exhibit fractured understanding. The examples we have seen suggest that they can and do develop a deeper understanding of many systems.[55]

8 Conclusions

I have proposed a minimal, naturalistic model of **systematic understanding**, suitable for deep learning systems. On this account, an agent system systematically understands a property of a target system when it contains an adequate internal model that tracks the relevant property without mere memorization, is coupled to the target by stable bridge principles and can be used to derive (approximately) correct predictions. This model might offer a clearer shared framework for the debate around understanding in deep learning systems. I hope that both proponents and skeptics of understanding in deep learning systems might locate their disagreements more precisely, or dissolve them, either by using this model of understanding, or by specifying alternatives to it.

I have argued that deep learning systems often do achieve real systematic understanding, on the basis of this model. This understanding is substantial and goes far beyond mere compression. However, such understanding often fails to resemble the classical ideal of scientific understanding. This motivates the **Fractured Understanding Hypothesis**. Even when deep learning systems track genuine regularities, their understanding is often symbolically misaligned with the target domain's natural variables, non-reductive and non-unifying, in the sense of being implemented as a patchwork of locally effective mappings rather than a small stock of reusable principles, and brittle,

[55]Indeed, it may be instructive to note that just as human researchers may develop successively improved scientific theories, so too can machine learning systems quite suddenly coalesce around new and improved models, under the appropriate circumstances. The double descent and grokking phenomena described in section 3.1 illustrate this in different ways: performance can appear to plateau or even degrade in one regime, yet later improve sharply as training or capacity pushes the system into a qualitatively different solution. This invites a loose analogy with the frameworks of Kuhn (1962); Lakatos (1978). In Lakatosian terms, a research programme may maintain empirical success for a time through increasingly protective adjustments before being displaced by a rival programme. Likewise, in the double descent phenomenon, phases of overfitting can give way to new models that better track the underlying structure of the target system.

with competence that can change sharply across internal boundaries or distributional shifts.

One clear future step would be to apply this model to understanding in transformer-based large language models. Applying this framework would have two distinct steps. The first is to assess whether language models can and do systematically understand *language* itself. Namely, do they build an internal model that captures the relevant structural distinctions in human language (whether those are syntactic features, compositional relations, or other contextual facets of linguistic meaning). The second is to assess whether such linguistic understanding, endows the models with understanding of the *world*. Here the target properties are not purely intra-linguistic regularities but extra-linguistic structures of the world that language describes, for example potentially including causal, spatial and other relations. An LLM would count as having a **world model**, in the present sense, only if some internal variables reliably track those extra-linguistic regularities and can be used to derive predictions that remain stable under interventions that preserve the underlying structure (Ha and Schmidhuber, 2018; Li et al., 2023a).

Furthermore, the basic formulation of systematic understanding here might naturally invite other variants, beyond structural and reductive understanding. One natural and potentially promising contender would be a form of **causal understanding**. On such a variant, the relevant tracking relation would not merely preserve patterns, but would also capture the target's *causal* dependency structure: the model's variables would correspond to features of the target system in a way that supports stable predictions under causal interventions, for example through a do-calculus framework (Pearl, 2018). This might well offer a richer conception of understanding than that offered here.

The framework also suggests some lessons for artificial intelligence development. First, deep learning can in principle achieve, and support, scientific and other kinds of understanding. However, this is greatly facilitated when the training, architecture, and interfaces of the system lead the learned representation to correspond with the target system's salient invariances. Importantly, the bridge principles and representational interfaces are not neutral conveniences; they are major determinants of what kinds of understanding are even *possible* to the system. The Keplerian contrast makes this explicit: changing the space in which learning occurs can convert a patchwork fit into a compact, globally meaningful model. More generally, if we care about scientific understanding, then we should expect to do substantive work in the design of measurement, featurization, tokenization, and decoding, not merely in scaling parameters.

Second, if fractured understanding is widespread, then it suggests a corresponding research program. We should expect progress to come from methods that encourage factorization, variable discovery, and robust structure-preservation. One natural approach would be hybrid architectures. Neuro-symbolic systems, broadly construed, promise a way to combine the expressive pattern-learning strengths of neural models with symbolic resources that support explicit variable-binding, compositional derivations, and re-use across contexts (Besold et al., 2018; Garcez et al., 2002; LeCun, 2022; Rocktäschel and Riedel, 2017; Votsis, 2024). Importantly, this should not be read as a return to a strict

dichotomy between "mere" pattern recognition and "real" reasoning. Human cognition itself plausibly combines tacit, model-based competence with more explicit symbolic scaffolding. There is no principled reason to rule out a comparable division of labor in artificial systems, whether by integrating symbolic modules, by training neural components to implement more explicit algorithmic subroutines, or by coupling learned representations to external tools and verification procedures.

Symbolic regression methods (Cranmer et al., 2020b; Kamienny et al., 2022; Petersen et al., 2021; Schmidt and Lipson, 2009; Udrescu and Tegmark, 2020) offer an alternative, but related, route forwards. Whereas standard deep learning typically yields an implicit model whose structure is difficult to inspect, symbolic regression aims to extract an explicit, human-legible functional form (often a compact equation) that approximately governs the dependence between variables in the learned representation or in the target data. When it succeeds, this can convert tacit structural understanding into something closer to the scientific ideal.

Other promising approaches to solving related problems could be explicitly causal models (Pearl, 2018; Peters et al., 2017; Schölkopf et al., 2021; Spirtes et al., 2000). Recent work on causal discovery and causal representation learning makes this idea precise, aiming to recover latent variables and structural equations that are invariant across heterogeneous data or experimental settings. Alternatively, scientific machine learning approaches, such as equivariant architectures and Hamiltonian or Lagrangian neural networks, suggest that imposing the right structural biases can turn locally effective fits into compact, globally meaningful models that more closely resemble the scientific ideal (Cranmer et al., 2020a; Greydanus et al., 2019).

The framework also motivates the growing body of work on mechanistic interpretability (Elhage et al., 2021a; Olah et al., 2020,1). If systematic understanding requires that internal variables track target-relevant structure and support stable predictions under intervention, then interpretability is not merely a transparency add-on but an epistemic tool. Mechanistic analyses aim to identify internal components and circuits that play determinate computational roles, allowing us to test whether competence depends on robust, reusable structure or on brittle, locally effective shortcuts.

If the general argument of this paper is correct, then we can make progress in the understanding debate without either inflating deep learning into a human analogue or dismissing it as mere mimicry. Deep learning systems can acquire real systematic understanding, but the default form of that understanding is often fractured. The philosophical and engineering challenge is therefore not to decide, in the abstract, whether deep learning "can understand", but to identify the conditions under which it does, to diagnose the characteristic failure modes, and to develop methods that turn locally successful tracking into more robust, communicable, and generalizable models of the world.

References

Achille, A., Paolini, G., and Soatto, S. (2020). Where is the information in a deep neural network?

Achille, A. and Soatto, S. (2018). Emergence of invariance and disentanglement in deep representations. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9.

Alain, G. and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Bitton, J., et al. (2022). Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in AI safety.

Anagnostidis, S., Bachmann, G., Noci, L., and Hofmann, T. (2022). The curious case of benign memorization. *arXiv preprint arXiv:2210.14019*.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. (2017). A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 233–242. PMLR.

Balestriero, R. and Baraniuk, R. G. (2018). A spline theory of deep learning. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 374–383. PMLR.

Balestriero, R., Pesenti, J., and LeCun, Y. (2021). Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*.

Ball, B., Freeborn, D., Helliwell, A., and Loi-Heng, K. (2025). Concepts and classification algorithms: A case study involving a large language model. Unpublished manuscript.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482.

Beckmann, P. and Queloz, M. (2025). Mechanistic indicators of understanding in large language models. *arXiv preprint arXiv:2507.08017*.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. Association for Computing Machinery.

Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Bereska, L. F. and Gavves, E. (2024). Mechanistic interpretability for AI safety—a

review.

Besold, T. R., d'Avila Garcez, A., Bader, S., Bowman, H., Domingos, P., Hitzler, P., Lamb, L. C., Lowd, D., de Penning, L., Pinkas, G., Poon, H., and Zaverucha, G. (2018). Neural-symbolic learning and reasoning: A survey and interpretation. *Cognitive Systems Research*, 52:1–22.

Bishop, J. M. (2021). Artificial intelligence is stupid and causal reasoning will not fix it. *Frontiers in Psychology*, 11:513474.

Block, N. (1981). Psychologism and behaviorism. *The Philosophical Review*, 90(1):5–43.

Bokulich, A. and Bokulich, P., editors (2011). *Scientific Structuralism*. Springer Science+Business Media, Dordrecht.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2022). On the opportunities and risks of foundation models.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. (2023). Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread. Available at `https://transformer-circuits.pub/2023/monosemantic-features`, accessed 2025-12-29.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J. A., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y.-F., Lundberg, S. M., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. (2023). Extracting training data from diffusion models. In *Proceedings of the 32nd USENIX Conference on Security Symposium*, SEC '23, USA. USENIX Association.

Chaitin, G. J. (1975). Randomness and mathematical proof. *Scientific American*, 232(5):47–52.

Chollet, F. (2019). On the measure of intelligence. *arXiv preprint*.

Churchland, P. M. and Churchland, P. S. (1990). Could a machine think? *Scientific American*, 262(1):32–37.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. (2020a). Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. (2020b). Discovering symbolic models from deep learning with inductive biases. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Cunningham, H., Ewart, S., Riggs, L., Hubinger, E., and Sharkey, L. (2023). Sparse autoencoders find interpretable directions in language models. *arXiv preprint arXiv:2309.10312*.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

da Costa, N. C. A. and French, S. (1990). The model-theoretic approach in the philosophy of science. *Philosophy of Science*, 57(2):248–265.

de Groot, A. D. (1965). *Thought and Choice in Chess*. Mouton Publishers, The Hague. Originally published in Dutch in 1946; English translation widely cited in cognitive psychology and chess literature.

de Regt, H. W. (2017). *Understanding Scientific Understanding*. Oxford University Press, New York.

Dell'Acqua, F., McFowland III, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K. C., Rajendran, S., Krayer, L., Candelon, F., and Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. Working Paper 24-013, Harvard Business School. Date written: September 15, 2023.

Dennett, D. C. (1991a). Real patterns. *The Journal of Philosophy*, 88(1):27–51.

Dennett, D. C. (1991b). Real patterns. *The Journal of Philosophy*, 88(1):27–51.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dewar, N. (2022). *Structure and Equivalence*. Cambridge University Press.

Dizadji-Bahmani, F., Frigg, R., and Hartmann, S. (2010). Who's afraid of nagelian

reduction? *Erkenntnis*, 73(3):393–412.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Dretske, F. I. (1981). *Knowledge and the Flow of Information*. MIT Press, Cambridge, MA.

Dretske, F. I. (1988). *Explaining Behavior: Reasons in a World of Causes*. MIT Press, Cambridge, MA.

Dreyer, J. L. E., editor (1913). *Tychonis Brahe Dani Opera Omnia*. In Libraria Gyldendaliana, Hauniae [Copenhagen]. Collected edition, 15 vols., published 1913–1929. Cite the relevant volume if you use a specific table/passage.

Dyson, F. J. (2004). A meeting with enrico fermi. *Nature*, 427(6972):297.

Elhage, N., Hume, T., Olsson, C., Nanda, N., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Lovitt, L., Hatfield-Dodds, Z., Kernion, J., Jones, A., Brown, T., Clark, J., Kaplan, J., McCandlish, S., Amodei, D., and Olah, C. (2021a). A mathematical framework for transformer circuits. Transformer Circuits Thread. Available at `https://transformer-circuits.pub/2021/framework/index.html`.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. (2021b). A mathematical framework for transformer circuits. *arXiv preprint arXiv:2104.08654*.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., et al. (2022). Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Erasmus, A. and Brunet, T. D. P. (2022). Interpretability and unification. *Philosophy & Technology*, 35(2).

Feldman, V. (2020). Does learning require memorization? a short tale about a long tail. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1–26. PMLR.

Ferrer-i Cancho, R. and Solé, R. V. (2003). Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences*, 100(3):788–791.

Floridi, L. (2023). Ai as agency without intelligence: on chatgpt, large language models, and other generative models. *Philosophy & Technology*, 36(1):15.

Freeborn, D. (2025a). Sloppy models, renormalization group realism, and the success of science. *Erkenntnis*, 90(2):645–673.

Freeborn, D. P. W. (2025b). Effective theory building and manifold learning. *Synthese*, 205(23).

Friedman, M. (1974). Explanation and scientific understanding. *The Journal of Philosophy*, 71(1):5–19.

Frigg, R. (2022). *Models and Theories: A Philosophical Inquiry*. Routledge, London, 1 edition.

Frigg, R. and Nguyen, J. (2017). Models and representation. In Magnani, L. and Bertolotti, T., editors, *Springer Handbook of Model-Based Science*, pages 49–102. Springer, Cham.

Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138.

Friston, K. (2013). Life as we know it. *Journal of the Royal Society Interface*, 10(86):20130475.

Friston, K. J., FitzGerald, T. H. B., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: A process theory. *Neural Computation*, 29(1):1–49.

Garcez, A. d., Gabbay, D. M., and Lamb, L. C. (2002). *Neural-Symbolic Learning Systems: Foundations and Applications*. Perspectives in Neural Computing. Springer.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673.

Georgi, H. (1993). Effective field theory. *Annual Review of Nuclear and Particle Science*, 43:209–252.

Gobet, F. and Lane, P. C. R. (2005). The chrest architecture of cognition: The role of perception in general intelligence. *Journal of Experimental and Theoretical Artificial Intelligence*, 17(3):209–236.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Greco, J. (2014). Episteme: Knowledge and understanding. In Timpe, K. and Boyd, C. A., editors, *Virtues and Their Vices*, pages 285–302. Oxford University Press.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Grimm, S. R. (2014). Understanding as knowledge of causes. In Fairweather, A., editor, *Virtue Epistemology Naturalized*, pages 329–345. Springer.

Grimm, S. R. (2017). Understanding and transparency. In Grimm, S. R., Baumberger, C., and Ammon, S., editors, *Explaining Understanding: New Perspectives from Epistemology and Philosophy of Science*, pages 212–229. Routledge.

Grünwald, P. D. (2007). *The Minimum Description Length Principle*. MIT Press, Cambridge, MA.

Grzankowski, A., Downes, S. M., and Forber, P. (2025a). Llms are not just next token predictors. *Inquiry*, pages 1–11.

Grzankowski, A., Keeling, G., Shevlin, H., and Street, W. (2025b). Deflating deflationism: A critical perspective on debunking arguments against llm mentality.

Ha, D. and Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*.

Harnad, S. (1989). Minds, machines and searle. *Journal of Experimental and Theoretical Artificial Intelligence*, 1(1):5–25.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, Berlin, 2 edition.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M., Ali, M. S., Yang, Y., and Zhou, Y. (2017). Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*. https://doi.org/10.48550/arXiv.1712.00409.

Hewitt, J. and Manning, C. D. (2019). A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4129–4138.

Hills, A. (2016). Understanding why. *Noûs*, 50(4):661–688.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Kamienny, P.-A., d'Ascoli, S., Lample, G., and Charton, F. (2022). End-to-end symbolic regression with transformers. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models.

Kepler, J. (1992). *Johannes Kepler: New Astronomy*. Cambridge University Press, Cambridge. Contributor: Owen Gingerich, English translation of Kepler's 1609 *Astronomia nova*.

Kistermann, F. W. (1985). Abridged multiplication: The architecture of wilhelm schickard's calculating machine of 1623. *Vistas in Astronomy*, 28:347–353.

Kitcher, P. (1981). Explanatory unification. *Philosophy of Science*, 48(4):507–531.

Kuhn, T. S. (1962). *The Structure of Scientific Revolutions*. University of Chicago Press, Chicago.

Kumar, A., Clune, J., Lehman, J., and Stanley, K. O. (2025). Questioning representational optimism in deep learning: The fractured entangled representation hypothesis.

Kvanvig, J. L. (2009). The value of understanding. In Haddock, A., Millar, A., and Pritchard, D., editors, *Epistemic Value*, pages 95–111. Oxford University Press.

Ladyman, J. and Ross, D. (2007). *Every Thing Must Go: Metaphysics Naturalized*. Oxford University Press, New York.

Lakatos, I. (1978). *The Methodology of Scientific Research Programmes*, volume 1 of *Philosophical Papers*. Cambridge University Press, Cambridge.

LeCun, Y. (2022). A path toward autonomous machine intelligence.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, K. et al. (2023a). Do large language models have a world model? *arXiv preprint*

arXiv:2303.15447.

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., and Wattenberg, M. (2023b). Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*. Also available as arXiv preprint arXiv:2210.13382.

Li, M. and Vitányi, P. M. B. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, New York, 3rd edition.

Liu, Z., Kitouni, O., Nolte, N., Michaud, E. J., Tegmark, M., and Williams, M. (2022). Towards understanding grokking: an effective theory of representation learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, pages 163–169. ACM.

Lycan, W. G. (1990). Mental content in linguistic form. *Philosophical Studies*, 58(1-2):147–54.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Mayer, J., Khairy, K., and Howard, J. (2010). Drawing an elephant with four complex parameters. *American Journal of Physics*, 78(6):648–649.

Millikan, R. G. (1984). *Language, Thought, and Other Biological Categories: New Foundations for Realism*. MIT Press, Cambridge, MA.

Millière, R. and Buckner, C. (2024a). A philosophical introduction to language models – part i: Continuity with classic debates.

Millière, R. and Buckner, C. (2024b). A philosophical introduction to language models - part ii: The way forward.

Mitzenmacher, M. (2003). A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251.

Morgan, M. S. and Morrison, M., editors (1999). *Models as Mediators: Perspectives on Natural and Social Science*. Ideas in Context. Cambridge University Press, Cambridge.

Nagel, E. (1961). *The Structure of Science: Problems in the Logic of Scientific Explanation*. Harcourt, Brace & World, New York.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. (2021). Deep double descent: where bigger models and more data hurt*. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003.

Nanda, N. (2023). Actually, othello-gpt has a linear emergent world model.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. (2023). Progress measures for grokking via mechanistic interpretability. In *Proceedings of the International Conference on Learning Representations*. Spotlight paper.

NASA Jet Propulsion Laboratory (2019). Planetary physical parameters. NASA JPL Solar System Dynamics Group. Lists Mars orbital eccentricity (0.0934) and sidereal period (686.98 days).

Neander, K. (1995). Misrepresenting and malfunctioning. *Philosophical Studies*,

79(2):109–141.

Nikankin, I., Mikolov, T., et al. (2025). Arithmetic without algorithms: Learning exact computation with neural networks. *arXiv preprint arXiv:2501.XXXXX*.

Norelli, M. F., Votsis, I., and Williamson, J. (forthcoming). The interplay of data, models, and theories in machine learning. *Philosophy of Science*, pages 1–16.

Olah, C., Cammarata, N., Schubert, L., Goh, G., and Petrov, M. (2020). Zoom in: An introduction to circuits. *Distill*.

Olah, C., Mordvintsev, A., and Schubert, L. (2018). The building blocks of interpretability. *Distill*.

OpenAI, :, Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Jozefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petrov, M., d. O. Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F., and Zhang, S. (2019). Dota 2 with large scale deep reinforcement learning.

OpenAI (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Orwell, G. (1949). *Nineteen Eighty-Four*. Secker and Warburg, London.

Parker, W. (2020). Model evaluation: An adequacy-for-purpose view. *Philosophy of Science*, 87(3):457–477.

Parr, T., Pezzulo, G., and Friston, K. J. (2022). *Active Inference: The Free Energy Principle in Mind, Brain, and Behavior*. The MIT Press, Cambridge, MA.

Pearl, J. (2018). Theoretical impediments to machine learning with seven sparks from the causal revolution. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 3, New York, NY, USA. Association for Computing Machinery.

Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, Cambridge, MA.

Petersen, B. K., Larma, M. L., Mundhenk, T. N., Santiago, C. P., Kim, S. K., and Kim, J. T. (2021). Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations (ICLR)*.

Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21:1112–1130.

Polanyi, M. (1958). *Personal Knowledge: Towards a Post-Critical Philosophy*. University of Chicago Press, Chicago.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Prasetya, Y. (2022). ANNs and unifying explanations: Reply to Erasmus, Brunet, and Fisher. *Philosophy & Technology*, 35(2).

Pritchard, D. (2010). Knowledge and understanding. In Pritchard, D., Millar, A., and Haddock, A., editors, *The Nature and Value of Knowledge: Three Investigations*, pages 1–88. Oxford University Press, New York.

Psillos, S. (1999). *Scientific Realism: How Science Tracks Truth*. Routledge, London

and New York.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2847–2854. PMLR.

Railton, P. (1978). A deductive-nomological model of probabilistic explanation. *Philosophy of Science*, 45(2):206–226.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do ImageNet classifiers generalize to imagenet? In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 5389–5400. PMLR.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

Rocks, J. W. and Mehta, P. (2022). Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models. *Phys. Rev. Res.*, 4:013201.

Rocktäschel, T. and Riedel, S. (2017). End-to-end differentiable proving. In *Advances in Neural Information Processing Systems*, volume 30.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rudin, C. (2019). Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Ryle, G. (1949). *The Concept of Mind*. Hutchinson, London.

Saharia, C., Chan, W., Saxena, S., Lit, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Gontijo-Lopes, R., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. (2022). Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Said, K. (2024). Tully–fisher relation. In Di Valentino, E. and Brout, D., editors, *The Hubble Constant Tension*, pages 219–233. Springer Nature Singapore.

Salmon, W. (1984). *Scientific Explanation and the Causal Structure of the World*. Princeton University Press, Princeton, NJ.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. (2019). Analysing mathematical

reasoning abilities of neural models. *International Conference on Learning Representations*.

Schaffner, K. F. (1967). Approaches to reduction. *Philosophy of Science*, 34(2):137–147.

Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. (2020). Improving robustness against common corruptions by covariate shift adaptation.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634.

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–457.

Seck, F. (2005). Schickard, wilhelm. In *Neue Deutsche Biographie*, volume 22, page 727. Duncker & Humblot, Berlin.

Seeley, T. D. (1995). *The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies*. Harvard University Press, Cambridge, MA.

Sellars, W. S. (1963). Philosophy and the scientific image of man. In Colodny, R., editor, *Science, Perception, and Reality*, pages 35–78. Humanities Press/Ridgeview.

Shakespeare, W. (1997). *Othello*. Arden Shakespeare. Bloomsbury, London. Edited by E. A. J. Honigmann.

Shea, N. (2007). Content and its vehicles in connectionist systems. *Mind & Language*, 22(3):246–269.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT Press, 2 edition.

Strevens, M. (2008). *Depth: An Account of Scientific Explanation*. Harvard University Press, Cambridge, MA.

Sullivan, E. (2022). Understanding from machine learning models. *The British Journal for the Philosophy of Science*, 73(1):109–133.

Suppes, P. (1960). A comparison of the meaning and uses of models in mathematics and the empirical sciences. *Synthese*, 12:287–301.

Tamir, M. and Shech, E. (2023). Machine understanding and deep learning representation. *Synthese*, 201.

Thilak, V., Saremi, O., Littwin, E., Paiss, R., Zhai, S., and Susskind, J. (2022). The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*.

Thoren, V. E. (1990). *The Lord of Uraniborg: A Biography of Tycho Brahe*. Cambridge University Press, Cambridge.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Roziere,

B., Goyal, N., Fernandez, A., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Udrescu, S.-M. and Tegmark, M. (2020). Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631.

Valle-Pérez, G., Camargo, C. Q., and Louis, A. A. (2018). Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*.

van Fraassen, B. C. (1980). *The Scientific Image*. Oxford University Press.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS) 30*, pages 5998–6008.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Voelkel, J. R. (2001). *The Composition of Kepler's Astronomia nova*. Princeton University Press, Princeton, NJ.

von Frisch, K. (1967). *The Dance Language and Orientation of Bees*. Harvard University Press, Cambridge, MA. English translation of the original 1965 German edition.

Votsis, I. (2015). Unification: Not just a thing of beauty. *Theoria*, 30(1):97–114.

Votsis, I. (2024). A neuro-symbolic approach to the logic of scientific discovery. In Ippoliti, E., Magnani, L., and Arfini, S., editors, *Model-Based Reasoning, Abductive Cognition, Creativity*, pages 306–330, Cham. Springer Nature Switzerland.

Wallace, D. (2022). Stating structural realism: Mathematics-first approaches to physics and metaphysics. *Philosophical Perspectives*, 36:345–378.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022). Emergent abilities of large language models.

Wei, J., Zhang, Y., Zhang, L. Y., Ding, M., Chen, C., Ong, K.-L., Zhang, J., and Xiang, Y. (2024). Memorization in deep learning: A survey.

Woodward, J. (2003). *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, New York.

Worrall, J. (1989). Structural realism: The best of both worlds? *Dialectica*, 43(1–2):99–124.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017*.

Zhang, Y. (2024). Causal abstraction in model interpretability: A compact survey.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley Press.