Why Unification Is Conditional in Deep Learning

David Peter Wallis Freeborn
Northeastern University London

Abstract

Deep learning systems often understand domains in a fragmented way, using disconnected internal components that work locally rather than a single broadly applicable structure. I call this the Fractured Understanding Hypothesis. I explain it through a conditionality thesis: fragmentation is the default because deep networks can fit data in many ways, and training usually finds patchwork solutions first, with little pressure to unify once prediction is good enough. Unification can occur, but typically only when simpler solutions are strongly favored and the domain contains discoverable regularity. A modular arithmetic case illustrates this transition from memorization to symmetry-respecting algorithmic understanding.

1 Introduction

Suppose that an advanced mathematics undergraduate is preparing for an arithmetic test. She might memorize a large collection of individual sums: $7+5=12$, $13+9=22$, and so on. Given enough effort, she could answer many questions correctly. But we would not say she understands addition. She has acquired a patchwork of isolated facts, each adequate on its own, but with no connection between them: knowing that $7+5=12$ tells her nothing about what $8+5$ might be. Compare this with a student who grasps the general rule: line up the digits, carry when needed, and the procedure works for any pair of numbers. This student has fewer things to remember and broader competence. Her knowledge is *unified*: one procedure covers all cases. And compare both with a student who understands *why* carrying works, deriving the addition algorithm from properties of the place-value number system. Each stage represents a deeper, more unified understanding of the same domain.

Preprint prepared for PSA 2026 Please cite the published version if available.

Deep learning systems face an analogous situation. Trained on data from a target domain, they build internal models that can range from patchworks of disconnected, region-specific components, each handling a portion of the input space on its own terms, to genuinely unified representations in which a single structure governs the whole domain. The question of whether these systems "understand" their targets has generated considerable debate. There are good structural reasons to expect that deep learning systems will often settle for the former, fragmented kind of model. Yet there is also evidence that, under the right conditions, they are capable of discovering genuinely unified structure. Making sense of when and why each outcome obtains is the task of this paper.

This produces a characteristic pattern: deep learning systems outperform human experts on some difficult tasks while failing badly on others that humans find easy, forming what Dell'Acqua et al. (2023) call a "jagged frontier" of competence. The philosophical literature has increasingly recognized that the more salient question is not whether these systems understand, full stop, but what kinds of understanding they possess, to what degree, and through what internal mechanisms (Millière and Buckner, 2024; Grzankowski, 2024). Sullivan (2022) identifies "link uncertainty," the evidential gap between model and target, as what determines whether machine learning supports genuine understanding. Tamir and Shech (2023) propose indicators of machine understanding that naturally come apart across different systems and tasks. Most recently, Beckmann and Queloz (2026) offer a tiered framework distinguishing three 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 genuinely understanding-like and others not.

These contributions tell us what kinds of understanding deep learning systems exhibit and what evidence would support attributing it. But they do not provide a unified account of why fragmentation is so common. I argue that this question has a structural answer. Deep learning systems are powerful but permissive learners that can fit data in many ways. A patchwork of region-specific solutions is typically found early, and once the training objective is satisfied, this patchwork becomes a stable resting point. Unified understanding can emerge, but generally only under specific enabling conditions. I call the descriptive observation that deep learning understanding is typically fractured the *Fractured Understanding Hypothesis* (FUH). The paper's more important contribution is the *conditionality thesis* (CT): an explanation of why the FUH holds and when it does not.

To be precise about the contrast at stake: by *unified understanding* I mean a state in which the system's internal model is organized around a simple, broadly

applicable structure that governs the whole domain, so that competence on one case is connected to competence on others through shared principles. By *fragmented understanding* I mean a state in which the system's internal model consists of disconnected, region-specific components, each locally adequate but collectively incoherent, like the memorizing student rather than the one who grasps the rule.

The argument engages a standing dispute over unification in machine learning, between Prasetya (2022) and Erasmus and Brunet (2022), though the paper's contribution is not limited to that exchange. The broader aim is to explain a structural pattern and identify its philosophical consequences.

2 The Unification Dispute

Philosophers of science widely agree that understanding goes beyond predictive success. To understand a phenomenon is to produce correct predictions *for the right reasons*: from a model that tracks genuine regularities and supports generalization to novel cases (de Regt, 2017; Strevens, 2008; Woodward, 2003).

A prominent tradition connects understanding to unification. On Philip Kitcher's (1981, 1989) influential account, 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. The guiding thought is compression: a good theory brings many cases under few principles. Newton's laws unify the motion of cannonballs, planets, and tides under a single framework. The power of such a theory lies not in the accuracy of any individual prediction, but in the fact that one set of principles generates all of them. Votsis (2015) develops a complementary point that will prove useful below: a unified hypothesis is 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, what Votsis calls a "monstrous" theory, achieves no such connection. This notion of monstrosity gives us a precise way to characterize patchwork understanding: a system whose separate responses to different inputs are confirmationally disconnected, each justified by its own training data and nothing else, is monstrous in exactly Votsis's sense.

This framework speaks directly to a recent debate about whether artificial neural networks (hereafter ANNs, that is, computational systems loosely inspired by biological neural networks and trained by adjusting connection weights) are amenable to unifying explanation. In their original paper, Erasmus et al. (2021) argued that four standard philosophical accounts of explanation (deductive-nomological, inductive-

statistical, causal-mechanical, and new-mechanist) all apply straightforwardly to ANNs, regardless of their complexity. Prasetya (2022) pointed out that they had overlooked a fifth account: Kitcher's unificationist model. And on this account, complexity *does* matter.

Prasetya's argument targets Kitcher's machinery directly. On Kitcher's view, an explanation is a derivation that belongs to a set of general reasoning patterns that collectively unify a body of accepted beliefs. The best such set achieves the optimal trade-off between using as few general patterns as possible and covering as many phenomena as possible. To explain an ANN's output, one must trace specific parameter values through the network's computation. But a network with millions of parameters generates derivations of enormous complexity, each applying to essentially one trained model. Where Newtonian mechanics derives the behavior of many different systems from one pattern of reasoning, neural network explanation threatens to require a distinct pattern for every model. Prasetya concludes that, on the unificationist account, ANNs are not explainable.

Erasmus and Brunet (2022) reply on two fronts. First, they draw a distinction Prasetya's argument passes over: between being an explanation at all and being a *maximally* unifying explanation. Even if neural-network derivations belong to a less unifying set than the best available, they do not cease to be explanations; they merely advance understanding to a lesser degree. Second, they argue that the analytic methods researchers use to investigate neural networks (for example, techniques for identifying which input features drive a network's decisions, or for approximating a complex network with a simpler model) are themselves unifying, because the same methods apply across many different networks.

Both sides capture something important but incomplete. Prasetya is right that a typical trained network, with millions of individually tuned parameters each contributing in its own way, does not implement a small stock of reusable principles. Its internal model consists of many locally distinct components, exactly the kind of proliferation that works against unification. Erasmus and Brunet are right that this does not settle the matter, because some networks *do* discover simple, general internal structures that apply uniformly across their whole domain.

But neither side identifies *when* a network achieves genuine internal unification and when it settles for patchwork. Both treat neural networks as if the answer must be the same for all of them. The more interesting question is what determines the outcome in any given case. To answer that, we need to look at how these systems actually build their models.

3 Why Patchwork Comes Cheap

A deep learning system learns by adjusting a parameterized function to fit data from a target domain (Goodfellow et al., 2016). All of the system's learned information is encoded in this function and its parameters.

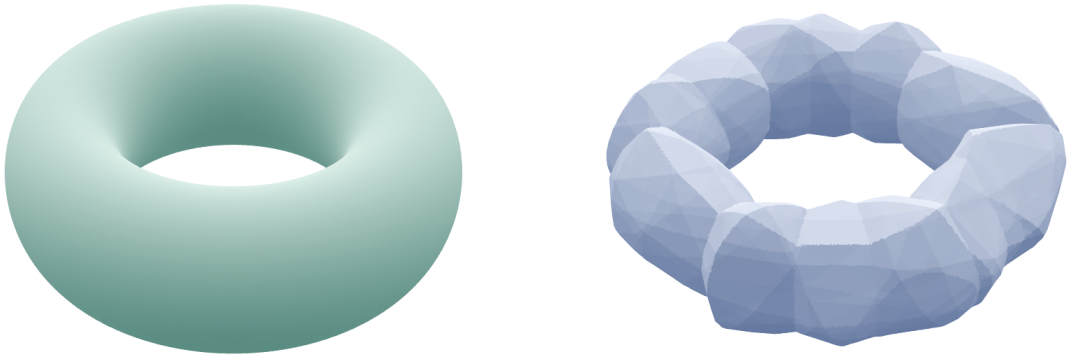
To make this concrete, consider the most familiar architecture. Feed-forward networks with ReLU activations (the most common building block in deep learning) partition the input space into many small regions. Within each region, the network computes a simple linear transformation. The overall function is a continuous surface assembled from many locally flat pieces, like a geodesic dome built from planar facets, approximating a smooth target through a patchwork of straight-edged tiles (Balestriero and Baraniuk, 2018). The approximation can be highly accurate: given enough pieces, any smooth surface can be approximated to arbitrary precision. But such a patchwork need not capture the structure of the function it approximates. A surface tiled from a thousand flat facets may closely match a sphere, yet encode nothing about the sphere's rotational symmetry or constant curvature.

Figure 1 makes this vivid. A small neural network was trained to learn the shape of a torus (a doughnut-shaped surface) from local data points. The left panel shows the smooth ground-truth surface; the right panel shows what the network actually learned. The learned surface is visibly assembled from many small flat facets, stitched together into a continuous whole. It captures the global topology of the torus, including the hole through the middle, yet the surface itself is built entirely from locally planar pieces. This is the spline theory of deep learning in action: the network approximates a smooth, curved target by assembling a patchwork of locally flat regions.

This specific architecture illustrates a more general phenomenon. Deep learning systems generally, including transformers, recurrent networks, and other architectures, share a crucial property: they have enough capacity to fit training data in many different ways, some involving globally coherent structure and others consisting of locally adequate but globally disconnected components of the internal model. I will call the hypothesis class *permissive*: it admits both unified and patchwork solutions.

The argument that patchwork is not merely available but preferentially found rests on three claims.

Availability. A system with sufficient capacity can fit training data by assigning separate internal resources to different portions of the input space, without those resources needing to cohere. Zhang et al. (2017) demonstrated this vividly: modern neural networks can perfectly memorize completely random labels, fitting pure noise to zero error, while still generalizing well on structured data with the same architecture.



(a) Ground truth torus surface. (b) Neural network's learned isosurface.

Figure 1: A neural network trained to approximate a smooth toroidal surface. The learned isosurface (right) is assembled from piecewise-linear patches, yet reproduces the global topology of the target (left), including the genus-one hole. This illustrates how deep learning systems build their internal models: by stitching together many locally flat pieces. Reproduced from Freeborn (2026).

The capacity that enables genuine pattern extraction also enables rote memorization and every intermediate form of local adequacy. Patchwork solutions are abundant in the space of possible models.

Discoverability. The standard training procedure (gradient-based optimization) works by making small, incremental adjustments to the function. This naturally reaches locally adequate solutions before globally unified ones. Fitting each region of the input space separately requires only that each component handle its own neighborhood; discovering that many components instantiate the same underlying regularity is a different and harder kind of search. Geirhos et al. (2020) show that networks routinely acquire what they call "shortcuts": the internal model latches onto features that happen to be predictive in the training data (such as image texture or background context) rather than features that reflect the deeper structure of the task (such as object shape or causal relations). These shortcuts are not random errors. They are exactly the kind of locally adequate, region-specific solution that incremental optimization finds naturally.

Stability. Most crucially, once the training objective has been satisfied by a patchwork solution, the standard training process provides little or no signal pushing toward further reorganization. A patchwork that correctly labels every training example is, from the training objective's perspective, just as good as a globally unified

model that does the same. The system is not stuck in a suboptimal state. It has found a perfectly adequate solution; the solution just happens to be achieved through patchwork rather than through broadly applicable structure. Moving to a more unified solution requires pressure toward simplicity that the basic training objective does not supply. Patchwork is not just easy to find; it is a stable resting point, meaning that optimization will remain there indefinitely unless additional forces (such as regularization) push it onward (Power et al., 2022).

The Minimum Description Length framework (Grünwald, 2007) offers a useful way to formalize what is at stake. Think of a learning system as trying to communicate its data as efficiently as possible. It can either transmit the raw data point by point, or it can first transmit a rule and then transmit only the discrepancies between the rule and the actual data. Genuine understanding corresponds to finding a rule that compresses the data: the rule plus the discrepancies is shorter than the raw data alone. Memorization is the case where no compression occurs. Patchwork solutions achieve compression within small regions but miss the broader regularities that would permit much greater compression across the full domain.

Taken together, availability, discoverability, and stability yield a structural account of why fragmented understanding is the default. These three claims operate at increasing levels of abstraction. At the mathematical level, the permissive hypothesis class makes many local fits available. At the optimization level, gradient-based training reaches these before globally unified solutions. At what we might call the *training dynamics* level, once reached, patchwork solutions are stable because the training objective is already satisfied: the loss function provides no gradient signal toward further reorganization. At the epistemic level, the resulting models can achieve impressive predictions without internalizing robust structural understanding. At the philosophical level, this means that unification, in the sense Kitcher describes, is conditional on factors beyond mere model capacity: it requires exogenous pressure toward compression and a target domain whose regularities are amenable to simple, broadly applicable representation. This bears directly on the dispute discussed in Section 2.

4 Structure Discovery in Modular Arithmetic

If patchwork comes cheap, can genuine unification emerge? It can. To see this, consider a case that is simple enough to analyze completely but rich enough to be philosophically instructive.

Modular arithmetic is ordinary addition with a twist: after adding two numbers,

you keep only the remainder when dividing by some fixed number m. For example, with $m = 12$ (as on a clock face), $10 + 5 = 3$, because 15 divided by 12 leaves remainder 3. The rule is perfectly general: for any two inputs, the correct answer depends only on their sum modulo m. But discovering this rule from examples is not trivial, because individual input-output pairs do not "look alike" in any obvious way. The pair $(10, 5) \to 3$ does not resemble $(1, 2) \to 3$ in input space, even though both instantiate the same modular sum. There is no geometric notion of "nearby" that would help a system interpolate from known pairs to unknown ones. Generalizing requires discovering structure that cuts across the entire domain.

Power et al. (2022) and Nanda et al. (2023) trained a transformer (a widely used type of neural network architecture) on this task. The system receives two numbers as input and must predict the correct remainder. It is trained on a random fraction of all possible input pairs, with the rest held out for testing. Each number is represented by its own distinct token, and the system selects its answer by choosing the highest-scoring output.

The training process reveals a remarkable two-phase pattern. In the first phase, the system rapidly achieves near-perfect accuracy on the training pairs while performing no better than chance on the held-out pairs. It has memorized the specific examples without extracting any general rule. This is patchwork in its purest form: each training pair is handled by its own dedicated internal response, and success on one pair tells us nothing about success on another.

Then, after a prolonged period of continued training well past the point where training accuracy has already reached perfection, test accuracy suddenly shoots up to near-perfect levels. Power et al. (2022) call this phenomenon "grokking." The same architecture, on the same data, has reorganized its internal representations from a collection of memorized associations into a simple, broadly applicable model of modular addition.

How do we know this second phase constitutes genuine unification rather than just more efficient memorization? The answer comes from mechanistic analysis of the network's internal computations. For each possible answer, the model assigns a score to every input pair. This scoring pattern can be decomposed using Fourier analysis (a mathematical technique for identifying periodic or symmetric structure). A memorizing network would show a diffuse, unstructured pattern, with no particular organization to the scores. A network that had internalized the modular rule would show a sharply concentrated pattern, organized around the algebraic symmetry of the target: the fact that the correct answer depends on the sum of the inputs modulo m.

The post-transition network shows exactly this concentration (Nanda et al., 2023). Nanda et al. further identify the specific computational circuit responsible: an

algorithm that converts addition into composition of rotations, a simple structure that applies uniformly to every input pair in the domain. Before the transition, the system needed a separate internal response for each memorized pair. After the transition, one computational structure handles all m^2 cases. The internal model has shifted from many disconnected components to a single reusable principle.

In the unificationist terms of Section 2, this transition is naturally read as a dramatic compression. The analogy with Kitcher's framework is not exact, since the system does not literally instantiate derivations from a body of accepted beliefs, but the structural parallel is substantive. The pre-transition internal model is closer to what Votsis calls "monstrous": its components are confirmationally disconnected, each region-specific sub-model supported by its own data and confirming nothing about the rest. The post-transition internal model is closer to organic: knowing how the system handles one input pair confirms how it will handle others, because the same computational structure applies throughout. It is the system's *internal model* that has been unified, and the components of that model that were previously fractured.

What enabled this transition? Two factors are clearly identifiable; a third is plausible but not directly established by this case. First, the training process included weight decay, a form of regularization that penalizes large parameter values and thereby biases the system toward simpler solutions (Arpit et al., 2017; Liu et al., 2022). This provided sustained pressure toward compression. Without it, the system remains indefinitely in the memorization regime (Power et al., 2022), exactly the stability mechanism described in Section 3: the patchwork was already adequate, and only exogenous pressure dislodged it. Second, the target domain possesses a genuine simple regularity, the group-theoretic structure of modular addition, that falls within what this architecture can represent. Not every target domain will have such regularity. Third, and more tentatively, the way the inputs are represented (each number getting its own distinct token) preserves the discrete algebraic structure of the target, plausibly making the relevant regularity available for discovery. Whether a less structure-preserving representation would have prevented the transition is a reasonable conjecture but one the present case does not directly establish.

5 Why This Case Matters

Modular arithmetic is a deliberately stylized domain: finite, algebraically clean, and mechanistically transparent. A skeptic might reasonably ask why it tells us anything about deep learning more generally.

The answer is that the case functions as a *model organism*: a system chosen

not for typicality but for the clarity with which it isolates a structural phenomenon that is much harder to observe in messier domains. In real-world deep learning, the distinction between locally adequate patchwork competence and genuinely unified internal structure is extremely difficult to identify, because we lack the mechanistic tools to tell them apart. Modular arithmetic is valuable precisely because the Fourier analysis and circuit identification make the distinction sharp and unmistakable. The point is not that most deep learning tasks resemble modular addition. The point is that the forces identified in Section 3, the availability, discoverability, and stability of patchwork solutions, operate in those tasks too, even when we cannot see the resulting internal structure as clearly.

A particularly instructive contrast arises within the same cognitive domain. When large language models encounter arithmetic as a marginal part of their training (predicting the next word across a vast text corpus), they do not learn the underlying rules. Instead, as Nikankin et al. (2025) show through careful mechanistic analysis, they acquire what they call a "bag of heuristics": a collection of disconnected, region-specific components of the internal model, such as individual neurons that activate when an operand falls within a certain numerical range. The conditionality thesis predicts exactly this. The training objective is satisfied by heuristic adequacy; there is no compression pressure specific to arithmetic; and the way numbers are represented (broken into subword pieces) fragments numerical structure rather than preserving it. The same domain that yields unified understanding under dedicated training with regularization yields persistent patchwork when those enabling conditions are absent.

The connection to more familiar deep-learning pathologies is direct. Shortcut learning (Geirhos et al., 2020), in which networks latch onto locally predictive but structurally shallow cues, is an instance of patchwork stability: the shortcuts are adequate for the training distribution, so there is no pressure to seek deeper structure. Brittle generalization under distributional shift is the epistemic consequence of patchwork: competence that does not extend beyond the conditions under which each region-specific component of the internal model was fitted. In a richer domain, Li et al. (2023) demonstrate that a transformer trained on sequences of moves in the board game Othello acquires an internal representation of the board state, verified through causal interventions, despite never being shown the board. This provides further evidence that internal structure discovery is not confined to toy algebraic tasks.

Let me now state the conditionality thesis more explicitly. In the modular arithmetic case, two enabling conditions were clearly present: (i) sustained compression pressure via weight decay, which dislodged the system from its stable patchwork resting point, and (ii) the existence of a genuine simple regularity in the target domain that

the architecture could represent. These two conditions directly counteracted the forces identified in Section 3. Compression pressure overcame *stability* by providing a gradient signal even after training loss reached zero. The existence of discoverable structure meant that a more unified solution was available for the system to find, addressing the *availability* of unified alternatives. The general form of the CT is therefore: unified understanding generally requires conditions that counteract the availability, discoverability, and stability of patchwork solutions. I do not claim these two conditions are exhaustive; the representational interface is plausibly a third factor, and there may be others. But the modular arithmetic case illustrates the logic clearly: where the enabling conditions were present, unification emerged; where they were absent, as in the LLM arithmetic case, patchwork persisted.

The CT provides a framework for the Erasmus-Prasetya dispute, though I would describe it as a framework rather than a resolution. Prasetya is right that typical trained networks implement something close to a monstrous patchwork. Erasmus and Brunet are right that interpretability can reveal genuinely unifying structure. What the CT adds is a principled account of when to expect each outcome: unified internal structure where compression pressure is sustained and the target domain has discoverable simple regularities; patchwork otherwise.

Scope and Qualifications

The argument identifies a tendency, not a universal law. Some architectures reduce the availability of patchwork solutions by building structural constraints into the system: equivariant networks encode physical symmetries, Hamiltonian neural networks enforce conservation laws. The CT predicts that such architectures should more readily achieve unified understanding, and this prediction appears empirically borne out (Cranmer et al., 2020). The claim is also not that locally adequate approximation precludes understanding. A collection of local computations *can* track genuine structure if the pieces are coherently organized. The claim is that permissive hypothesis classes make patchwork solutions readily available and dynamically stable, creating a bias toward fragmentation that must be overcome by specific enabling conditions.

6 Implications

The conditionality thesis complements existing accounts of machine understanding. Beckmann and Queloz's (2026) tiered framework classifies the kinds of understanding deep learning systems exhibit; the CT suggests why the distribution across those tiers is so uneven. Systems often remain at lower tiers not because higher-tier understanding

is architecturally impossible, but because lower-tier patchwork is a stable resting point under the standard training objective. Sullivan's link uncertainty framework identifies when *we* can gain understanding from machine learning models; the CT addresses when *the models themselves* achieve the internal unification that would make such understanding robust.

The sharpest philosophical consequence concerns how machine learning models should be evaluated in scientific contexts. The relevant question is not merely whether a model predicts accurately, nor even whether its internal mechanisms are transparent to investigators. It is whether the model has organized its internal computations around simple, broadly applicable structure that supports generalization through shared principles. A model that predicts well on held-out data from the same distribution may be relying on patchwork competence that will collapse under genuinely novel conditions. The CT suggests that the question "does this model understand its domain?" can be given a more precise form: has the system's internal organization undergone the transition from locally adequate patchwork to broadly unified structure, or not?

If patchwork is cheap and unification is conditional, then scaling parameters alone will not overcome fragmentation. A larger patchwork is still a patchwork. This is not a counsel of despair but a claim about where the relevant lever is. The CT may help point toward productive directions: training regimes that sustain compression pressure, architectures that constrain patchwork availability (such as those encoding physical symmetries; Cranmer et al. 2020), and methods for extracting simple descriptions from learned representations (Udrescu and Tegmark, 2020). Each of these can be understood, from the perspective developed here, as intervening on the availability, discoverability, or stability of patchwork solutions. More broadly, the conditionality thesis may help point philosophers toward what to look for when assessing whether a deep learning system has achieved the kind of understanding that warrants epistemic trust.

References

Arpit, D., Jastrzebski, S., Ballas, N., et al. (2017). A closer look at memorization in deep networks. In *Proceedings of the 34th ICML*, 233–242.

Balestriero, R. and Baraniuk, R. G. (2018). A spline theory of deep learning. In *Proceedings of the 35th ICML*, 374–383.

Beckmann, P. and Queloz, M. (2026). Mechanistic indicators of understanding in large language models. *Philosophical Studies*, forthcoming.

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., et al. (2020). Discovering symbolic models from deep learning with inductive biases. In *Proceedings of NeurIPS 2020*.

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

Dell'Acqua, F., McFowland III, E., Mollick, E. R., et al. (2023). Navigating the jagged technological frontier. Working Paper 24-013, Harvard Business School.

Erasmus, A., Brunet, T. D. P., and Fisher, E. (2021). What is interpretability? *Philosophy and Technology*, 34:833–865.

Erasmus, A. and Brunet, T. D. P. (2022). Interpretability and unification. *Philosophy and Technology*, 35:42.

Freeborn, D. P. W. (2026). A model of understanding in deep learning systems. *arXiv*:2604.04171. doi:10.48550/arXiv.2604.04171.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., et al. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673.

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

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

Grzankowski, A. (2024). Real sparks of artificial intelligence and the importance of inner interpretability. *Inquiry*, 67:1233–1256.

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

Kitcher, P. (1989). Explanatory unification and the causal structure of the world. In Kitcher, P. and Salmon, W. C., editors, *Scientific Explanation*, pages 410–505. University of Minnesota Press.

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., and Wattenberg, M. (2023). Emergent world representations: Exploring a sequence model trained on a synthetic task. In *Proceedings of ICLR 2023*.

Liu, Z., Kitouni, O., Nolte, N., et al. (2022). Towards understanding grokking. In *Proceedings of NeurIPS 2022*.

Millière, R. and Buckner, C. (2024). A philosophical introduction to language models, Part I. *arXiv*:2401.03910.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. (2023). Progress measures for grokking via mechanistic interpretability. In *Proceedings of ICLR 2023*.

Nikankin, Y., Reusch, A., Mueller, A., and Belinkov, Y. (2025). Arithmetic without algorithms: Language models solve math with a bag of heuristics. In *Proceedings of ICLR 2025*.

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

Prasetya, Y. (2022). ANNs and unifying explanations: Reply to Erasmus, Brunet, and Fisher. *Philosophy and Technology*, 35:43.

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

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

Tamir, M. and Shech, E. (2023). Machine understanding and deep learning representation. *Synthese*, 201:51.

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

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

Woodward, J. (2003). *Making Things Happen*. Oxford University Press.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *Proceedings of ICLR 2017*.