Deep Learning as Method-Learning:

Pragmatic Understanding, Epistemic Strategies and Design-Rules

Phillip Hintikka Kieval and Oscar Westerblad

Abstract

We claim that scientists working with deep learning (DL) models exhibit a form of pragmatic understanding that is not reducible to or dependent on explanation. This pragmatic understanding comprises a set of learned methodological principles that underlie DL model design-choices and secure their reliability. We illustrate this action-oriented pragmatic understanding with a case study of AlphaFold2, highlighting the interplay between background knowledge of a problem and methodological choices involving techniques for constraining how a model learns from data. Building successful models requires pragmatic understanding to apply modelling strategies that encourage the model to learn data patterns that will facilitate reliable generalisation.

Word count: 4,496

1 Introduction

The predictive capabilities of contemporary deep learning (DL) models have generated widespread optimism at the prospect of using machine learning to enhance the march of scientific progress in a diverse range of fields. Yet, DL models are opaque in ways that make them difficult to understand (Creel, 2020; Lipton, 2018; Zerilli, 2022). Their superior predictive accuracy thus appears to come at the cost of central aims of scientific inquiry like explanation and understanding. This raises a concern that, without clear explanations or understanding of how a model behaves, we lack rigorous justification for relying on DL models in our epistemic activities. Perhaps, then, all scientific DL is little more than a sophisticated kludge, "a piece of program or machinery which works up to a point but is very complex, unprincipled in its design, ill-understood, hard to prove complete or sound and therefore having unknown limitations, and hard to maintain or extend" (Clark, 1987: 278).

Recent work by philosophers of science aims to address this challenge by examining how scientists can gain understanding by using DL models (see e.g. Sullivan, 2022; Tamir and Shech, 2023). This work tends to focus on how scientists can use DL models to furnish explanations by establishing an empirical link between model and target phenomena. Such explanations depend on whether scientific evidence supports the connection between a given phenomenon and relationships between features represented by data that a DL model exploits to make its predictions.

Our aim in this paper is to move beyond the narrow focus on explainability in discussions of DL models. On the view we develop, scientists working with DL models exhibit a form of pragmatic understanding that is not reducible to or dependent on explanation. We argue that such pragmatic understanding comprises a set of methodological principles that underlie DL model design-choices and secure their reliability. While modellers themselves do not always make these methodological principles explicit, we argue that DL modelling practices reveal clear epistemic strategies that facilitate their success. We illustrate these strategies with a case study of AlphaFold2, a DL model that predicts the three-dimensional structure of proteins from their amino acid sequences. AlphaFold2 highlights the ways that various design choices depend on an iterative interplay between domain-specific background knowledge of a scientific problem and particular methodological choices involving general-purpose techniques for imposing constraints on how a model learns from data. Building successful models requires pragmatic understanding, in our sense, to apply modelling strategies and techniques that encourage the model to learn data patterns that will facilitate generalisation. This pragmatic understanding provides scientists with principled epistemic grounds for their modelling choices.

2 Pragmatic understanding: strategy, design, and method

While discussions of DL models tend to focus on explainability and explanatory understanding, we want to explore the notion of pragmatic understanding – or understanding how to do something. De Regt and Dieks (2005) introduced pragmatic understanding in terms of the effective use of intelligible theories to achieve explanatory understanding. But this notion has also been deployed to characterise a non-explanatory notion of understanding that arises in contexts wherein opacity precludes intelligibility and explanation. Lenhard (2006, 2009), for instance, develops an account of pragmatic understanding involved in computer simulation modelling that does not relate to the use of intelligible theories or to the aim of explanation. This is because the kinds of simulations Lenhard studies are opaque, cobbled together by a complex array of instrumental modelling choices. Instead, Lenhard argues simulation modellers achieve a pragmatic form of use-based acquaintance with simulation models that allows them to develop design-rules, make predictions, and build practical devices

(Lenhard, 2006: 608-609). On this view, pragmatic understanding is action-oriented and can develop "even when the dynamics [of a system] have not been grasped in theoretical terms" (Lenhard, 2009: 171).

By developing design-rules that allow scientists to build devices that help them achieve their epistemic and practical ends, scientists exhibit a form of understanding that is not necessarily theoretical or explanatory. By building and using these models, scientists can develop their learned familiarity with them into epistemic strategies (Knuuttila and Merz, 2009: 158) for dealing practically with systems of interest. Such epistemic strategies underpin the development of procedures and practices that help scientists achieve their aims. In a slogan, scientists pragmatically understand things because they know how to build them (Dretske, 1994). When dealing with complex, black-box systems for which suitable theories or explanations are not forthcoming, we may be better off trying to develop the kinds of design-rules and epistemic strategies that are characteristic of pragmatic understanding.

We suggest that this action-oriented form of pragmatic understanding should be thought of in terms of *method-learning*. Method-learning captures a particular form of pragmatic understanding because it concerns the ways in which scientists learn how to achieve their aims effectively and successfully, structuring their activities in relation to their ends. There is understanding in how the scientists do things successfully, by developing design-rules and epistemic strategies based on their methodically performed activities. We thus think of a method as capturing a pattern exhibited in an activity that accounts for the general success of that activity. It provides a strategy or procedure that, if followed, helps scientists tend towards success in achieving their aims. A method is, in this sense, "a course of action, or a way of reasoning, in view of an aim" (Cartwright et al., 2023: 18).

By developing fruitful methods, scientists can improve their chances of succeeding in their performance of some activity or improve their capacity to reason towards some end. In this way, methods provide ways of learning about and interacting with a system of interest that are not necessarily tied to theory or explanation. They thereby provide grounds for a form of action-oriented pragmatic understanding in cases where a system may be resistant to understanding through explanation. Crucially, such methods may not always be transparent to the scientists working with them. Rather, they are practically articulated in the patterns of successful performance of activities, structuring those performances so that they tend towards success.

Methods are practically articulated in our successful epistemic activities, even if we are not fully aware that we are relying on those methods. The success of these epistemic activities and subsequent method-learning can provide us with principled reasons for relying on DL models even in the absence of explanation or theory. We return to this idea in more detail in Section 4. To first demonstrate our case for this form of pragmatic understanding, we look at the case of AlphaFold2, before returning

to some broader lessons for the epistemology of science.

3 AlphaFold2: a case study of design-rules for deep learning

We characterise deep learning in science as an instrumental practice that aims to achieve coherence through the practical articulation of stable design-rules for building reliable models. Design-rules for deep learning take shape in the identification of specific inductive biases associated with robust performance on certain kinds of learning problems. An inductive bias refers to any preference or constraint over the range of functions a model can learn. Inductive biases ensure that a model tends to learn some patterns rather than others. Pragmatic understanding of inductive biases is necessary for building reliable models. Given a finite amount of training data, the only way to muster robust generalisation over new, unseen inputs is to encode a set of preferences and assumptions about the solution we are after (cf. Wolpert, 1996).

There are a range of different ways to encode inductive biases into a DL model. A handful of these methods include architectural constraints, explicit regularisation, implicit regularisation associated with different optimization methods, self-supervised pre-training, or choices of prior distributions in a Bayesian network. The relationship between model design choices and inductive biases is among the most central concerns of contemporary machine learning research. The empirical and theoretical investigation of inductive biases thus marks a core facet of the ongoing refinement of design-rules for building effective DL models.

In what follows, we use AlphaFold2 as a case study highlighting the practical articulation of instrumental design-rules. AlphaFold2 is a DL model designed by a group of AI researchers at Google DeepMind that accurately predicts the three-dimensional (3D) structure of folded proteins from their amino acid sequence (Jumper et al., 2021). Proteins start out as linear chains of amino acids called polypeptides. These chains fold spontaneously into complex 3D shapes, known as tertiary structures, which determine protein function. The folding process involves local interactions forming secondary structures, such as alpha-helices and beta-sheets along with variable side chains, before finally settling into a stable tertiary structure. Despite vast knowledge of amino acid sequences, experimental mapping of protein structures remains limited, particularly in determining tertiary structures essential for understanding protein function and designing interventions.

AlphaFold2's startling success on one of biology's most intractable problems thus catapulted the model into headlines following its decisive victory at CASP14. Here we discuss several design decisions that are key to AlphaFold2's success, including the adoption of a modified self-attention mechanism, the choice of attention function, and the implementation of 3D-equivariant update operations. These choices all depend on relating background knowledge of a target system to general-purpose inductive

biases that place desirable constraints on learning.

3.1 Self-attention for learning co-evolutionary correlations

One critical factor in AlphaFold2's success lies in the distinctive strengths of the transformer architecture. First introduced by Vaswani et al. (2017), transformer-based models are responsible for the most dramatic advances in contemporary natural language processing and generative AI. These models are built around a mechanism known as self-attention. In short, the transformer uses self-attention to learn what parts of an input are likely to be most important for predicting the next word in a sentence.

Self-attention first arose as a technical solution to the challenge of learning to track long-range semantic and syntactic dependencies found in a body of text. Self-attention allows the network to evaluate how each word in a sequence of text informs the meaning of every other word in the sequence. Transformers do this using three types of activation patterns called keys, queries, and values. Queries represent the term whose meaning we want to consider in view of the context. Keys provide information about all the other words in the input that could influence the meaning of the query term. Values represent how the meaning of the query term would change based on each key. These keys, queries, and values are organised into matrices called Q, K, and V, and the network then computes an attention function over these matrices. Each output sequence is a linear combination of the values weighted by an attention matrix computed from the attention function. This attention matrix helps weigh the importance of different words in the sequence when understanding the meaning of a particular word.

In a transformer, there is an adjustable parameter associated with each query-key-value triplet in an attention layer. This allows the model to learn independently which query-key interactions between words tend to have the most significant impact on the meaning of an input sequence irrespective of their location in that sequence. Before training, the model treats all potential interactions as equally relevant. Transformers thus differ from other popular architectures like convolutional neural networks that are biassed towards local interactions in a fixed region of the input space. In effect, transformers excel at learning to track long-range dependencies in sequence data because they do not incorporate prior inductive biases towards interactions of a particular distance or scale.

AlphaFold2 uses self-attention to take advantage of the biological principles underlying a common bioinformatics technique known as multiple sequence alignment (MSA). MSA helps detect correlations in protein sequences by exploiting the evolutionary conservation of structure over sequence mutations. Picture a scenario where a folded protein contains a positively charged lysine and a negatively charged glutamate close together. The Coulombic force between these amino acids contributes to

the stability of the protein's overall structure. If the lysine mutates to become negatively charged, it puts pressure on the glutamate to also mutate into a positively charged state to preserve stability. MSA assembles a collection of related sequences to identify these sorts of correlations. The basic idea is that correlations between amino acids that are far apart along the polypeptide chain reveal possible contact points on the folded protein. These contact points are crucial predictors of the 3D protein structure. Long-range sequence dependencies are thus instrumental in predicting tertiary structures. Hence, modellers could leverage self-attention as an effective strategy for extracting predictive patterns from polypeptide chains.

AlphaFold2's main trunk, which Jumper et al. (2021) calls the Evoformer, consists of two modified transformers running in parallel that operate over both an MSA generated from an input sequence (hereafter, the MSA transformer) and a pairwise matrix of intra-sequence residue correlations (hereafter, the pair transformer). Both transformers share information about learned dependencies in their respective inputs by including additional update steps that create pathways for information to flow between them.

The motivation for this architectural choice lies in pragmatic understanding of self-attention as a generic mechanism for learning long-range sequential dependencies. The DeepMind team used their background knowledge of prediction methods in bioinformatics to identify long-range sequence dependencies as important predictive markers of protein structure, which enabled them to make an informed decision about the right architectural constraints to use.

3.2 Triangle attention as a geometric constraint

As mentioned above, AlphaFold2's pair transformer computes self-attention on a pairwise matrix of residues along an input sequence. The idea is to produce a summary description of the protein structure in terms of pairwise distances between individual amino acids. This pairwise representation then feeds into the structure module, which predicts the final 3D tertiary structure. The pair transformer involves a distinctive update pattern that enforces crucial constraints on how AlphaFold2 learns this summary description.

The pair transformer performs a modified version of axial attention. Transformer-based architectures require that modellers specify an attention function. Vaswani et al. (2017: 3-4), for instance, used scaled dot-product attention.Dot-product attention is optimised for relatively small, one-dimensional inputs like chunks of text. But its memory use explodes quadratically with the size of input. This computational burden creates significant practical difficulties when scaling up the workable input size. Multidimensional inputs like images only make this problem worse.

To work around these practical constraints, Ho et al. (2019) developed "axial atten-

tion" to deal with high-dimensional tensor data such as images. Axial attention works by computing self-attention along a single axis of an input tensor. So, if the input is a two-dimensional array, axial attention computes self-attention over a single column or row along that array. Stacking these axial attention layers allows a transformer to compute self-attention over the entire input while significantly reducing computational complexity. Axial attention thus carries an affinity for grid-like data structures, making it a natural choice for computing self-attention over stacks of amino acid sequences in an MSA.

Jumper et al. (2021) arranged the axial attention updates in the pair transformer in terms of triangle shaped graphs involving three different nodes, which they aptly name "triangle attention" (Jumper et al., 2021: 586). For a given triplet of residues ijk, this operation treats each residue as a node in a graph and the distance between them as the corresponding edge. Each graph is arranged like a triangle with residues i, j, and k as the vertices. Triangle attention also adds an extra logit bias term to the axial attention function, which serves to compensate for the 'missing' third edge of the triangle. So, when i is the query node, triangle attention updates the edge ij with the values given by all other possible edges ik_n (where k_n is an arbitrary residue in the sequence) that share the same starting node modulated by the bias term b_{jk} representing the third edge of the triangle (see Suppl. Material for Jumper et al., 2021: 18).

There's a simple yet ingenious idea underlying this seemingly quite complex version of self-attention. The 'third edge' bias term is a clever way of enforcing a much needed geometric constraint on learning of pairwise residue distances. For a pairwise summary description of amino acid residues to be representable as a single, coherent 3D structure, those pairwise distances need to consistently obey basic principles of Euclidean geometry. In effect, triangle attention acts an inductive bias that forces pairwise distances learned by the pair transformer to obey the triangle inequality on distances, which says that the sum of any two sides of a triangle is greater than or equal to the length of the third side.

This design choice depended on the modellers' ability to identify a necessary constraint on learning given their epistemic aim and to devise strategies for implementing that constraint in a working model. Triangle attention thus illustrates how moddellers exhibit pragmatic understanding.

3.3 3D-Equivariant Updates for Learning Structure

Alphafold2 implements end-to-end structure prediction by passing the outputs of both MSA and pair transformers through what they call the "structure module," which maps the output of the Evoformer stack to concrete 3D atomic coordinates. The structure module operates on a graph representation of the 3D polypeptide backbone, treat-

ing it as a series of N_{res} independent rotations and translations with respect to a global frame of reference. In short, the model treats each amino acid along the backbone as a triangle-shaped rigid body. All of these bodies start out squished together at the origin point in space—the authors call this "black-hole initialization" (Suppl. Material for Jumper et al., 2021: 23). The structure module then performs a series of parameterized transformations and rotations on these rigid bodies. We can think of these transformations as affine matrices—a mathematical method of representing translations and rotations of points in a field—that define Euclidean transformations from a residue's local frame to coordinates in the global frame.

The structure model learns to parameterize these transformations. It does this with another variant of self-attention that DeepMind calls "invariant point attention (IPA)" (Jumper et al., 2021: 587). IPA augments each query-key-value triplet with 3D point coordinates in a way that achieves two things. First, IPA produces those 3D points within the local frame of each residue such that the final value becomes invariant to global rotations and translations. Second, the 3D queries and keys impose a strong spatial inductive bias on the attention function that helps iteratively refine the structure. After each attention operation, the structure module then computes an update to the translation and rotation of each backbone frame.

This series of computations makes each block of the structure module an equivariant operation. This shift from invariant operations to equivariant ones is subtle but important. The outputs of invariant operations are insensitive to arbitrary shifts in the input. Equivariant operations, on the other hand, keep track of arbitrary shifts. Equivariance is thus a form of symmetry that preserves shifts in the input with corresponding shifts in the output. This is important for two reasons. First, it reduces the space of possible solutions in the global frame. Since the model operate over 3D coordinates, even a minute rotation results in unique computational object. But molecules like polypeptides don't have a unique orientation in space. Rotating a structure doesn't change its identity. Equivariance ensures that 3D coordinates overdetermine structure predictions in this way. Second, equivariance is a necessary property for performing valid operations over 3D rotations and translations. Shifts in the input space need to result in corresponding shifts in the output. Enforcing equivariance is thus crucial since the structure module needs to keep track of each preceding rotation and transformation in the frame at each subsequent processing block.

So, IPA involves using attention to learn the optimal parameters of a series of 3D equivariant transformations on rigid bodies against a frame of reference. IPA thus enforces necessary geometrical constraints without imposing any physical laws. Herein lies the real expressive power of the model. Whereas prior work used top-down, rule-based physics engines to to resolve physical inconsistencies in distance predictions, AlphaFold2 resolves the structure by learning the values of free parameters on geometric transformations directly from data.

There is a clear kind of epistemic strategy at work here. The DeepMind team identified 3D-equivariance as a necessary symmetry and subsequently developed IPA to implement that symmetry in the learning process. This demonstrates the kind of methodical design choice that characterises pragmatic understanding.

4 Deep learning design principles as a method: being principled by being methodical

With our case detailed, we now want to draw the discussion back to some general epistemological lessons. Our aim is to get clearer on precisely how to evaluate the understanding possessed by scientists working with DL models. As the case above shows, the practical articulation of methods in the activities of scientists working with DL models provide a host of epistemic strategies and procedures that provide pragmatic understanding, making DL model-based practices principled and intelligent. This understanding through method gives modellers principled epistemic grounds for the choices that they make despite the lack of explanatory understanding.

We can begin now to reflect more generally on method-learning and its epistemic significance. Useful methods are important for securing the reliability of our knowledge claims (Cartwright et al., 2023). But we want to take a step further, arguing that methods also provide understanding. Our suggestion draws on Dewey's (1916; 1938) wide-ranging theorising about scientific inquiry and the role of method in inquiry

Dewey argues that developing fruitful methods is an important part of scientific knowledge production. Such methods arise "organically" out of our successful activities, rather than being formulated prior to inquiry. As Dewey argues, in past successful activities "[w]e see that a certain way of acting and a certain consequence are connected, but we do not see how they are. [...] We [must] analyze to see just what lies between so as to bind together cause and effect, activity and consequence". Only when we investigate the relationship between our actions and their consequences do we discover and make explicit "the thought implied in cut and try experience" (Dewey, 1916: 145; our italics). These connections provide us with a way of uncovering the understanding implicit in successful activities. The patterns that we uncover that are implied in successful epistemic activities provide grounds for understanding the connections between what we do and what the outcome of our actions are. These patterns provide an understanding of how to perform such activities successfully, if we were to explain why they did succeed or if we wanted to try to perform them again or improve them. In making these patterns explicit, we codify particular norms for how to do things by producing methodological principles for the construction of fruitful practices.

The kind of pragmatic understanding exhibited by scientists working with DL models successfully provides them with standards for being principled, even when they do

not live up to the standards of theoretical explanation. Past successes provide norms for future inquiry:

> We know that some methods of inquiry are better than others in just the same way in which we know that some methods of surgery, farming, road-making, navigating or what-not are better than others. [...] They are the methods which experience up to the present time shows to be the best methods available for achieving certain results, while abstraction of these methods does supply a (relative) norm or standard for further undertakings. (Dewey, 1938: 108)

Dewey's views are thus not only helpful in providing a connection between method and understanding, but also in further explicating how pragmatic understanding can assuage the worry that DL models may be mere kludges, unprincipled and ill-understood. Past successful activities practically articulate methods from which scientists can extract principles to guide them in learning what to do.

Past successes provide the grounds for the articulation of methods. These methods establish design-principles and epistemic strategies for future work and model-building, which enable scientists to be principled by being methodical. They can gain a method-based form of pragmatic understanding that allows them to progress in their tasks even when explanations are lacking. This is not to say that theoretical understanding would not be useful in these cases, but that in the current state of play, pragmatic understanding is a very valuable epistemic good that allows us to treat DL models as more than mere kludges.

One might worry that this kind of pragmatic understanding is only valuable for the purpose of achieving theoretical or explanatory understanding at the end of properly conducted inquiry (see Parker 2014 and Lenhard 2019, ch. 4). However, in light of the rise of epistemically opaque and complex tools like those discussed in our case above, we cannot take for granted that pragmatic understanding is only valuable to such an end. In fact, it may be the best we can achieve in these cases, providing a form of genuine understanding despite the opacity of the systems being used and studied

To this, one may object that, as Sullivan (2022) argues, opacity is simply not an obstacle to explanatory understanding. Instead, Sullivan argues that it is link uncertainty—"a lack of scientific and empirical evidence supporting the link that connects the model to the target phenomenon"—that precludes explanatory understanding. If so, then perhaps there's no need to look towards pragmatic understanding in the first place. Even so, we think that it would be fruitful to reorientate discussions of understanding in DL-based science away from explanatory understanding to pragmatic understanding. We think that pragmatic understanding performs a different and perhaps more epistemically fundamental function that is not fulfilled by explanatory understanding

alone. Towards the end of her paper, Sullivan discusses cases where "there are no explanatory questions the model can answer or [where the] models are mere predictive tools" (2022, 129). We think that method-learning plays a crucial epistemic role in grounding the choices and design-decisions that scientists make when constructing DL models in such cases. Hence, we claim that such cases can involve pragmatic understanding even where there is no explanatory understanding to be had.

5 Conclusion

In an influential paper entitled "Understanding Deep Learning Requires Rethinking Generalization," Zhang et al. (2017) present several experiments which they use to argue that traditional, theoretical approaches from statistical learning theory fail to explain why overparameterized DL models generalise as well as they do in practice. In the years since, a large breadth of work has taken to using empirical studies, "designing systematic and principled experiments" that aim to understand how DL models achieve their remarkable results (Zhang et al., 2021: 114). This work displaces mathematical approaches that aim to establish guaranteed upper bounds on generalisation error with empirical analyses of inductive biases (Goyal and Bengio, 2022), effective capacity for learning rules (Zhang et al., 2017; Zhou et al., 2023), implicit regularisation (Neyshabur et al., 2015), and performance over data distribution shifts (De Silva et al., 2022; Hupkes et al., 2022; Singh et al., 2021). We suggest that this empirical trend reveals a discipline grappling with a self-conscious attempt to make explicit the methods and design principles already implicit in their practice. DL models exhibit interesting and unexpected behaviour. Through "cut and try" experience, practitioners have learned various epistemic strategies and design-rules for harnessing these behaviours in service of their epistemic aims. These strategies involve pragmatic understanding marked by a kind of method that is practically articulated through patterns of successful modelling activities. The empirical work just mentioned aims to systemise this body of pragmatic understanding into a well-grounded theory that explains the capacities of DL models. We think this nicely illustrates the sense in which pragmatic understanding as method-learning is a significant epistemic achievement that can even underwrite future growth in explanatory understanding and theoretical knowledge.

References

Cartwright, N., Hardie, J., Montuschi, E., Soleiman, M., and Thresher, A. C. (2023). *The Tangle of Science: Reliability Beyond Method, Rigour, and Objectivity*. Oxford University Press, Oxford.

Clark, A. (1987). The Kludge in the Machine. *Mind & Language*, 2(4):277–300.

Creel, K. A. (2020). Transparency in complex computational systems. *Philosophy of Science*, 87(4):568–589.

De Regt, H. W. and Dieks, D. (2005). A Contextual Approach to Scientific Understanding. *Synthese*, 144(1):137–170.

De Silva, A., Ramesh, R., Priebe, C. E., Chaudhari, P., and Vogelstein, J. T. (2022). The Value of Out-of-Distribution Data. pages 1–24.

Dewey, J. (1916). *Democracy and Education: An Introduction to the Philosophy of Education*. The Free Press, New York.

Dewey, J. (1938). Logic: The Theory of Inquiry. In Boydston, J. A., editor, *The Later Works,of John Dewey 1925-1953, Volume 12*. Southern Illinois University Press, Carbondale, IL.

Dretske, F. (1994). If You Can't Make One, You Don't Know How It Works. *Midwest Studies in Philosophy*, 19:468–482.

Goyal, A. and Bengio, Y. (2022). Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2266):1–49.

Ho, J., Kalchbrenner, N., Weissenborn, D., and Salimans, T. (2019). Axial Attention in Multidimensional Transformers. pages 1–11.

Hupkes, D., Giulianelli, M., Dankers, V., Artetxe, M., Elazar, Y., Pimentel, T., Christodoulopoulos, C., Lasri, K., Saphra, N., Sinclair, A., Ulmer, D., Schottmann, F., Batsuren, K., Sun, K., Sinha, K., Khalatbari, L., Ryskina, M., Frieske, R., Cotterell, R., and Jin, Z. (2022). State-of-the-art generalisation research in NLP: A taxonomy and review. 5(October).

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Knuuttila, T. and Merz, M. (2009). Understanding by Modeling:. In *Scientific Understanding: Philosophical Perspectives*, pages 146–168. University of Pittsburgh Press, Pittsburgh, PA.

Lenhard, J. (2006). Surprised by a nanowire: Simulation, control, and understanding. *Philosophy of Science*, 73(5):605–616.

Lenhard, J. (2009). The great deluge. In *Scientific Understanding: Philosophical Perspectives*, pages 169–186. University of Pittsburgh Press, Pittsburgh, PA.

Lipton, Z. C. (2018). The mythos of model interpretability. *Communications of the ACM*, 61(10):35–43.

Neyshabur, B., Tomioka, R., and Srebro, N. (2015). In search of the real inductive bias: On the role of implicit regularization in deep learning. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, pages 1–9.

Singh, H., Joshi, S., Doshi-Velez, F., and Lakkaraju, H. (2021). Learning Under Adversarial and Interventional Shifts. pages 1–19.

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

Tamir, M. and Shech, E. (2023). Understanding from Deep Learning Models in Context. In Lawler, I., Khalifa, K., and Shech, E., editors, *Scientific Understanding and Representation: Modeling in the Physical Sciences*, pages 323–340. Routledge.

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 Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 47–82. Curran Associates, Inc.

Wolpert, D. H. (1996). The Lack of a Priori Distinctions between Learning Algorithms. *Neural Computation*, 8(7):1341–1390.

Zerilli, J. (2022). Explaining Machine Learning Decisions. *Philosophy of Science*, 89(1):1–19.

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

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J., Bengio, S., and Nakkiran, P. (2023). What Algorithms can Transformers Learn? A Study in Length Generalization. pages 1–39.