Isotropic Deep Learning: You Should Consider Your (Foundational) Biases

George Bird
Department of Computer Science
& Department of Physics and Astronomy
University of Manchester
george.bird@postgrad.manchester.ac.uk

May 1, 2025

Abstract

This position paper explores an alternative mathematical formulation, '*Isotropic Deep Learning*', by analysing the implications of current functional forms in deep learning. Modern networks almost universally rely on foundational forms respecting discrete permutation symmetry. However, this is an underappreciated *choice* in form, argued to introduce unrecognised biases without suitable alternatives. Initially, this discrete symmetry observation is promoted to a continuous rotation defined framework, then broadened to primitive sets defined by various other symmetries. This constitutes a new symmetry-led design-axis: rather than enforcing it through model design, which transfers symmetry through the structure, it studies how foundational form symmetries inherently *act* on and *interact* within general architectures — one objective is a systematic approach to the consequences of network symmetry breaking in addition to symmetry making when emerging from the primitive-level. In addition, determining whether non-trivial expressibility is contingent on which function symmetries are preserved moreover broken. The goal is to expose and leverage unintended biases by deducing principles applicable in broader contexts for beneficial computation. Proposed is a systematic reformulation of *all* foundational primitives into classes that respect particular groups, and to determine the resultant implications. This constitutes an inverted ontology framework where general symmetries are situated definitionally prior to neurons, rather than a permutation symmetry being deduced from them. This design axis motivates reselection of compositions upwards, as they underpin current constructions and may enable new models contingent on alternative foundations. Hence, the paper advocates for a distinctly bottom-up reformulation aiming to deduce general principles for broad leverage.

This is motivated by prior work demonstrating that current functional forms influence activation distributions: discrete symmetries in functions induce similar discrete structure in embedded representations through training. Thus, geometric artefacts can arise in learned representations solely due to human-imposed design choices rather than task-driven necessity. Therefore, the prevailing choice is shown to carry unappreciated and unintended task-agnostic biases. Moreover, there appears to be no compelling a priori justification for why such representations or functional forms are universally desirable; this paper hypothesises three testable pathologies of the current formulation with significant connections to mechanistic interpretability. Hence, this motivates the construction and analysis of alternative foundational primitives, aiming to alter geometric constraints on representations and improve performance. The underlying inductive biases of the isotropic approach may constitute a preferable default which could be adopted if a wide array of suitable and well-performing functions are developed. A variety of preliminary functions are proposed, including new activation functions, normalisers, and operations, and an audit is provided across various primitives in use. The symmetry-principled construction is then generalised, enabling a broad class of group-defined reformulations across primitives, positing a new foundational design axis with distinct inductive biases. Thus, Isotropic Deep Learning becomes just *one case study* among such parallel implementations for all models.

This initial group-theoretic generalisation of primitives is systematically extended upwards to encompass their hierarchical compositions, motivating its applicability across all scales in architectures. This yields an initial three generations of symmetry strength for categorisation in the framework. This extension recovers Geometric Deep Learning as the strongest generation when composing functions, ensuring model-scale compliance with the symmetry constraint derived from data for specialist applications. This substantially contrasts with this paper's enquiry, diverging through a bottom-up philosophy starting from primitives rather than working recursively top-down from model constraints. This establishes a further role for symmetry emerging within deep learning. This taxonomic formalism also encompasses the Parameter Symmetry approach as a distinct compositional case, studying the consequences of computational equivalences under reparameterisations deduced from current permutation-like primitives. In contrast, this work redefines primitives through a symmetry-led design-axis *and* investigating the ramifications more broadly, not just restricted to discrete parameter degeneracies. Hence, this "Taxonomic Deep Learning" approach reveals all three to be distinct special cases, characteristic of various compositional scales and strengths — a unification of contemporary approaches to symmetry in an intuitive, hierarchical, and complementary formalism. This may facilitate a better, comprehensive comparison and exploration of their interplay, while clarifying further regimes that may remain to be considered. Encouraged is a systematic audit into the influence of symmetry generally, but particularly the reformulation and comparison of various group-defined primitive sets. From this, the study of downstream phenomena can proceed after a primitive algebra is fixed. This can span from determining representation biases, reassessing theorems contingent on prior primitives, optimisation, performance, and diverse new model architectures.

1 Introduction

Elementwise functional forms singularly dominate contemporary deep learning [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. This is particularly evident in, but not limited to, activation functions sometimes referred to as 'ridge' [11] activation functions. Activation functions are often displayed in univariate form [12, 13, 14], generally characterised by the form shown in *Eqn.* 1, with σ being a placeholder activation function, e.g., ReLU ($f(x) = \max(0, x)$) [15], Tanh ($f(x) = \tanh(x)$), etc.

$$f : \mathbb{R} \to \mathbb{R}, \quad x \mapsto f(x) = \sigma(x) \tag{1}$$

However, this display choice obfuscates a crucial (standard) basis dependence. This dependence is made explicit in *Eqn.* 2, which displays the multivariate functional form of a given activation function. This should be considered a more implementation-faithful form[1]. This reveals the functional form's usually hidden $\hat{e}_i$ basis dependence. The multivariate form is depicted for an n-neuron layer, with activation vector $\vec{x} \in \mathbb{R}^n$. This *standard* basis dependence is arbitrary and appears to be largely a historical precedent, rather than a problem-aligned, intentional inductive bias. This is discussed further in *App.* F.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}) = \sum_{i=1}^{n} \sigma(\vec{x} \cdot \hat{e}_i)\, \hat{e}_i \tag{2}$$

Due to this basis dependence, non-linear transformations differ angularly in effect [16, 17]. Therefore, this will be termed an *anisotropic function*, indicating this rotational asymmetry. Particularly, it could be termed a *standard-anisotropic function*, indicating its dependence on the standard basis. Due to the pervasive use of these functional forms, including activation functions, normalisers, initialisers, regularisers, optimisers, architectures, operations, and gradient clipping, amongst others, contemporary deep learning as a paradigm may consequently be termed a form of '*anisotropic deep learning*'. Despite its implications and prevalence, this *choice* of basis-dependent anisotropic form appears underappreciated and incidental in the development of most contemporary models.

Anisotropic forms have largely become an unquestioned default, approaching an axiomatic-like definition for the general field rather than a considered choice. Hence, re-evaluating their impact and then systematically reformulating this foundational aspect of modern deep learning, with potentially wide-reaching consequences, is suggested to constitute distinct approaches, such as '*Isotropic Deep Learning*' — effectively based upon differing foundational, axiomatic-like, primitive definitions. This is emphasised by such choices underlying all downstream compositions, including those in models. It should be determined whether their respective phenomena, theoretical results, and various consequences are contingent upon these foundational choices. It is arguably the general composition of such primitives, including parameterised maps, which defines the current ontology of deep learning — contrasting it with other machine learning approaches.

The asymmetry in current non-linear transforms is usually about the standard (Kronecker) basis vectors, and frequently their negative, $\{+\hat{e}_i, -\hat{e}_i\}\ \forall i \in [0, 1, \cdots, n]$ for n width layers, and these are equal in privilege. This is often due to their elementwise application. Therefore, it can be said to distinguish the standard basis — a '*distinguished basis*'[2]. This basis-dependence is often overlooked in consequence, and this work argues that it acts as an implicit inductive bias for representational geometry; therefore, it must be evaluated. For example, the standard basis' activation space distortions are visible in *Fig.* 1 showing the mapping of elementwise-$\tanh$ on a variety of test shapes.

Crucially, one can consider this *choice* of functional form to break a *continuous* rotational-like symmetry, and reduce it to a *discrete* rotational-like symmetry (alongside some specific mirrors). The latter is specifically referred to as a permutation (S_n) symmetry of the standard basis, and the former is an orthogonal ($O(n)$) symmetry about the origin. In effect, if a function is treated in its multivariate form, in the current formulation of deep learning, it is equivariant to a permutation of the components of its vector decomposed in the standard basis — this also defines the notion and individuality of neurons.

For (a representation of) an element of the permutation group, notated in shorthand by $\mathbf{P} \in \mathcal{S}_n$, the following equivariance relation holds: $f(\mathbf{P}\vec{x}) = \mathbf{P}f(\vec{x})$. However, permutation symmetry is a subgroup of the orthogonal symmetry proposed: $\mathcal{S}_n \subset \mathrm{O}(n)$ — therefore, this discrete permutation symmetry could be considered a broken continuous orthogonal symmetry[3]. Further details on the categorisation and nuance of such symmetries are discussed further in *Sec.* 5.2.

Non-linearities are usually pivotal to the network's ability to achieve a desired computation, as seen through the universal approximation theorem's [18] explicit dependence on the form of the activation function [19]. Non-linearities produce differing local transformations, such as stretching, compressing, and generally reshaping a manifold — displayed elegantly in Olah [20][4]. Consequently, the network may be expected to adapt by moving representations to geometries about these distinguished directions, using specific localised mappings to achieve the desired computation, discussed further in *Sec.* 2.2. Hence, an anisotropy about distinguished vectors may be expected to be induced into the activation distribution, through optimisation in general networks. This anisotropic inductive bias on representations appears to be reinforced directly through most functional forms, indirectly through many optimisers, and is inherent in the connectivities of many architectures. Hence, it seems largely systematic to contemporary deep learning through each aspect of this triad — all of which typically share the same underlying and characterising permutation symmetry. Each is hypothesised to contribute to such representational structures.

[1] Softmax has an extra denominator term, but still displays the basis-dependent nature of elementwise forms.

[2] This is suggested as a generalisation from a *'privileged basis'* discussed in Elhage et al. [16]. *'Distinguished directions'* may better reflect how representations can be encouraged or deterred in alignment about these directions; whereas *'privileged basis'* would suggest a greater preference for alignment. The term 'basis' will often be retained despite the set of '*distinguished vectors*' potentially being under-/over complete for spanning the activation space, as demonstrated by Bird [17].

[3] The mirror transform in the group $\mathrm{O}(n)$ appear to result in no change in (single-argument) functional forms from those defined through the pure rotation group $\mathrm{SO}(n)$. Hence, $\mathrm{O}(n)$ is used since Isotropic deep learning automatically respects this larger group.

[4] Olah also stated disillusionment with the elementwise form for other reasons in this article.

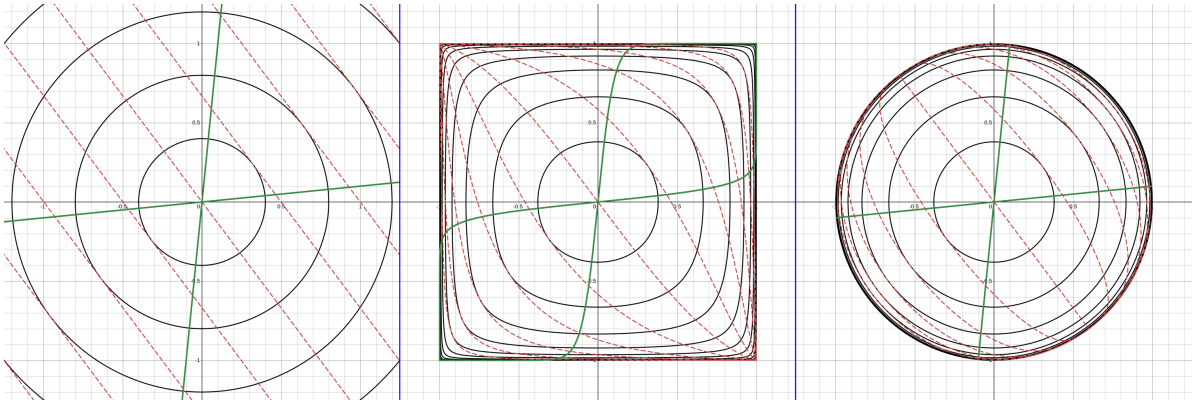

Figure 1: Left shows a 2-dimensional plane, $\mathbb{R}^2$, populated with various shapes: black concentric circles, green lines through the origin, red parallel lines and in faint black the standard (cartesian) coordinate axes $\hat{e}_1$ and $\hat{e}_2$ (which remain untransformed). If this space is then imaged through elementwise-$\tanh$, the individual pointwise coordinates making up the shapes are passed through the standard $\tanh$ activation. The resultant shapes are shown in the centre plot. The rightmost plot is similar to the centre plot, but for the so-called isotropic-$\tanh$ presented in *Sec:* 3.1. One can see that the objects in the centre plot are distorted around the basis directions, whilst in the right-most plot, they are not distorted due to the basis directions. For example, the green lines are significantly curved towards the corners of the boundary. An interactive demonstration of these functions is available here.

The direct effect has been empirically demonstrated in activation functions [17]: training results in the discrete symmetry of the functional forms, inducing a broken symmetry in the activations which transforms with transformations to the distinguished directions of the form. Since these non-linear zones are centred around the distinguished directions, the embedded representations are expected to adopt advantageous angular arrangements with respect to the arbitrarily imposed geometry of the distinguished basis. For example, they appear to move towards the non-linearities' extremums, aligned, anti-aligned, or other geometries [16], through training [17]. This may correspond to a local, dense, sparse coding [21] or superposition [16], respectively. This indicates that such directions mark out an absolute reference structure about which representations are observably shaped. Therefore, the network has adapted its representations through optimisation due to the properties of the foundational functional form choices present.

These general representational biases are entirely distinct considerations from enforcing a specific end-to-end symmetry in a network. In such cases, the form is leveraged to preserve a data structure through the network for a targeted application, where representations predictably transform or remain unchanged with respect to the group. In these circumstances, the symmetries do not 'act' on the network internally as a bias in the general manner being suggested in this work — it is these latter considerations that are argued to be unappreciated and unintentional, in universal settings. Thus, a differing approach to symmetry's role in deep learning: *a task-driven versus a function-driven approach*. Due to this separate motivation, this formalism is a tangent line of enquiry, and the group-theoretic reasoning emerged independently in response. It is argued to be an important consideration, regardless of the underlying data structure, and hence its applicability is considered general. Nevertheless, such approaches both draw on group-theoretic roots and can be unified under a formalism presented in *Sec.* 5.2, which may enable beneficial cross-considerations and sharing of tooling at times.

Moreover, this function-driven causal hypothesis aids in explaining the observed tendency of distinguished-direction alignment. This is the hypothesis underlying the encouraged position: **functional forms should be deliberate and carefully considered design choices, with a suitably optimal and minimally harmful *default* since they can induce a representational structure *not required by the task***. Currently, anisotropic primitives appear to induce a human-imposed representational collapse onto the distinguished directions. Hence, this was shown to frequently not be a task-necessitated collapse, but instead a task-agnostic structure induced by function primitives. There appears to be little justification for why this particular form and induced structure is universally desirable, with several key negative implications predicted in *Sec.* 2. Without a priori justification, this inductive bias may be detrimental to computation; therefore, unconstraining the activation is argued to be generally preferable. In addition, the added structure into functional forms, which produces these distinguished directions, may be considered a needless additional assumption for some applications applying deep-learning models.

Throughout the rest of this paper, it is argued that a departure from this anisotropic functional form paradigm towards the isotropic reformulation may be generally preferable as a universal inductive bias, unless supported by task-aligned justification for differing primitive algebra. This paper encourages consideration of these choices when designing a model generally, alongside the usual architectural toolkit. In particular, isotropic choices, initially led by a basis independence principle, are argued to unconstrain the representations into more optimal arrangements for general tasks and architectures, free from imposed discrete structure. Some instances where isotropy may be particularly beneficial, such as the amendments discussed for self-attention, are discussed in *App.* D. However, the development of functions, then models, which suitably leverage isotropy may require substantial time to parallel the existing approach to deep learning in empirical results, since they are based upon a fundamentally differing foundational set of primitives. These will then require downstream verification of the argued optimality. Additionally, many contemporary architectures may be a product of selection over anisotropic primitives and anisotropic benchmarks, whose intrinsic anisotropies may be indicative of. Such outcomes of selection may not pertain

to isotropically defined primitives or benchmarks (or other taxonomies). Therefore, reselection of architectures from a clean-slate foundational approach may be necessitated, but generally productive ideas could be analogised. Hence, due to these considerations, such reformulations may require considerable development before they mature into practical implementations. The understanding of such phenomena, and the potential resultant impact of this, is argued to be a worthwhile exploratory avenue.

The most fundamental addition of this work is that these inductive bias considerations motivate a broader symmetry-unifying construction for functional forms, discussed in *Sec.* 5. This produces a taxonomic class for deep learning extending bottom-up from *sets of* primitives defined through their group structures. Isotropic and Anisotropic deep learning represent just two among the array of possible groups capable of generating functional form branches using the tools presented — a seemingly rich and unexplored proposal. Exploring such alternatives may offer better optimised functional forms beyond those discussed in this paper. Thus, the specific case study of Isotropic deep learning should not detract from the wider scope of primitives defined over the broader symmetry taxonomy and general groups.

This taxonomic approach extends a group-theoretic formalism for considering and imposing symmetry constraints on functional forms, thereby organising them under distinct branches, each generating a complete set of primitives. This taxonomy is organised by group criteria across three degree-tiers/generations and three flavours, discussed further in *Sec.* 5.2. Through these group-defined sets, one can *choose* which primitive class to implement. This is argued to approach an axiom-like choice in ramifications, since it is these primitives which are later composed into all downstream models. Hence, the primitive selected occupies an underivable base choice preceding any model design, and its group-defined form is usually assumed instead of a leveraged design-axis. Therefore, reformulation requires a broad reevaluation, extending from the reselection of models following from primitive changes, investigating each's resultant emergent phenomena, as well as the reanalysis of theorems predicated on the current form for primitive, among numerous further implications. Comparison between choices may be advantageous in yielding more fundamental insights into the innate properties of deep learning.

Such sets of primitives can be constructed a priori for general applications; yet, the taxonomy also indicates how arbitrary graphs can inherently break such symmetries. This symmetry-breaking is argued to be both useful, if influences are leveraged correctly, or perhaps detrimental, if it occurs haphazardly; therefore, establishing such a link is critical, and this paper advocates for it being a careful design choice. This connection is achieved through evaluating automorphisms of an *arbitrary* graph structure. Typically, one would then set the functional form symmetry constraints as a subset of the available automorphisms to fix a primitive class from those available[5]. Hence, generally, forms are applied which remain unbroken by the inherent connectivity. This, in turn, would specify how to typically apply the primitive form, such as channel-wise for convolution, since these connectivities are not inherently symmetry broken by the graph's connectivities. This motivates the expanded set of generation strengths to which group-theoretic constraints can apply. 'Closures' are the broadest functional class, which would typically be chosen and derived from automorphisms of an arbitrary graph and then can be selectively elevated to stronger generations in design for narrower classes. Geometric deep learning's considerations arise when a single group is elevated to the strongest level network-wide, and hence provides a restriction on the subset of architectures and functions which can provide these initial closures. However, it is argued that awareness of the generalised group-theoretic choices and their consequences may be beneficial in universal settings.

Due to the argued pervasive applicability of such a symmetry-formalism approach across foundational primitives, the terms "branch" or "fork" are used, e.g. "Isotropic-fork". These are felt to be appropriate descriptors for the groups of distinct forms produced, as well as downstream architectures, theorems, and phenomena contingent upon them. This indicates that the use of graph-based computation, from the continued use of linear algebra, is preserved; however, all intermediate functions have parallel implementations that respect their new chosen symmetries. These alternative classes of primitives would typically diverge substantially from the form of contemporary functions, and likely their respective models and consequences — warranting a differing subclassification system. Encouraged are differing semi-autonomous subdisciplines of exploration to determine the implications and leveragability of each, a systematic analysis revealing how they may incur different biases through their various reformulations. However, the consistent use of linear algebra and a primitive in a layered structure makes it appropriate to continue grouping them under the "deep learning" heading, rather than a distinct machine learning approach. Nevertheless, it is argued that in most other meaningful ways, the forks may be largely distinct, likely preferring differing architectures, applications, and resultant phenomena (such as interpretability consequences). This highlights a potentially broader ontology for what may be constitute a deep learning system, where these taxonomically-organised and underpinning choices offer axiomatic-like branching within the field.

This axiomatic-like nature is underscored by each symmetry functional form requiring a respective Universal Approximation Theorem due to the current theorems [18, 22, 19] being contingent on the contemporary activation function primitive. This should be undertaken, for each new class of primitives, providing existence proofs for *dense* networks as standard. It is also hypothesised that theoretical efforts may be able to extend this through the group structure, enabling the determination of which symmetry families may yield useful functional forms a priori. This speculation would constitute a more overarching Universal Approximation Theorem, and would likely be a desirable long-term objective in any case. This could be termed a 'Group Universal Approximation Theorem' GU(A)T for discussion purposes. Additionally, the symmetry automorphisms are derivable from arbitrary graphs, including *intrinsic privileged directions*. This extended UAT approach could perhaps be further developed to derive bounds for a given symmetry on any given graph structure (generalising the typical dense network assumptions) — which could be referred to as a 'Group Universal Bound Theory' GU(B)T for discussion purposes. Both remain conjectures and may serve as long-term aspirational objectives, representing a beneficial theoretical direction to be

[5]Then one would choose to implement a beneficial instantiation from a selection of functions which abide by the chosen functional form.

pursued, particularly in terms of the proposed symmetry formalism. This may aid in further narrowing down which groups are suitable for deep learning and enable a better directed search, alongside inductive bias considerations.

Overall, the approach outlined, which also utilises symmetry and a new taxonomic organisation, stems from a group-theoretic formalism as a *definitional* tool for all primitives. *Then* the implications of differing sets of primitive-level algebras are extended upwards in general compositions, considering their respective inductive biases, generalised resultant model architectures, scale-interplays, theorems, and characteristic phenomena. This is distinct from both Geometric Deep Learning's Invariant/Equivariant networks [23, 24, 25, 26, 27] as well as recent observations, and leveraging of Parameter Symmetries [28, 29]. The former is a task-driven end-to-end respect of a particular symmetry, such that the entire model transforms predictably under its action. Hence, it is a model-level consideration that can extend down to achieve this, such as into layer maps for group convolution and further. The latter is a compositional consideration that concerns the computational equivalences under reparameterisation of surrounding affine layers, deduced from contemporary permutation-like activation function algebras. All can be similarly united under the overarching formalism of *Sec.* 5.2. Hence, the taxonomic system, with varying generation strengths, flavours, and scales, can be shown to recover the new and prior approaches to symmetry in deep learning as particular regimes/philosophies of an overarching group-theoretic perspective. This unification also enables novel findings when considering differing compositional regimes and layerwise constructions that have not been researched thus far.

In conclusion, this paper argues that the existence of such a definitional choice for functional forms, and their consequences extending upwards, has remained a substantially underappreciated approach and should be investigated thoroughly with an aim to leverage findings generally. Such choices and their effects are typically obfuscated, neglected, and seldom [30] questioned in general model-design. A basis dependence has resulted in an internal absolute frame that appears to have become ubiquitous throughout most primitives in nearly every model. This may be partly a result of accidental notational oversimplification, suppressing basis factors, enabling the consequences of that form to remain obscure and unquestioned. There is also a decades-long history of successful and practical precedent behind it, which has become entrenched in even hardware alignment, having formed around and potentially having also shaped the wider practice. Additionally, its practicality has so far surfaced minimal apparent tensions in observations[6]. Hence, it is argued that it has become largely an unintended default, as there is a lack of suitable alternative forms, much less primitive sets, in wide circulation and unrecognised consequences arising from the current form. However, a causal link between the current arbitrary basis's transforms and internal representations has recently been empirically demonstrated as significant [17]. An influence on models' internal representations in turn will alter their behaviour and likely downstream performance, where it is hypothesised to display some pathological consequences. These motivate the need for a reconsideration.

Hence, alternative choices and establishing their implications can now be systematically explored and developed. This includes a reselection of instantiations for each form to leverage their symmetries better, which has not been undertaken even when alternative forms have sometimes surfaced. Well-justified and understood decisions can then be drawn from a range of choices. A suitable, minimally detrimental default can also be selected, and specialist choices can be made for particular applications. This culminates in an extended formalism which provides a unifying perspective of several naturally emerging group-theoretic approaches. This has had the effect of demarcating complementary but distinct regimes and scales to consider. Pursuing this may eventually yield cross-disciplinary findings if these disparate approaches are bridged, while other scales and compositions may yield further insights beyond what is currently established. This is argued to be a good motivation for considering this broad and unifying approach.

The following section discusses the hypothesised pathological consequences, which initially motivated such reconsiderations.

2 Predicted Detriments of Anisotropy

This section outlines a non-exhaustive set of predicted pathologies that anisotropic functional forms may introduce. These mainly centre on the role of the activation functions, since this is the area that has been primarily explored thus far. However, similar considerations may be equally applicable to other primitives (particularly quantisation). To the author's knowledge, some of these failure modes are newly characterised phenomena, such as the so-called '*neural refractive problem*'. It may indicate that if *Isotropic Deep Learning* substantially matures, it may form a better default inductive bias unless an alternative is task-necessitated. A further intuition is expressed in *App.* F.1.

2.1 The Neural Refractive Problem

The '*neural refractive problem*' describes how linear and origin-intersecting trajectories of activations may converge or diverge from their initial path after an activation function is applied. This is analogous to a light ray refracting through optically varying media or boundaries.

This phenomenon appears to occur in all anisotropic activation functions examined to date. The 'refraction effect' typically occurs more significantly at larger magnitudes — potentially producing a failure mode under network extrapolation. Neural refraction is demonstrated by curvature of previously straight origin-intersecting lines (in green) in the centre plot of *Fig.* 1, but is absent in the rightmost plot of the same figure. This refraction is a mathematical consequence of the form, but its impact on representations and pathological nature requires validation.

Mathematically, this phenomenon has several representations, a magnitude-varying 'dynamic refraction' shown in *Eqn.* 3 or differentially in *Eqn.* 4. Also defined is a 'static refraction' definition shown in *Eqn.* 5. Geodesic-based constructions may also be defined. These are intended only as provisional formalisms of the phenomenon. These current formalisms are

[6]Except, perhaps, in observations of interpretability phenomena that may emerge from the form's structure.

described for a multivariate activation function $\mathbf{f}$ and vector $\vec{x} = \alpha\hat{x}$ where $\hat{x}$ is a unit vector. This relation may be satisfied for a single direction, a subset of the space or all directions $\hat{x} \in \mathcal{X} \subseteq S^n$. The relations generally show how the activation function alters the direction of its input vector in an anisotropic manner.

$$\exists \hat{x} \in \mathcal{S}^{n-1}, \exists \alpha_1 \neq \alpha_2 > 0 : \frac{\mathbf{f}(\alpha_1 \hat{x})}{\|\mathbf{f}(\alpha_1 \hat{x})\|} \neq \frac{\mathbf{f}(\alpha_2 \hat{x})}{\|\mathbf{f}(\alpha_2 \hat{x})\|} \tag{3}$$

$$\exists \hat{x} \in \mathcal{S}^{n-1}, \exists \alpha_0 : \frac{\partial}{\partial \alpha} \frac{\mathbf{f}(\alpha \hat{x})}{\|\mathbf{f}(\alpha \hat{x})\|}\bigg|_{\alpha_0} \neq \vec{0} \tag{4}$$

$$\exists \hat{x} \in \mathcal{S}^{n-1}, \exists \alpha : \frac{\mathbf{f}(\alpha \hat{x})}{\|\mathbf{f}(\alpha \hat{x})\|} \neq \hat{x} \tag{5}$$

It can be seen that along a straight-line trajectory in direction $\hat{x}$, the result of the activation function is a curved line if dynamically refracted. Therefore, if the linear feature hypothesis is followed, then *every* linear feature, in refracted directions, becomes curved following the activation function. The network may exploit some of this curvature to construct new linear features in the subsequent layers; however, there may be many instances where this curvature is detrimental to established semantics. The network may lose semantic separability, produce magnitude-based semantic inconsistency or produce compensatory maladaptations in later layers. However, due to the non-linear nature of refractions in generalised directions, which continues to be compounded over subsequent layers, the network may struggle to mitigate the effect. Hence, these maladaptations may fail disproportionately for out-of-distribution samples. This may hinder the generalisation performance of the network and indicates a mode which may make representations more susceptible to adversarial attacks. Isotropic choices would resolve the refraction, potentially resulting in fewer such adaptations. An illustrative example of neural refraction is shown in *Fig.* 2.

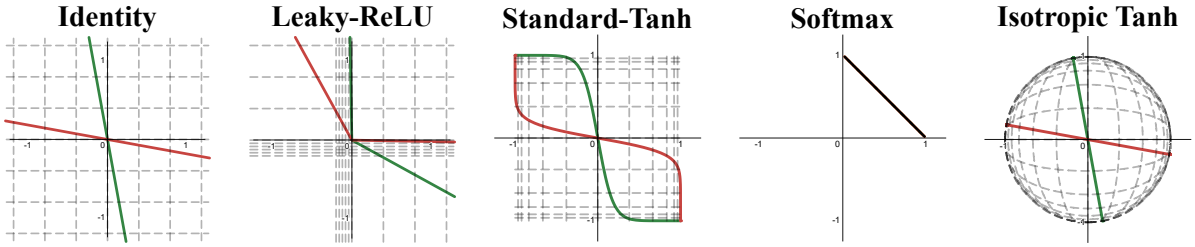

Figure 2: Displays the $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^2$ maps, for the identity map (leftmost), standard Leaky-ReLU (centre-left), standard Tanh (centre), Softmax (centre-right), and isotropic-Tanh (rightmost). These maps transform various objects within the space, including two lines with zero-intercept, shown in red and green, as well as sets of horizontal and vertical lines in pale grey. The three centre plots demonstrate 'neural refraction' in its static form for Leaky-ReLU and its dynamic form for standard Tanh, as well as a more general case for Softmax. The identity plot and isotropic plot do not cause such refractions to these objects.

Postulated to be especially detrimental, in both refraction cases, is the loss of semantic separability. If two distinct trajectories, representing different semantics, are transformed into curves which intersect or converge, then the separability of these concepts is lost or misrepresented. For example, suppose one direction is a linear feature for the presence of a dog in an image, whilst the other is for a horse. In that case, if these activations are of sufficient magnitude where the activation function causes convergence, the identity of the activation's meaning can be misconstrued. This is in addition to the aforementioned deflection of linear trajectories, which may reduce the effectiveness of following linear transforms to effectively separate representations.

The convergence may be particularly consequential for functions such as Sigmoid and Tanh, since large magnitude inputs end up at particular limit points (discussed as trivial representational alignments in Bird [17]). For example, Tanh produces the limit points shown in *Eqn.* 6 when $\hat{x} \cdot \hat{e}_i \neq 0$ for all i. If there exists an i such that $\hat{x} \cdot \hat{e}_i = 0$, then the transformed vector has a 0 in the corresponding index. Therefore, all vectors end up at limit points with sufficient magnitude when using elementwise-Tanh or Sigmoid. A fully-connected layer would typically only effectively separate two such converging directions at a time, which are then further curved by a subsequent activation function.

$$\lim_{\substack{\forall i, \vec{x} \cdot \hat{e}_i \neq 0 \\ \alpha \to \infty}} \mathbf{f}(\alpha \hat{x}) = \sum_{i=1}^{N} \tanh(\alpha \hat{x} \cdot \hat{e}_i)\, \hat{e}_i \approx \sum_{i=1}^{N} \pm \hat{e}_i = (\pm 1, \cdots, \pm 1)^T \tag{6}$$

Consequently, semantic separability is lost for large magnitude representations except for 3^n discrete limit points for Tanh and Sigmoid. Therefore, embedded activations may be expected to align with these limit points. This explains some results empirically observed by Bird [17]. Similarly, ReLU has one distinct limit point, $\vec{0}$, but otherwise an orthant unaffected by neural refraction. It is speculated that this is an additional reason for the success of ReLU, as only a subset of directions experiences the neural refraction phenomenon. Furthermore, this suggests an advantage of Leaky-ReLU: despite featuring static refraction, directions do not become overlapped, so semantic separability is retained. The network may otherwise 'expend' training time on producing robust semantic separability, having a potentially discretising effect on representations. This would be a needless compensatory adaptation, which may lower representational capacity and extend training as a result of inefficiency.

More generally, the dynamic deflection of trajectories may cause semantic ambiguity for the network, where only samples interpolable from training samples are reliably semantically identifiable. Particularly, *the more significant the deflection, the greater the semantic ambiguity may be expected* due to the resultant position of representations becoming unpredictable. Therefore, a magnitude-dependent semantic inconsistency may arise due to such deflections. A deflection function can be a trivial diagnostic measure, defined by *Eqn.* 7 for a particular activation function.

$$\theta\left(\alpha;\hat{x},\mathbf{f}\right)=\arccos\left(\frac{\mathbf{f}\left(\alpha\hat{x}\right)\cdot\hat{x}}{\left\|\mathbf{f}\left(\alpha\hat{x}\right)\right\|}\right) \tag{7}$$

This may result in an additional mode of degraded performance for a network, especially on out-of-training-distribution samples. For example, suppose a linear feature roughly represents the quantity of cows in a field. In that case, the network may fail to extrapolate its function when an anomalous amount of cows are present. This would be due to a considerably larger magnitude of the linear feature, which is typically deflected significantly. Therefore, the deflection is unprecedented and becomes uninterpretable. The activation function would result in a loss of semantic consistency. Consequently, a network seeking to preserve linear features may constrain activation magnitudes through training to regions where the non-linear response is approximately predictable and stable, thereby avoiding the damaging consequences of neural refractions. Moreover, the network may move representations towards locally linear positions, limiting the beneficial transformative properties of the non-linearity.

Current angular anisotropies fundamentally cause the refraction phenomenon. If compression and rarefaction occur in certain angular regions, linear features will be deflected in various ways. A fix for this is to introduce isotropy (or norm-based forms of quasi-isotropy). This is the initial motivation for developing the approach. Isotropy does not prevent compression and rarefaction of activation distributions in general, as a bias can be added to reintroduce these useful phenomena predictably. It is argued that these issues only arise when they affect linear features, not affine ones, in a potentially unpredictable and thus semantically uninterpretable manner. Applying this to all affine features would be restrictive enough to return linear approximations only; whilst shifting the origin of linear features could be considered through a symmetry-broken transformation.

The phenomenon is eliminated from networks by rearranging *Eqn.* 5 shown in *Eqn.* 8, then applying the simplification $\left\|\mathbf{f}\left(\alpha\hat{x}\right)\right\|=\sigma\left(\alpha\right)$ in *Eqn.* 9.

$$\mathbf{f}\left(\alpha\hat{x}\right)=\left\|\mathbf{f}\left(\alpha\hat{x}\right)\right\|\hat{x}' \tag{8}$$

$$\mathbf{f}\left(\alpha\hat{x}\right)=\sigma\left(\alpha\right)\hat{x}' \tag{9}$$

Finally choosing $\hat{x}'=\mathbf{R}\hat{x}$ for isotropy and $\mathbf{R}\hat{x}=\mathrm{I}_n\hat{x}=\hat{x}$ for simplicity, shown in *Eqn.* 10.

$$\mathbf{f}\left(\alpha\hat{x}\right)=\sigma\left(\alpha\right)\hat{x} \tag{10}$$

In standard notation, *Eqn.* 10 can be rewritten into the *functional form for isotropic activation functions* shown in *Eqn.* 11. This should be a piecewise function, defined using the identity at $\vec{x}=\vec{0}$, but this is suppressed for simplicity. Alongside an appropriate smoothness condition on the Jacobian, this ensures the apparent 'singularity' at $\vec{x}=\vec{0}$ is only a *coordinate singularity* present only due to how the functional form is denoted. Future work involves establishing a universal approximation theorem for this functional form, which is currently an ongoing area of research for the author. This form can be generalised to other functional forms in *App.* A, but is discussed briefly below using symmetry equivariance.

$$\mathbf{f}:\mathbb{R}^n\to\mathbb{R}^n,\quad \vec{x}\mapsto\mathbf{f}\left(\vec{x}\right)=\sigma\left(\left\|\vec{x}\right\|\right)\hat{x} \tag{11}$$

The form of *Eqn.* 11 is $\mathcal{O}\left(n\right)$ time for $\mathbb{R}^n$ activation vectors, and only computes the non-linear term *once*, unlike n-computations for the non-linear term in current activation functions. In addition, radial basis functions suffered from a $\mathcal{O}\left(nm\right)$ cost. This bilinear scaling arguably impeded the widespread adoption of this functional form in ever-larger models, displayed in *Eqn.* 12 and *Tab.* 3.1.

$$\mathbf{f}:\mathbb{R}^n\to\mathbb{R}^m,\quad \vec{x}\mapsto\mathbf{f}\left(\vec{x}\right)=\sum_{i=1}^{m}\sigma\left(\left\|\vec{x}-\vec{c}_i\right\|\right)\hat{e}_i \tag{12}$$

Isotropy can be generalised to a result of rotational equivariance of the function, expressed as a condition in *Eqn.* 13. This uses a commutator bracket for convenience, with $\forall\mathbf{R}\in\mathrm{O}\left(n\right)$. This bracket can be used to similarly define the current anisotropic discrete rotational (permutation) paradigm, by using the transform $\forall\mathbf{P}\in\mathcal{S}_n$ instead of the rotation. Connecting forms of machine learning through symmetry is further elaborated on in *Sec.* 5, including the apparant functional form indifference between $\mathrm{O}\left(n\right)$ and $\mathrm{SO}\left(n\right)$. Hence, these constraints constitute an effective definitional tool for generating and categorising all primitives across various taxonomies. The equivariance relation may be recognised as superficially similar to equivariant neural networks, due to an analogous equivariance relation; however, the differences in both implementation and motivations are substantial, and discussed further in *App.* E.1.

$$\left[\mathbf{R},\mathbf{f}\right]=\left(\mathbf{R}\mathbf{f}-\mathbf{f}\mathbf{R}\right)=\vec{0} \tag{13}$$

The relation may be more familiar as $\mathbf{f}\left(\mathbf{R}\vec{x}\right)=\mathbf{R}\mathbf{f}\left(\vec{x}\right)$. This relation only applies to single-argument functions and requires generalising to more circumstances, shown in *App.* A. A similar condition suffices: $\mathbf{f}\left(\mathbf{R}\vec{x}_1,\cdots,\mathbf{R}\vec{x}_N\right)=\mathbf{R}\mathbf{f}\left(\vec{x}_1,\cdots,\vec{x}_N\right)$ for $\mathbf{f}:\bigotimes_N\mathbb{R}^n\to\mathbb{R}^n$.

The 'neural refractive problem' outlines how semantic meanings may become intertwined or ambiguous due to current functional forms skewing linear features in undesirable ways. This is predicted to be especially detrimental for out-of-distribution activations, which are likely to be most deflected and hence most semantically corrupted. Thus, the network's generalisation may then fail in such circumstances. It may be expected that the network produces compensatory adaptations for the phenomenon, which may be narrow in the scope of their corrections. Since neural refraction is a non-linear and anisotropic phenomenon, it cannot be inverted by a single subsequent layer, potentially incurring unnecessary training overhead on producing corrections due to unintended refraction.

2.2 Quantised Representations, Emergence of Linear Features and Semantic Interpolatability

Symmetry-broken functional forms have been shown to induce symmetry-broken representations which transform with the basis [17], which indicates a dependency on the anisotropy and offers an explanation why *approximately* discrete embedding directions are tended towards [31, 16, 17]. In this section, that conjecture of dependence will be made clear. Additionally, it can be hypothesised that because embedded activations are often discretised and meaningful directions may be expected to align with these embeddings, then semantic directions also become quantised. This generally appears to be the case in observations [31, 32, 17]. Reversing this proposed causality would indicate that a continuous rotational symmetry may enable a continuous embedding. Functional forms would not directly induce arbitrary direction-based symmetry breaking in their embeddings through training; such a structure would only emerge from task necessity.

One can start with the prediction of form-induced representational collapse. In this context, representational collapse is the following heuristic: *The induced discretisation of what would otherwise be an approximately smooth continuum of representations as samples drawn over a dataset.* Where *Discretisation is the increasing concentration of representations of clusters through training, until they eventually approach a nearly discrete-like cluster in representation space*. Hence, this is also described as a quantisation, to differentiate it from other representational collapses. Quantisation would be the induced discretisation of an otherwise continuous quantity. At this early stage, until the nature of this predicted phenomenon is suitably understood, this heuristic may be more appropriate than a premature, rigid mathematical definition. The following discussion provides an informal motivation for the prediction of quantisation, followed by a more principled discussion.

One may expect that the angular unevenness of various anisotropic primitives will result in some form of general effect on optimisation. Particularly, such unevenness would likely result in slight preferred directions for embeddings and slightly discouraged directions for embeddings organised around the anisotropic distinguished directions. It is the degree to which this effect may occur which is of interest. For example, in extreme cases, this may result in the absence of representations over discouraged directions and a clustering of representations over encouraged directions. This is the suggested form-induced quantisation into discrete-like clusters. Such a collapse results in informative representational degrees of freedom being suppressed in otherwise approximately connected data. This is suggested to be pathological when arbitrarily imposed through task-agnostic functional form inductive biases. This may be beneficial where redundancy can be suppressed, as discussed in *Sec.* 4; however, the form-induced arbitrary structure may remain detrimental. This extreme discretisation would be very distinct and may aid in detection — which is suggested to have observably already appeared [16, 17]. However, extreme discretisation itself may not be ubiquitous; it is the more general production of task-agnostic 'structure' about these directions which constitutes the general inductive bias, and these are suggested to be indicative of the algebraic symmetries of the forms. Without such initial unevenness, preferential angular regions would not exist, and representations may distribute more 'naturally', perhaps smoothly or be indicative of structure in the dataset, rather than task-agnostic structure due to the choice of primitives.

Practically, this may materialise in numerous modes through optimisation, depending on the function's particular analytical qualities; however, these are suggested to all result from the underlying group structure of the forms. Particularly in primitives defined through discrete group algebras. This may arise regarding both forward and backwards pass considerations.

For example, current anisotropic primitives respect at least a standard-basis permutation symmetry (S_n), which can result in a discrete orthant partitioning of the space in two or more dimensions. A functional form can then be described piecewise through this partitioning. Given a univariate function, f, which is applied elementwise, it can be represented piecewise as two differing functions $f_<$ and $f_\geq$ for the negative and positive semi-definite domains, respectively. When applied elementwise, the representation space's orthants have various combinations of these two functions acting on elements, dependent upon the particular orthant. Several of these orthants are hence analytically equivalent, but rotated, in function. Generally, there are $n+1$ distinct orthants for n-width layers, with a $\binom{m}{n}$ degeneracy for $m \in \{0, 1, \cdots, n\}$. This is indicative of the underlying S_n standard-basis permutation symmetry arising from the elementwise application. Effects on optimisation may then result, where representations shift over different orthants to leverage the differing localised maps for computation. Hence, structure is expected to arise as a consequence of this symmetry partitioning. Hence, all S_n functions are expected to be influenced in this generalised symmetric manner, with specific modes contingent upon each orthant's particular map, yet remaining tied through this underlying construction.

Other permutation-based symmetries in functional forms can be considered. For example, hyperoctahedral B_n, including the standard-basis permutation with sign-flip symmetry, makes all orthants analytically degenerate when correcting for rotation; hence, its effect on representations through optimisation may indicate this. Similar for even-sign flips, which produce two sets of analytically degenerate orthants, if constructed piecewise using $0 \leq \prod_{i=1}^{n} \operatorname{sign}(\vec{x} \cdot \hat{e}_i)$ (These are denoted D_n but are not to be confused with the dihedral group). Other discrete symmetries can result in other partitionings to be considered, such as the simplex-based symmetries in Bird [17]. Such a partitioning does not occur in the continuous-symmetry definition for isotropic primitives under $\mathrm{O}(n)$, so such an aligned structure emerging through optimisation is not expected. This is illustrated in *Fig.* 3 for $3D$ multivariate maps under symmetry, while *Fig.* 2 demonstrates the phenomenon in $2D$ for Leaky-ReLU and

standard Tanh, where the S_n and B_n respective symmetries produce quadrant partitioning of their maps (isotropic-Tanh does not partition in this manner).

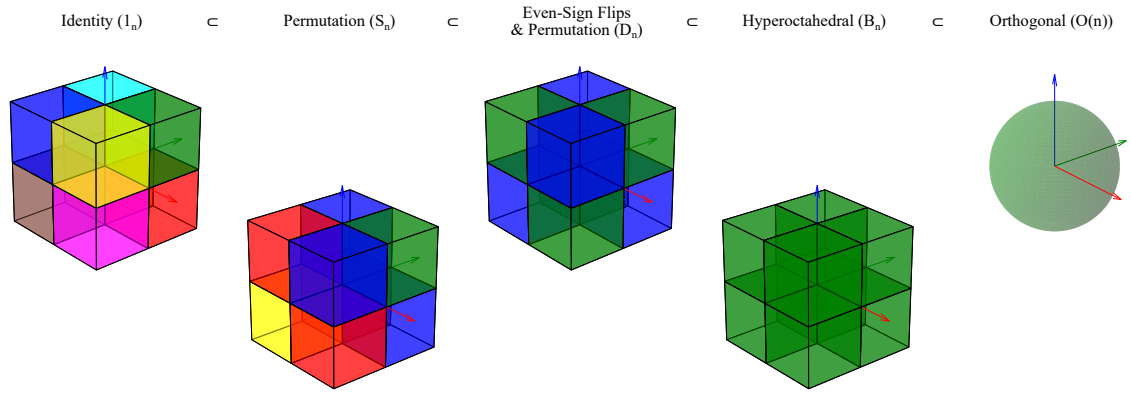

Figure 3: Illustrates the effect of the various symmetries in $3D$ about the standard basis. The standard bases are shown as red, green and blue arrows, with the various octants (orthants in nD) demonstrated for discrete symmetries. Left-to-right shows, the identity symmetry of elementwise functions I_n, the permutation symmetry S_n, the even-sign permutation symmetry D_n, the hyperoctahedral symmetry B_n, and the continuous orthogonal symmetry $O\left(n\right)$ producing an angularly continuous depiction. The colour-shading of the various octants demonstrates which octants are analytically identical under a rotation/permutation of their map. This intuitively shows how different octant regions relate in their maps and may influence the representation space. Particularly, the discrete maps effectively incur an absolute frame for the internal representation spaces, whilst orthogonal maps only incur an absolute origin.

Additionally, there may be a hierarchical interplay of influences on representations from various functional forms. These may interact non-trivially, potentially privileging differing bases, with an overall privilege which may evolve through training. Potentially, accumulation may occur, as suggested, up to a point where an alternative basis becomes privileged and begins to disperse the existing structure; this may result in interesting dynamics, additional phase-changes and steady-state equilibrium behaviour. Whether this occurs is speculation, but it could be explored.

One may also consider the consequences on the associated semantics being represented through these embeddings. Many real-life semantics are continuums: colours, positions and poses of objects, broad morphology, even within a single species or objects. Induced representational collapse onto a single discrete semantic may lose vital nuance and meaningful degrees of freedom. Discretised representations encouraged by functional forms appear to be a poor default inductive bias under these considerations. Without spurious structure added to representations from functions, the quantising bias would vanish, potentially enabling more continuous representation for the task. In this manner, isotropic functions would not prevent discrete semantics, which can be clustered through bias parameters; however, they do not promote discretisation either. Hence, Isotropic deep learning would be well-positioned to enable networks to acquire more naturally distributed representations, driven by the task and free from structure. Therefore, moving towards isotropy is hypothesised to encourage embeddings to be more smoothly distributed and better representative of the task and data. In addition, this is expected to better enable interpolatable semantics for intermediate representations between typically discrete linear features. This may substantially enhance the expressivity and representational capacity of networks — only limited by concept interferences. This may position an Isotropic approach as producing more optimal representations.

Moreover, in such a case, the discrete concept of 'representation capacity' may become inapplicable. Each layer may express different continuous arrangements, where differing concepts are angularly suppressed and expressed in analogy to the linear features hypothesis [33]. Instead, the '*magnitude-direction hypothesis*' is proposed as a continuous extension: with magnitudes indicating the amount of stimulus present, direction indicating the concept. Activations then populate this more continuous manifold, which is argued to enable more meaningful interpolations.

This continuous semanticity may also produce a better-organised semantic map at each network layer, since intermediate representations may now relate otherwise discrete features. The lack of discretising bias may allow semantics to be brought continuously into proximity (which 'weight locking' discussed in *Sec.* 2.3 may typically prevent). A manifold without form-induced structure may aid researchers in the emerging field of representational alignment, discussed further in *App.* D.4.

Therefore, in terms of representations, the inductive bias of isotropy appears more appropriate as a default, due to many real-world semantics being continuous and not being quantised into discrete bins through functional form induced structure. However, anisotropy may also be a good inductive bias if universal discretisation of concepts at all scales and abstraction levels is expected along the standard basis. Isotropy can be thought of as introducing an inductive bias that enables continuous and interpolatable semantics while retaining discrete semantics when task-necessitated, as opposed to being design-imposed structure. Hence, it generalises the discrete linear features paradigm into a more continuous setting.

2.3 Weight Locking, Optimisation Barriers and Disconnected Basins

'*Weight locking*' is a term to describe how, particularly, the weight parameter may suffer from being stuck in local minima found further into loss valleys, encountered only after a sufficient amount of training. Similar locking of biases near $\vec{0}$ may also occur. This optimisation artefact is predicted to occur through two modes — both a result of the anisotropic functional

form's *discrete* permutation symmetry. The secondary effect may arise as a consequence of discrete representations due to functional form symmetry breaking, and this more intuitive mode is discussed first.

Qualitatively, this indirect mode may arise if semantically meaningful linear features tend to become discretised and aligned with geometric positions about the distinguished bases. Hence, any small perturbation to a parameter may misalign activations to the network's existing semantics. A simultaneous corresponding downstream realignment by subsequent parameters would be unlikely, as it would be prevented by rotationally asymmetric non-linearities and the refraction of linear features. Consequently, this may largely halt further progress shortly after the formation of discrete semantic directions.

In effect, further small perturbations to the parameters may move representations from one discrete cluster (perhaps considered as a semantically aligned state) to a dislocated state (possibly without a precisely aligned semantics). An activation in such a state may be unprecedented for the network, so its action is untrained and prone to error, with potential semantic ambiguity as well. This anomalous state may negatively impact its corresponding output[7], resulting in reduced performance and increased loss. Thus, an emergent 'pseudo-minima' may form, due to the functional form's discretised semantics, discouraging such a scenario from even occurring. Consequently, this would suggest functional forms create a plethora of architectural local minima in the space, which are an indirect optimisation artefact due to anisotropic choices affecting representations. Only sufficiently large perturbations to a parameter can move activations between two semantically aligned directions, analogous to activation energy in chemistry.

It may also suggest that the optimisation barrier is some function of the angular separation between the semantically meaningful directions. Angularly closer linear features would be less likely to suffer activation dislocation if parameters nudge activations towards another close discrete cluster. By increasing the number of geometric positions representations occupy through varying anisotropies, this effect may be controlled. Extremising this would suggest that isotropy is beneficial since it is expected to produce continuous and interpolable representations. A dependence between the discretisation angles of representations and the optimisation barrier may offer a means to test this prediction.

This semantic dislocation would be an emergent consequence of breaking the continuous symmetric forms. The dual of this argument is basin connectivity, a mathematical result of the symmetry construction. Without continuous rotational symmetry, the loss landscape loses *direct* connectivity along the otherwise continuous orbit of its symmetry. Hence, these flat loss basins are changed upon breaking the symmetry, with many of their minima likely becoming isolated. In an $\mathbb{R}^n$ layer, an $n!$ degeneracy in weights arises if the activation function has an algebraic permutation symmetry equivariance, and the surrounding weights form a left/right-invariant general-linear closure — the precise definitions of such symmetry language are described in *Sec.* 5.2. This is a comparative isotropic-to-anisotropic consequence, which is a continuous analogue to the result of the discrete phenomenon observed in parameter symmetries work [29, 28].

Yet, enforcing the isotropy constraints results in sets of continuously connected local minima instead, which can be smoothly transformed into one another, by corresponding parameter rotations shown in *Eqn.* 14, a consequence of *Eqn.* 11. Convergence may also be hastened due to more efficient parameter utilisation, since directions resulting in functional equivalence are loss-invariant; they would not need to be redundantly explored. Additionally, analysis over these orbits may offer a mode to search for isotropic implementations by determining whether their constant loss is equal to or smaller than the infimum of loss on a corresponding anisotropic function across one of these isotropic orbits.

$$\forall \mathbf{R} \in \mathrm{O}\left(n\right) : \underbrace{\mathbf{W}^{l}\mathbf{R}^{\top}}_{\mathbf{W}'^{l}}\mathbf{f}\Big(\underbrace{\mathbf{R}\mathbf{W}^{l-1}}_{\mathbf{W}'^{l-1}}\vec{x} + \underbrace{\mathbf{R}\vec{b}}_{\vec{b}'}\Big) = \mathbf{W}^{l}\mathbf{f}\left(\mathbf{W}^{l-1}\vec{x} + \vec{b}\right) \tag{14}$$

If forms are downgraded to permutation symmetry, then artificial optimisation barriers may reemerge in these basins. The network would settle into a local minimum with loss barriers emerging in what would otherwise be a constant-loss connected path in isotropic networks. Gradient descent becomes trapped along such paths, unless a sufficiently large perturbation to the parameters can then dislodge the anisotropic network into a lower minimum. This produces a functional form induced perturbation threshold in the loss landscape. Effectively, the discrete permutation symmetry may result in overlaid sets of discretised lattice solutions for the parameters, much like how it breaks the symmetry of activations through training, too [17]. This is an alternative, dual-mode approach for the predicted pathology, which could be explored by smoothly interpolating between isotropic-to-anisotropic functions during training, thereby recovering an anisotropic network from an isotropic one. Additionally, various dynamics may emerge due to the flat-loss orbits, which may be of interest or possibly be factored out.

This former is primarily a qualitative prediction, as the hypothesis remains challenging to verify until robust methods for determining semantic directions are developed. The latter is a mathematical result and may be immediately tractable. Nevertheless, in either case, steps can be taken to counteract the problem, and this involves introducing isotropy to connect these minima.

3 Isotropic Implementations

This paper argues for the implementation of isotropic functional forms in neural networks to eliminate functional form-induced structure. Therefore, it is proposed as a preferable default inductive bias in many circumstances. However, near-term adoption may be rate-limited due to the development of suitably optimal functions, particularly since *anisotropic deep learning* has a substantial head start and analogues to existing functions are not so trivially transferred. Additionally, there is no expectation that superficially analogising anisotropic functions to be isotropic is useful; practical reselection of primitives may be necessitated.

[7] A potentially second mode through which adversarial attack can act: these minimally trained and unpredictable regions could be exploited to damage network performance by purposefully disaligning representations with established clusters.

Despite this, several preliminary implementations of activation functions are outlined in this section as a starting point; however, these are likely far from optimal and substantial research and development are required to bring isotropic deep learning into practicality. *Hopefully, the arguments of this paper will encourage the field to begin a directed search for more suitable functions with this guiding principle of isotropy.*

A summary of other functions, including optimisers, initialisers and normalisers, is also discussed in *App.* A. In *App.* D, several initial applications for Isotropy are discussed, including a modification to self-attention in *App.* D.1. A proposal for training-time dynamic network topologies in *App.* D.2, also provides a mechanism for learning plasticity and a dynamic alternative approach to the Lottery Ticket Hypothesis [34]. Below is a non-exhaustive list of activation functions and some consequences.

3.1 Activation Functions

As stated, the isotropic functional form for activation functions is given in *Eqn.* 11. In *Tab.* 3.1, it is compared with the other common functional forms. This comparison shows that the isotropic functional form is basis-independent and relatively simple. Further criteria, in addition to isotropy, are also essential for performance. However, these will be explored in future work.

Beginning from this functional form, familiar analogous to elementwise functions can be developed: isotropic-Tanh, isotropic-ReLU, and isotropic-Leaky-ReLU. However, it is hoped that the development of the approach will produce further activation functions that are not just analogues of existing activation functions but exploit the novel properties of isotropy for optimal performance.

Radial Basis Form	**Elementwise Form**	**Isotropic Form**
$\mathbf{f}(\vec{x}) = \sum_{i=1}^{n} \sigma\left(\lVert\vec{x} - \vec{c}_i\rVert\right) \hat{e}_i$	$\mathbf{f}(\vec{x}) = \sum_{i=1}^{n} \sigma\left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i$	$\mathbf{f}(\vec{x}) = \sigma\left(\lVert\vec{x}\rVert\right) \hat{x}$

Isotropic-Tanh is described in *Eqn.* 15. In basis directions, $\hat{e}_i$, it is equal in function to standard elementwise-$\tanh$, as indicated by its name. It is bounded, up to a norm of one, but does not angularly saturate like standard $\tanh$, allowing activation to continue semantically shifting.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}\right) = \tanh\left(\lVert\vec{x}\rVert\right) \hat{x} \tag{15}$$

This function is reasonably cheap, computation of $r = \lVert\vec{x}\rVert$, $\tanh\left(r\right)$ and $\operatorname{sech}^2\left(r\right)$, need only be computed once (including for backwards-pass) rather than per-component like the anisotropic functional forms. The vector norms are naturally constrained to $[0, 1)$, acting as an implicit normaliser. Around the origin, the transform is approximately the identity: $\lim_{r\to 0} \mathbf{J}\left(r\hat{x}\right) = \mathrm{I}_n$, justifying $\mathbf{f}(\vec{0}) = \vec{0}$, to preserve a smooth gradient. It is also globally 1-Lipschitz.

Isotropic-ReLU is shown in *Eqn.* 16, an analogue to its traditional implementation.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}; R_0\right) = \max\left(\lVert\vec{x}\rVert - R_0, 0\right) \hat{x} \tag{16}$$

Effectively, all activations are reduced by a threshold magnitude, R_0, with negative resultant magnitudes set to zero. Variations can be made to this activation function as shown in *Eqns.* 17 and 18, which include a maximum magnitude, R_∞ or do not reduce magnitudes except for below R_0, respectively. These activation functions continue to use $\mathbf{f}(\vec{0}) = \vec{0}$ property.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}; R_0, R_\infty\right) = \min\left(\max\left(\lVert\vec{x}\rVert - R_0, 0\right), R_\infty\right) \hat{x} \tag{17}$$

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}; R_0\right) = \begin{cases} \vec{0} & : \quad \lVert\vec{x}\rVert < R_0 \\ \vec{x} & : \quad \lVert\vec{x}\rVert \geq R_0 \end{cases} \tag{18}$$

Isotropic-Leaky-ReLU follows a similar form to ReLU; however, it linearly rescales the magnitudes below the threshold, forming a ball of smaller rescaled magnitudes. It is displayed in *Eqn.* 19, with a small value $0 < \alpha \ll 1$.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}; \alpha, R_0\right) = \begin{cases} \alpha\vec{x} & : \quad \lVert\vec{x}\rVert < R_0 \\ \vec{x} - \left(1 - \alpha\right) R_0 \hat{x} & : \quad \lVert\vec{x}\rVert \geq R_0 \end{cases} \tag{19}$$

Isotropic-Soft-ReLU uses $\alpha = 0$, and 'Isotropic-Soft-Leaky-ReLU', $\alpha \in (0, 1)$, are left in the derivative form of the radial part $\sigma'\left(r\right)$, shown in *Eqn.* 20. $\phi\left(r\right)$ is a monotonically increasing function. There are many suitable candidates fulfilling this ϕ, and one may be selected that has a suitable balance between performance, computation cost, and desirable properties. Imposed is $0 < \delta < R_0$, where $\delta < R_0$ is the centre-point of the interpolation window and 2δ is the width of this window. Consequently, the function blends smoothly between two linear regions of differing scaling.

$$\sigma : \mathbb{R} \to \mathbb{R}, \quad r \mapsto \sigma'\left(r; R_0, \delta, \alpha, \phi\right) = \begin{cases} \alpha & : \quad \lVert\vec{x}\rVert \leq R_0 - \delta \\ \frac{\phi\left(\frac{r - R_0 + \delta}{2\delta}\right)}{\phi\left(\frac{r - R_0 + \delta}{2\delta}\right) + \phi\left(\frac{R_0 + \delta - r}{2\delta}\right)} & : \quad R_0 - \delta < \lVert\vec{x}\rVert < R_0 + \delta \\ 1 & : \quad \lVert\vec{x}\rVert \geq R_0 + \delta \end{cases} \tag{20}$$

Isotropic-Sinusoids are a possibility that does not appear to have an analogue within the current anisotropic paradigm. It is one of a broad array of new possibilities. The aforementioned isotropic-ReLU-like functions may normalise magnitudes such that the distributions 'escape' the non-linear ball-region of the function, since utilising the non-linearity constructively may initially present an unpredictable learning hurdle. Therefore, a function that introduces non-linearity throughout the space may be desirable. Proposed is 'isotropic-Sinusoids', which allows for distributions to be compressed, rarefied and folded (for $|\lambda_m| > 1$) in a predictable manner. It is hoped that the network can utilise this for effective computation. This activation function is demonstrated in *Eqn.* 21. The monotonicity-violating parameter $\lambda_m \in \mathbb{R}$, may aid performance by enabling folding of embeddings.

$$\mathbf{f}\left(\vec{x}\right) = \vec{x} + \lambda_m \sin\left(\|\vec{x}\|\right) \hat{x} \tag{21}$$

The existence of norm-based functions is likely to occur in the literature, due to the prevalence of the norm as a standard linear algebra operation. Such individual instances may now incidentally align with various defined taxonomies — these can now be retrospectively reconsidered under this general taxonomic approach. An interesting instance, in a differing context, is the 'squashing function' [35], which was used in an architecture for dynamic-routing of vector-valued capsules with the explicit goal of ensuring normalised probabilities after summation by restricting the vector norm to $[0, 1)$. This construction was intended to better represent information on how an object appears such as pose, skew and more. Despite the squashing function's application as a probability normalisation step, using a vector-norm argument could be retroactively considered to conform to the functional form discussed, albeit not being framed as the general functional class reformulations this work is advocating for, due to this function's intended specific application for normalising capsules.

Overall, the isotropic functions put forward serve as illustrative instantiations of the general isotropic functional form. These functions serve as placeholders, as they only superficially match the appearance of standard anisotropic functions, which may not be optimal and likely require reselection of primitives under various group-based approaches. A directed search for new functions, from general isotropic forms and conditions, is expected to be beneficial in providing more effective functions for models and applications. Moreover, the purpose of this study into alternative primitive formulations and their inductive bias is not limiting such considerations to just activation functions, the foremost contribution is to a full-stack reformulation of *all* primitives, studying of their downstream effects and interplay — this is discussed further in *App.* A.

4 Alternative View Against Isotropy

It is argued that anisotropies result in unintended structure in activation distributions, which may be detrimental to the network's performance. This is because this inductive bias is typically introduced universally and on a fixed standard basis, and if no justification exists for this particular distribution, then it may be a suboptimal imposition by the network designer. However, arguably, some symmetry-breaking anisotropy may be beneficial, particularly if discretised semantics are the desired outcome for clean human-interpretable classifications. By clustering parts of the activation distribution, redundant information can be usefully suppressed, leading to classifications developing quicker. In classification, one of the most common applications of deep learning, this clustering may be a suitable a priori justification. Similar reasoning applies for contrastive learning methods [36, 37]. Therefore, introducing isotropy may limit the network's performance in such cases.

Despite this, current activation functions produce anisotropies along a Cartesian grid due to their dependence on the standard basis. This particular arrangement does not seem justified through classification. For example, Papyan et al. [38] demonstrated the phenomenon of 'Neural Collapse' onto an equiangular tight frame for classification networks, a different structure compared to the anisotropic distinguished directions. Hence, this does not align with the standard basis. However, the works of Logan and Shepp [11] suggest that decomposition onto the standard basis can aid with the curse of dimensionality, though this work is only indirectly associated with deep learning. In addition, Elhage et al. [16] demonstrate the phenomenon of '*Superposition*', which appears about the privileged basis. However, using this or studies of local coding in deep learning as support for anisotropy would be circular in logic, since these phenomena may be inherent or predicated on anisotropic networks. An example is demonstrated by Bird [17], who showed that the representational alignment phenomenon transforms under rotations into anisotropy, indicating that it results from anisotropic forms.

Nevertheless, more generalised anisotropies could be reintroduced under differing symmetries, with varying distinguished arrangements and in such a way as to mitigate the aforementioned predicted problems, such as dynamic refraction. This could consist of a more uniform distribution of anisotropic directions from which semantics could then, more discretely, develop. This is outlined in *App.* B, proposed to be one of the most crucial developments in the enlarged framework presented in *Sec.* 5.

5 Taxonomic Deep Learning

This section discusses the generalisation of the principles discussed so far in the paper. It indicates how group theory arises as the natural definitional tool when expanding to consider groups of primitives, and enables a generalisation of this approach to primitive level inductive biases. First, it is shown how *all* foundational primitive choices can be generalised over arbitrary groups, yielding various parallel primitive implementations collected under their respective groups, which are used to define them. Following this, the framework is further broadened to show how such considerations can be applied at the primitive scale and all their various compositions up to model scale. This recovers the foundational primitive-first approach, the parameter-symmetry approach, and Geometric Deep Learning, among new categories. This taxonomisation defines primitives and their various compositions, along three primary generations and three flavours per generation, under their maximal respecting group. This framework could then be expanded to include other algebras, fields, and upgraded to gauge definitions if desirable. In this section, and the wider paper, it is assumed that groups are linearly represented in standard matrix representation $\rho : \mathcal{G} \to \mathrm{GL}\left(n, \mathbb{F}\right)$, but is suppressed for notational simplicity. Generally, the differing

representations are expected to be an important additional detail of the taxonomy. Overall, this is put forward as a provisional and novel taxonomic structure, already capable of uniting several disparate approaches, but it is subject to improvement and is encouraged to be refined. It is aimed to offer insight and should be expanded where necessary, rather than kept rigid.

It is suggested that the remaining functions of deep learning could be audited using such tools to determine their respective taxonomies, while also employing this definitional approach as a generative means for producing and collating new implementations, novel and parallel, through function design. This may then enable the systematic exploration of their downstream bias-like effects, emergent phenomena and interactions of primitives under composition. This may be a highly beneficial direction for deepening the understanding of the deep learning family of approaches, offering a much richer research avenue than just isotropic deep learning.

5.1 Alternative Groups for Functional Forms

The considerations thus far involve a dichotomy between anisotropic (standard-permutation) deep learning and orthogonally defined isotropic deep learning. However, this dichotomy limits the choices available; such definitional constructions can be broadened to encompass general groups.

This positions isotropy as a potentially more favourable case study of possible alternative axiom-like constructions for deep learning, and standard-permutation deep learning as another alternative. However, a systematic analysis of the multitude of other possibilities enabled through these group-based definitions may be advantageous.

These can include other discrete groups yielding anisotropic-like branches, continuous groups, and groups defined over various fields, such as complexified groups. These symmetry groups can be further organised into various taxonomies for comparison, indicative of their group-based foundations. The example definitions are first discussed, followed by a more overarching symmetry formalism in *Sec.* 5.2.

The general definition of isotropic deep learning's primitives is centred around functions that abide by algebraic equivariance to maximally the orthogonal group actions (i.e., no larger supergroup of orthogonal groups). This is formalised through an equivariance relation, such as in *Eqn.* 22 defining a functional form for $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$.

$$\forall \vec{x} \in \mathbb{R}^n,\ \forall \mathbf{R} \in \mathrm{O}\left(n\right) : \quad [\mathbf{R}, \mathbf{f}] = 0 \tag{22}$$

It has already been observed that current deep learning's functional forms can be connected through a similar relation, as an equivariance of functional forms to maximally the permutation group ($\mathbf{P} \in \mathcal{S}_n$), as shown in *Eqn.* 23.

$$\forall \vec{x} \in \mathbb{R}^n,\ \forall \mathbf{P} \in S_n : \quad [\mathbf{P}, \mathbf{f}] = 0 \tag{23}$$

Additionally, some deep learning functional forms also respect an enlarged hyperoctahedral symmetry maximally: $\mathbf{B} \in B_n \cong \mathbb{Z}_2^n \rtimes \mathcal{S}_n$ (signed-permutations). This can include functions such as the standard elementwise $\tanh$ and the application of dropout through its distribution. The equivariance relation is shown in *Eqn.* 24. One could imagine creating an altered form of dropout, where the sign is more meaningfully preserved through a projection to a constant, non-zero, and appropriately signed value along the standard basis components.

$$\forall \vec{x} \in \mathbb{R}^n,\ \forall \mathbf{B} \in B_n : \quad [\mathbf{B}, \mathbf{f}] = 0 \tag{24}$$

Another possibility is that one could explore permutation under *even* sign flips (denoted D_n and not to be confused with the similarly notated dihedral group), as briefly discussed.

Similarly, quasi-isotropic functional forms (though not the specific instance in *Eqn.* 61, but discussed in *App.* B) can be connected through an arbitrary finite discrete rotational symmetry group Ψ_n, denser ($|\mathcal{S}_n| \ll |\Psi_n|$) or sparser than permutations. This can be represented through *Eqn.* 25.

$$\forall \vec{x} \in \mathbb{R}^n,\ \forall \psi \in \Psi_n : \quad [\psi, \mathbf{f}] = 0 \tag{25}$$

Hence, this can instead transition from a categorisation approach to primitives and a definitional approach for isotropic deep learning, to also being extended as a general definitional approach for generating various new primitive *collections* over any group structure. This is a generalised approach across all deep learning primitives and enables the development of alternative categories.

Considering the standard permutation, specific discrete-rotational, hyperoctahedral, even-signed permutation, continuous-rotational and general linear symmetries for an $\mathbb{R}^n$ space associated with a single layer can be constructed such that they are unified under a group hierarchy shown in *Eqn.* 26.

$$\begin{matrix} \mathrm{I}_n \subset \mathrm{S}_n \subset D_n \subset B_n \subset \\ \mathrm{I}_n \subset \Psi_n \subset \end{matrix} \mathrm{O}\left(n\right) \subset \mathrm{GL}\left(n\right) \tag{26}$$

Applying each of these to functional forms produces current deep learning ($\mathcal{S}_n$), tanh-networks (B_n), quasi-isotropic deep learning (Ψ_n) (of *App.* B), isotropic deep learning ($\mathrm{O}\left(n\right)$) and linear approximations ($\mathrm{GL}\left(n\right)$), respectively. Hence, several forms of machine learning can be unified under this approach, including linear approximations for general $\mathbb{R}^n$ to $\mathbb{R}^m$. This also appears to act as a tradeoff between the size of the linear constraint imposed and the non-linearities possible to satisfy these maximally. Similar can be achieved over other algebraic fields and group hierarchies; various other groups or products of groups are also possible. In this regard, it emphasises the critical importance of internal symmetry breaking in its implicit inductive biases, their interactions, systematic compositional interplay, and their emergent consequences. When restoring

the $\mathrm{GL}(n)$ symmetry, the expressivity of networks drops as a linear approximator is returned; hence, this would suggest that further insight into the success of non-trivial expressivity of deep learning may emerge along the internal interactions of symmetry-broken primitives. Isotropic deep learning is motivated as one way in which a balance between linear trivial expressibility and the potential detriments of harsher symmetry breaking can be considered. For all of these considerations, the proposed taxonomy could help reconcile this, capturing the nature of how broken symmetries may interact — motivating a close yet generalised inspection of the emergent biases. Consequently, it could be of considerable significance. Such considerations also persist in Geometric Deep Learning models, as discussed in *App.* E.2.

It is then natural to generalise this functional form equivariance relationship more broadly. For a symmetry family $\mathcal{G}$, one could impose a functional form equivariance as shown in *Eqn.* 27. This can produce general functional form taxonomies for deep learning approaches if motivated.

$$\forall \vec{x} \in \mathbb{R}^n; \forall \mathbf{G} \in \mathcal{G}: \quad [\mathbf{G}, \mathbf{f}] = 0 \tag{27}$$

For example, complexified groups can also be used, such as $\mathrm{U}(n)$ for continuous or $\mathrm{S_n} \rtimes \mathrm{U}(1)$ for discrete-like. If one considers only activation functional forms, these can manifest as displayed in *Eqns.* 28 and 29, respectively, other non-trivial forms may arise additionally. These can be notated more efficiently, but it would make less clear the explicit dependencies or behaviours of directions in the map. This demonstrates the approaches' flexibility, for reformulating *all* primitives by arbitrary groups. These may have various applications, and other primitive forms could be similarly constructed.

$$\mathbf{f}(\vec{x}) = \frac{\sigma\left(\vec{x}^\dagger \vec{x}\right) \vec{x}}{\sqrt{\vec{x}^\dagger \vec{x}}} \tag{28}$$

$$\mathbf{f}(\vec{x}) = \sum_{i=1}^{n} \sigma\left(\left(\vec{x} \cdot \hat{e}_i\right)^* \left(\vec{x} \cdot \hat{e}_i\right)\right) \frac{\left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i}{\sqrt{\left(\vec{x} \cdot \hat{e}_i\right)^* \left(\vec{x} \cdot \hat{e}_i\right)}} \tag{29}$$

Further group examples include the special-orthogonal group, which can be explored when extending to two-argument functions, and whether this orientation-preserving structure produces differing niche functional forms compared to the current use of orthogonal groups in defining Isotropic deep learning. Furthermore, using the above unitary group primitive definitions may have advantageous consequences for better representing quantum systems, which could be explored. In addition, continuous or discrete translational symmetries are also possible, with unusual repercussions for functional forms, producing axial-like or periodical forms, respectively. These might require a suitable justification for implementation or just exploratory in terms of resultant inductive biases when the symmetry interplays with networks. Overall, with nearly an infinite array of symmetry-constructed functional forms, it reinforces the notion that anisotropic and isotropic forms are just one of a vast set of possibilities for deep learning, and that more optimal choices may plausibly exist. Hence, the primary position is to make functional forms a deeply considered choice.

Overall, this taxonomic approach could be used as both a definition-based unifying perspective and a generative approach. When applied across all primitives, it produces novel and foundational sets which can act as design axes for the wider deep learning, upon which everything else is constructed. For example, such primitives can then be composed to produce *general* models for their branch, leveraging their biases, and models may be reselected under their conditions for greater benefit. Concepts may be portable through parallel implementations, but ground-up reimplementation and reselection may be optimal. Hence, it may be considered an axiom-like choice, producing distinct deep learning branches with their own primitives and resultant general models, while retaining the underlying directed graph.

Moreover, it can be demonstrated how these symmetries can all be associated with symmetries of this underlying directed graph, endowed with a field for activations. In short, one can construct a directed graph which represents the neuron connectivity of an arbitrary architecture. If nodes within this graph are organised and systematically divided into groups of node-layers, then the continuous activations assigned to these nodes can be said to form an $\mathbb{F}^n$ activation space. Before primitive implementations, such as parameter initialisation and specific functional forms, these layers could be considered to feature innate symmetries of their $\mathbb{F}^n$ space. This connects to the closure principle in the following section and should not be confused with the restriction of model classes to the forthcoming algebraic constraints, which define geometric deep learning — the models considered in this paper are the broader functional class typically existing under the more general closure conditions.

Acting on the arbitrary graph with such an automorphism leaves these representation spaces invariant. As a result, one can elevate the intrinsic symmetries by applying the new group-structured primitive classes, which augment the graph and form the overall model. Typically, a class of symmetry-defined primitives is chosen that is not inherently symmetry-broken by this underlying graph — *this does not mean the model as a whole has an end-to-end symmetry, it is instead the very structure of the model itself which is invariant*. This arbitrary graph construction also generally indicates the form of constraint applicable for the class, which can then be upgraded from closure to a stronger constraint, such as *Eqn.* 27, and is discussed in the following section. These allow the selection of functional forms and parameter initialisations, which are an automorphism of the graph. It is proposed that this represents a powerful unifying direction for deep learning, grounding the construction in graph and set theory. Collections of group-defined primitives can be generally constructed for broad applicability and then implemented in specific models where the automorphisms permit. Hence, the various group-defined primitive forms serve as an axiomatic-like choice. This then percolates up through all the compositions, yielding distinct branches, functional forms, theorems, phenomena and models as suggested.

Hence, it principally enables the exploration and comparison of substantially differing categories of primitives as a primary new design choice, each with meaningful inductive biases and downstream consequences. It may be worth considerable exploration and generate categories of what could be considered novel and hopefully practical forms of deep learning, due to their most broad influence. It will require an assessment of which phenomena and analyses are contingent on such choices so far, which could yield deeper insights into even more fundamental aspects of the field — or branch-specific phenomena and results.

Moreover, this may appear to suggest that the *choice* of the underlying graph's structure is independent of the symmetry — this is argued to not be the case in practice. The various primitive classes can exist a priori, and only the arbitrary graph determines which ones are inherently applicable to that model. It is hypothesised that co-adaption can then occur, with performance metrics selecting for suitable architectures and instantiations of functions specified by the functional form. This is because it is hypothesised that the form induces unique inductive biases into the model, and the success of a model architecture or observable phenomena may be contingent on this choice. Hence, the structure of many large models may equally be a product of selection upon the primitives they were built from and the symmetry nature of benchmarks judged against. This requires rigorous analysis and auditing of models and their primitives. A circular co-adaptation may be strong and should be considered foremost. This circularity is also discussed in *App.* F. Hence, these are suggested to be the two fundamental interrelated bases for deep learning's construction, the group-defined primitives and the inherent symmetries of affine maps connected to graph theory — the former acts as a set of axiomatic-like choices, which are broadly universal in nature, whilst the latter depends on the specific objective for the model. Overall, alternative primitives likely incur the reselection of graphs requiring new models per branch, whilst the graph is shaped by which maximal symmetries it is automorphic to, and hence available to act on by primitives.

In practice, the axiomatic-like categories of primitives would pre-exist in either case, fixing one then leads to selection for a resultant model, which is a search among architectures bounded by those which are automorphic to the fixed group. Equally, a single fixed architecture may examine its possible primitive choices from this set and then be assigned a resultant branch. Sets of primitives exist for universal application, and then one is selected depending on the application — chosen to produce a novel architecture or chosen in accordance with an existing architecture. This fixing and then bounded architecture search, or vice versa, is a novel design choice. This highlights the potential consequences stemming from the group-theoretic choice in foundational primitives.

These are an ongoing direction for the author, including further details regarding the graph-automorphism construction and interplay.

One could also choose a group ($G_{\text{prim.}}$) for the primitive, which is a superset of the graph automorphism ($H_{\text{auto.}}$). The network innately symmetry-breaks the primitive symmetry, and would result in an intersection of the symmetry: $H_{\text{eff.}} = H_{\text{auto.}} = H_{\text{auto.}} \cap G_{\text{prim.}}$, since $H_{\text{auto.}} \subset G_{\text{prim.}}$. The practicality of this is unclear, but it may be investigated to probe the consequences of a primitive and connectivity mismatch in respective symmetries. This may result in considerable primitive biases from the mismatch and could be investigated with the aim of leveraging this phenomenon as well. These could be used to desirably shape representational geometry, much like any other symmetries being studied.

Overall, generalising to the above symmetry taxonomic approach, derived from architectural automorphisms, could be considered a unifying approach. This further enlarged formalism can be constructed to more precisely generalise such findings over all primitives systematically. It is also capable of recovering most contemporary symmetry approaches in deep learning as compositional cases. This unification could be beneficial in bridging these alternative approaches and uncovering more general group-based phenomena in the field.

5.2 Generalised Formalism for Taxonomic Deep Learning

There are various symmetries already discussed in the literature, defined in a plethora of contexts. The intention is to demonstrate how several distinct approaches to symmetry can be unified into a single overarching framework. This not only clarifies each's scope, but may also aid in comparing, bridging, and categorising the disparate approaches whilst making intermediate considerations apparent as well.

Additionally, these relations add further nuance to the constraints previously described. Each of the following relations is hypothesised to carry its own inductive biases, which should be systematically explored and reconsidered in implementations. This adds a new design axis to be conscious of in model engineering. However, this proposal is also preliminary and may need further revisions or additions. Some possible extensions are discussed.

The motivation for this meta-framework is to unify all the various symmetry considerations in the literature under the following table of symmetries. It will be demonstrated how this approach categorises end-to-end symmetry-based networks of *Geometric Deep Learning*, *Parameter-Symmetries*, and *Primitive Foundation Design* (both Isotropic, Anisotropic, among other branches) as special instances within this overarching symmetry table and taxonomy. However, other scales and compositions are also encouraged for consideration, and some are put forward.

Proposed are three primary generations: symmetry closure ('inherent'), probabilistic/statistical ('weak') and algebraic ('strong'). These are then subdivided into primary flavours: left-invariant, right-invariant (for maps $f : \mathbb{R}^n \to \mathbb{R}^m$) and equivariant (for maps $f : \mathbb{R}^n \to \mathbb{R}^n$; this can be generalised to $f : \mathbb{R}^n \to \mathbb{R}^m$ if desired, but is not for brevity). Generalisation to other representations is discussed in *Sec.* 5.2.1.

Much of this discussion revolves around these symmetries applied to the forward-pass of primitives; a similar consideration in terms of backwards-pass symmetries could be applied to optimisers with a similar effect.

A function can abide by more than one symmetry per subdivision and have differing symmetries for each subdivision. However, if a series of abiding groups forms a hierarchy, then only denote the maximal abiding group; this is the defining

form. Similarly, if used definitionally for generating new functional forms, they are constrained by this symmetry maximally. For example, if a function is left invariant to both S_n and $\mathrm{O}(n)$, denote only the latter, since $S_n \subset \mathrm{O}(n)$.

Additionally, the three primary categories also form a hierarchy, with probabilistic ensuring symmetry closure and algebraic ensuring the other two of their respective subdivisions. This is indicated by *Eqn.* 30.

$$\text{Algebraic} \Rightarrow \text{Probabilistic} \Rightarrow \text{Closure} \tag{30}$$

Each category will be discussed, followed by examples that motivate it, and then a table will outline the symmetries of several functions.

Symmetry-closure indicates that any element of a functional class can be transformed under a symmetry, and the result is also a member of the class. This is displayed in *Eqns.* 31, 32 and 33 for left-invariant, right-invariant and equivariant respectivly. The class $\mathcal{F}$ would be a chosen subset of all maps.

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad (g \circ f) \in \mathcal{F} \tag{31}$$

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad (f \circ g) \in \mathcal{F} \tag{32}$$

Equivariant closure is a differing requirement, involving the pairing of its group-inverse (but can be generalised):

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad \left(g^{-1} \circ f \circ g\right) \in \mathcal{F} \tag{33}$$

These statements indicate that any function in a class which is transformed under symmetry is still a member of the class. Using representation theoretic generalisation it can be denoted $\forall f \in \mathcal{F}$, $\forall g \in \mathcal{G}$, $\rho^{(1)}(g^{-1}) \circ f \circ \rho^{(2)}(g) \in \mathcal{F}$ for two representations $\rho^{(1)}$ and $\rho^{(2)}$. Closure conditions do not say if these two instances of the class are equally likely to be initialised, which is a stronger condition.

This latter case is the probabilistic condition and is given in *Eqns.* 34, 35 and 36 for left-invariant, right-invariant and equivariant respectivly. $\mathbb{P}$ gives the probability of the member of the class $\mathcal{F}$ to be initialised. These could be termed a 'weak' accordance with a symmetry group.

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad \mathbb{P}(g \circ f) = \mathbb{P}(f) \tag{34}$$

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad \mathbb{P}(f \circ g) = \mathbb{P}(f) \tag{35}$$

Again, probabilistic equivariance is a differing requirement:

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad \mathbb{P}\left(g^{-1} \circ f \circ g\right) = \mathbb{P}(f) \tag{36}$$

The probabilistic condition can be specified as time-like, initialisation-like, data-like, or any combination of these. This depends on how the probability is considered. Time-like would be, for example, over subsequent iterations of forward passes in the network, discussed in *App.* C. Initialisation-like indicates the distributions of parameters which are spontaneously symmetry broken on initialisation. Data-like can consider the probability defined over samples of the dataset. Other situational subcategories for probabilistic conditions may exist and require extension to the formalism. Using representation theoretic generalisation it can be denoted $\forall f \in \mathcal{F}$, $\forall g \in \mathcal{G}$, $\mathbb{P}(\rho^{(1)}(g^{-1}) \circ f \circ \rho^{(2)}(g)) = \mathbb{P}(f)$ for two representations $\rho^{(1)}$ and $\rho^{(2)}$.

Finally, there are algebraic symmetry relations, or a 'strong' accordance with a symmetry. These indicate that every instance of a function in the functional class respects a symmetry which leaves the computation unchanged. They are defined by *Eqns.* 37, 38 and 39 for left-invariant, right-invariant and equivariant, respectively. These are the familiar bracket relations.

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad g \circ f = f \tag{37}$$

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad f \circ g = f \tag{38}$$

Again, algebraic equivariance would be a differing requirement:

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{G} \quad f \circ g = g \circ f \tag{39}$$

Using representation theoretic generalisation it can be denoted $\forall f \in \mathcal{F}$, $\forall g \in \mathcal{G}$, $\rho^{(1)}(g) \circ f = f \circ \rho^{(2)}(g)$ for two representations $\rho^{(1)}$ and $\rho^{(2)}$. One can also consider if the function has multiple arguments or concatenated output spaces. These generalised domains and codomains can have these conditions applied in various ways, e.g. direct sums or tensor products. Hence, one can extend the above definitions to functions with multiple arguments, including differing relations applied to any combination of arguments. Additionally, weight sharing can be considered another extension to the model. The following discussion concerns several applications of the formalism.

To begin with, *App.* B.1, discusses a functional form which introduces anisotropy but in an isotropically initialised manner. This motivated the construction of this formalism. This enabled such nuance in classifications beyond algebraic constraints. For example, $f(\vec{x}; \mathbf{W}) = \mathbf{W}\vec{x}$ is considered anisotropic since its algebraic equivariance is the identity, but it could still be weakly isotropic. Then considering $f(\vec{x}) = \sum_{i=1}^{N} f(\vec{x} \cdot \hat{e}_i)\, \hat{e}_i$, which has algebraic permutation equivariance, one may consider their

composition: $f(\vec{x}) = \sum_{i=1}^{N} f((\mathbf{W}\vec{x}) \cdot \hat{e}_i)\,\hat{e}_i$. This latter class is not strongly isotropic but could be weakly isotropic under suitable initialisation — this difference is significant and motivated by classifying forms through this meta-framework. This produced the original distinction between weak and strong symmetry accordance, which was extended with closure. This is an example of how the composition of layers can then undergo spontaneous symmetry breaking.

Deep learning models are hypothesised to leverage symmetry-breaking phenomena, which are further adapted through training, to achieve practical computation. Therefore, defining a alternative approach which prevents any such phenomenon would be impractical to purpose. This is one of the primary motivations of this paper: elucidating the role of symmetry breaking in networks systematically by exploring this taxonomy and altered primitives.

The approach to defining each fork generally considers algebraic symmetries in primitives, while allowing parameterised maps to be closed under the general linear group, in whichever relevant flavour. *However*, a pure branch would also initialise such parameters under probabilistic constraints to the symmetry of the branch. Hence, this would result in primitives respecting the overall intended symmetry before symmetry-breaking initialisations and learning.

As stated, the specific group used in such constraints is chosen from a selection that is considered derivable from arbitrary directed graph automorphisms, due to their connectivity, and then chosen to be applied to associated primitives. This can then result in the production of functional classes which have a closure under these respective automorphism symmetries. These maps, with closure under general automorphisms, can then be chosen to be upgraded probabilistically or up to algebraic constraints for specific groups they are closed under. Such considerations can also be applied at all scales: each individual primitive functional form and any possible composition of these through the arbitrary graph. In effect, this indicates which symmetries are in principle available before specific downstream *choices* of functional forms for primitives are chosen or freshly formulated. This interplay between architectures and primitive-constraints outlines the family of deep learning approaches and is suggested to be axiomatic-like.

As stated, from the closures available, one can *choose* a specific automorphism to form a stronger constraint to. This can be a subgroup of, or equal to, the whole automorphism group produced by the graph. For 'pure' branches, this choice results in parameterised mappings being upgraded to probabilistic, whilst functional forms over nodes pick up algebraic constraints. Functions from this functional class, which maximally abide by such restrictions, are then used to produce particular primitives for the specific network.

For example, this would typically be a restriction to a permutation family over the standard basis in contemporary deep learning. This constraint is then applied either probabilistically (such as affine maps $f\left(\vec{x}; \mathbf{W}, \vec{b}\right) = \mathbf{W}\vec{x} + \vec{b}$, which can have left and right probabilistic invariance) or algebraically (such as in activation functional forms $\mathbf{f}(\vec{x}) = \sum_{i=1}^{n} f(\vec{x} \cdot \hat{e}_i)\,\hat{e}_i$, which is algebraically-equivariant). A different choice can yield Isotropic probabilistic and algebraic forms, or any other symmetry fork of primitives. Some orthogonal probabilistic initialisers are used; this can now be termed a hybridisation.

Overall, the suggestion is that these are a *considered* choice in design. Isotropic networks are the statement that, at minimum, the primitives should be probabilistically in/equivariant maximally to orthogonal family actions, but strive for algebraic isotropy wherever feasible. Yet this is not dogmatic: a linear map would be too restrictive for computation if algebraically-orthogonal in/equivariant, so in such cases, only probabilistic-orthogonal in/equivariance, as spontaneous symmetry breaking, would be desirable. It is the *consideration* of such choices which is important, and knowledge of their induced biases, not the *restriction* to them.

Hybridisations between differing symmetry group definitions in a single model can be explored and appear to have justification already; these would occupy intermediate positions compared to the more purely defined branches proposed. Therefore, although the pure branches are rigid in their form constraints, explorations of hybridisations are also encouraged, analysis of which is enabled through this formalism. To the best of our knowledge, these are believed to be distinct research directions compared to previous symmetry-based literature.

As stated, this formalism is effective at distinguishing several approaches, including the considerations of this paper from Geometric Deep Learning's end-to-end symmetry-defined networks, such as equivariant networks, and Parameter Symmetries. The latter two regard a different scale to which these relations apply.

Geometric deep learning's equivariant networks can be recovered from the framework by considering the function class of the entire *model* and restricting the model class to those which are *algebraically* equivariant to the intended symmetry group, observed in the underlying data distribution. Similar applies to invariant flavoured constraints on models. Additionally, one can consider group convolution as the next compositional scale down to which these apply: functional class blocks over which these constraints are applied. Such components do occupy smaller-scale compositions, though they are in furtherance of an end-to-end *algebraic symmetry of the full model*. This is an entirely different objective from those presented in this paper, which are intended for general application in arbitrary architectures, allowing and shaping symmetry breaking through considering differing taxonomic generations composed in models. This represents a significant distinction in approach. These taxonomic considerations consider the implicit inductive biases which act on and interplay within the network's dynamics; it is not solely focused on constructing models about a desired equivariance or invariance to the group necessary to preserve the data structure. Hence, one can consider this proposal as the consequences of symmetry breaking in general networks. Nevertheless, this overall formalism can encompass both approaches as a special case of symmetries applied over functional classes at differing scales and philosophies.

Hence, this typically differs in scale from most of the considerations in this paper's approach, which focuses on the functional form primitive's relation and its *increasing interacting compositions*, as opposed to the model as a whole and the restricted functional classes required downwards to achieve such model-scale results under composition. These are complementary but differing approaches, with independent objectives and philosophies; yet, some interplay can likely be

established. Such interplay may be considered through this broader formalism. However, at present, the approaches between these primitive foundation reformulations and equivariant networks appear to often be mutually exclusive in many GDL models due to the latter's frequent dependence on elementwise primitives, but this is not always the case. However, this may be reconciled over time under alternative implementations and is discussed further in *App.* E.1. This significantly differentiates the typical scales at which Geometric Deep Learning's equivariant models consider symmetry from those at which this paper considers them. An algebraic equivariance over the entire model has not been the primary concern of this paper. Moreover, other differences arise, such as how the reformulation of primitive pertains to symmetry *families* due to the different dimensionalities of their respective maps, whereas models adhere to a single group preserved throughout.

Additionally, this formalism appears to suitably distinguish recent observations and consequences due to *Parameter Symmetries*, alongside their parameter-space degeneracies. This is the avenue investigating how specific symmetry actions can leave aspects of *existing* networks unchanged in functionality, resulting in parameter degeneracies. This phenomenon can be united into the framework as a compositional consequence. Three such relations make this evident: a right-closure of a general linear layer, an algebraic-permutation equivariance of *activation functions* and a general linear left-closure of a linear layer. Using this formalism, one can now extend the considerations to other primitive compositions for analysis as well.

For example, a linear layer $f_1(\vec{x}) = \mathbf{W}_1\vec{x} + \vec{b}_1$ ($\mathbb{R}^m \to \mathbb{R}^n$) has a left-closure (and right-closure) to $n \times n$ general linear transforms, since the functional class can take on differing parameter values. Similarly for $f_3(\vec{x}) = \mathbf{W}_3\vec{x} + \vec{b}_3$ ($\mathbb{R}^n \to \mathbb{R}^p$) which has a right-closure (and left-closure) to $n \times n$ general linear transforms. Informally, this means these layers have the capacity to 'absorb' any general-linear, or subset of, transforms whilst remaining in the class. Finally, the activation function, say $\mathbf{f}(\vec{x}) = \sum_{i=1}^{n} \tanh(\vec{x} \cdot \hat{e}_i)\,\hat{e}_i$ has a signed-permutation algebraic equivariance, meaning *Eqn.* 40 follows from the algebraic-equivariance to $\mathbf{P} \in B_n$ group.

$$\mathbf{f}(\vec{x}) = \mathbf{f}(\mathrm{I}_n\vec{x}) = \mathbf{f}\left(\mathbf{P}^{-1}\mathbf{P}\vec{x}\right) = \mathbf{P}^{-1}\mathbf{f}(\mathbf{P}\vec{x}) \tag{40}$$

Following this, the composition of $f_3 \circ \mathbf{f} \circ f_1$ means that the left and right general-linear closures can 'absorb' these $\mathbf{P}$ and $\mathbf{P}^{-1}$ transforms whilst remaining in the class. This is because $f_3 \circ \mathbf{P} \in \mathcal{F}_3$ and $\mathbf{P}^{-1} \circ f_1 \in \mathcal{F}_1$. This combination of properties reproduces the parameter symmetries under composition $f_3 \circ \mathbf{f} \circ f_1$.

This recasting of parameter symmetries under the above symmetry formalism may aid generalised considerations and comparisons. This approach reveals a large number of degeneracies, per such sandwhiched construction of an algebraic-equivariant maps $\mathbb{R}^k \to \mathbb{R}^k$ and associated closed linear layers, the network acquires a multiplicative $(k!)^2$ factor degeneracy in its parameter space if the activation function is S_n algebraically equivariant. Independently, this was considered a pathology between anisotropy and isotropy within this work, and can be connected to the discussion in *Sec.* 2.3.

Additionally, this also highlights that the emergence of a B_n or S_n symmetry, in particular, is *not* due to the parameters (which are general linear invariant closures); instead, it is the function they sandwich, in this case the activation function, which is algebraically equivariant to a transform. This aligns with Godfrey et al. [28], which identified and explored permutation-related symmetries over existing activation functions. Their intertwiner groups in activation functions correspond to those of parameters, and can be directly mapped to the above formalism discussed[8].

Extensions to this can be considered under this generalised formalism, such as making clear the consequences if the affine maps are *not* left/right closed under group $\mathcal{G}$, when sandwiched with an algebraic-equivariant $\mathcal{G}$. In such cases, parameter symmetries are not applicable and the network can become unique. One could also consider the promotion or prevention of such closure symmetries up to probabilistic conditions in a similar manner. Prevention may cause the initialiser always to favour particular arrangements of the network under symmetry, potentially aiding in alignment efforts.

Furthermore, one can consider the compositional consequences of adding noise under a probabilistic constraint. This is a suggested regulariser discussed in *App.* C. It may also have consequences for generative efforts, which could be explored.

In addition, more careful treatment of symmetries in Radial Basis Functions [30] ($\mathbb{R}^n \to \mathbb{R}^m$) can be explored. These appear to feature a permutation left-closure (S_m) and orthogonal right-closure ($\mathrm{O}(n)$), which, when combined with linear layers, can form similar parameter symmetries to those discussed.

One can also consider other compositions, such as the maps which are composed to form residuals $\mathbf{f}(\vec{x}) = \vec{x} + \mathbf{g}(\vec{x})$, where one can consider how $\mathbf{f}$ acquires the symmetry of $\mathbf{g}$. This is because the function is the summation of an identity map with a general map $\mathbf{g}$. The identity commutes with any algebraic symmetry, so the resulting symmetry of their composition is only defined through $\mathbf{g}$'s subset symmetry.

Furthermore, it appears that there is a tradeoff between the level of the maximal constraint on the functional form and the result on representations. For example, algebraic S_n equivariance is less of a constraint on functional forms than algebraic $O(n)$ permutation equivariance. Yet, the latter appears to produce fewer constraints on representations by removing absolute reference directions, the distinguished directions. This tradeoff between algebraic constraints and resultant representational constraints presents an interesting avenue for exploration.

Overall, this classification construction seems capable of both demarcating and unifying multiple differing approaches to symmetries within deep learning. Hence, this broader symmetry formalism may be highly advantageous to explore further. The present focus of each's approach is pictorally demonstrated in *Fig.* 4. It also indicates how naturally the hierarchy of constructing models may require analysis of smaller compositions, such as discrete convolutions for equivariant networks. Still, it is in furtherance of the model-scale regime to which the symmetry constraints are applied. Due to its lowest hierarchical positioning, if a reformulation of the primitive foundation occurs, then it has consequences for all layers, all compositions, and

[8]They also demonstrate a connection to representations. This is further supporting evidence for this paper's hypothesis that activation functions produce an inductive bias on representations.

all models in all applications, extending upwards. It is the study of the *interplay* and emergent phenomena between symmetry and networks these bring, as well as using it as a definitional tool across *all* primitive to generate group-defined classes, from which reselection of specific instantiations can occur.

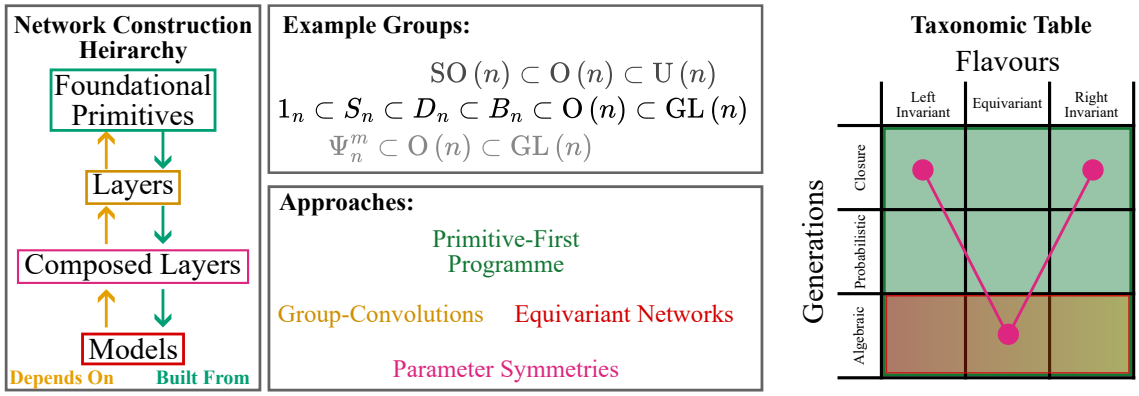

Figure 4: Pictorally demonstrates the various approaches to symmetry in contemporary deep learning, through their generation and flavour regimes as well as their typical scales. Left demonstrates the hierarchical dependencies in the construction of deep learning systems. Dark orange indicates a top-down approach philosophy, which seeks model-scale symmetries derived for purpose-built, targeted applications, and consequently constrains *algebraically* downwards to ensure this. In contrast, the mint colour represents a bottom-up, causal-effect, and group-theoretic philosophy, where new compositions are generated from and are contingent upon smaller-scale constructions and may occupy more general generations within the taxonomy. Centre-top provides several group taxonomies which may be considered for implementation. The centre-bottom specifies several approaches to symmetry and is colour-coded to identify regimes in the leftmost and rightmost diagrams. The rightmost depicts the taxonomic organisation put forward. Parameter symmetries are the composition of left/right-invariant closures with contemporary activation function equivariance to permutation-like groups, which is indicated by the fused triangle in the taxonomy. This also raises the problem of defining the notions of layers and primitives, which is addressed in an upcoming paper.

As a consequence, end-to-end models can contain instances where specific primitives can be reformulated such that the model as a whole respects the symmetry; however, they cannot encompass the primitive-first paradigm as a whole, as this is a superset due to forming the foundational base and its argued consequences for all general models. This indicates the present approach's universal, axiomatic-like importance for consideration, as any consequences further interact with compositions contingent upon them. This is not to imply that one philosophy is superior to another; they target differing objectives yet may be complementary. One is already well-established and growing, with several state-of-the-art results that already indicate success in achieving its intended goals. The other is attempting to determine how symmetries from functional forms may interact with networks as unintended inductive biases, and then reformulating primitives for beneficial purposes in general architectures. Distinctions are drawn to avoid confusion between their objectives and considerations, as they independently share a group-theoretic root as their natural expression, which may appear similar at first due to its relative infrequency in deep learning. This is similar to how parameter symmetries also share a group-theoretic root and are again distinct. The objective of this formalism is to draw on this shared overlap of group-theoretic considerations to construct an overarching framework that provides a more comprehensive and high-level perspective.

Hence, this group-theoretic approach provides unifying terms to compare these within the context of deep learning. The approach of this paper is to be primarily conscious of such decisions regarding functional classes, *at all scales*, in their introduction of inductive biases and interactions. This process begins with generating numerous new families of foundational primitive implementations through careful searches and selection within these functional classes, followed by building these upwards in compositions for novel architectures and potentially improved, generally applicable models.

Tab. 5.2 roughly indicates the *typical* symmetry properties of conventional forms. For example, a linear layer can be chosen to be initialised isotropically, but itself does not display the associated algebraic symmetry. Each such choice restricts the functional class and incurs specific inductive biases to consider. Question marks on the equivariant network indicate the specific model's chosen initialisations.

Name	**Function**	**Closure**			**Probabilistic**			**Algebraic**		
		L	R	E	L	R	E	L	R	E
Affine Layer	$\mathbf{W}\vec{x} + \vec{b}$	$\mathrm{GL}(n)$	$\mathrm{GL}(n)$	$\mathrm{GL}(n)$	$\mathrm{O}(n)$	$\mathrm{O}(n)$	$\mathrm{O}(n)$	I_n	I_n	I_n
Standard-Tanh	$\sum_{i=1}^{N} f(\vec{x} \cdot \hat{e}_i)\,\hat{e}_i$	I_n	I_n	B_n	I_n	I_n	B_n	I_n	I_n	B_n
Composed	$\sum_{i=1}^{N} f((\mathbf{W}\vec{x}) \cdot \hat{e}_i)\,\hat{e}_i$	S_n	$\mathrm{GL}(n)$	S_n	S_n	$\mathrm{O}(n)$	S_n	I_n	I_n	I_n
Equivariant-Nets	f_{models}	$\mathcal{G}$?	$\mathcal{G}$?	$\mathcal{G}$	$\mathcal{G}$?	$\mathcal{G}$?	$\mathcal{G}$	I_n	I_n	$\mathcal{G}$
CE Loss	$\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$	I_n	S_n	I_n	I_n	S_n	I_n	I_n	S_n	I_n
Isotropic-Tanh	$\sigma(\lVert\vec{x}\rVert)\,\hat{x}$	I_n	I_n	$\mathrm{O}(n)$	I_n	I_n	$\mathrm{O}(n)$	I_n	I_n	$\mathrm{O}(n)$
Layer-Norm (γ_i, $\vec{\beta}$)	$\gamma \odot (\vec{x} - \mathbb{E}(\vec{x}))/\mathrm{Var}(\vec{x}) + \vec{\beta}$	$\mathbb{R}^{\times}, +\vec{\lambda}$	$\mathbb{R}^{\times}, +\lambda\vec{1}$	$\mathrm{O}(n-1)$	1_n	$\mathbb{R}^{\times}, +\lambda\vec{1}$	$\mathrm{O}(n-1)$	1_n	$\mathbb{R}^{\times}, +\lambda\vec{1}$	I_n
Layer-Norm (γ, β)	$\gamma(\vec{x} - \mathbb{E}(\vec{x}))/\mathrm{Var}(\vec{x}) + \beta\vec{1}$	$\mathbb{R}^{\times}, +\lambda\vec{1}$	$\mathbb{R}^{\times}, +\lambda\vec{1}$	$\mathrm{O}(n-1)$	1_n	$\mathbb{R}^{\times}, +\lambda\vec{1}$	$\mathrm{O}(n-1)$	1_n	$\mathbb{R}^{\times}, +\lambda\vec{1}$	$\mathrm{O}(n-1)$

One can extend this formalism in numerous ways. For example, one can upgrade the group-theoretic considerations to gauge-theoretic, if an application suitably justifies such a generalisation of the approach.

Additionally, one can consider more general symmetries, for example $\mathrm{O}(1,3)$, where a metric-tensor can be inserted into the functional form to produce a pseudo-norm: $\mathbf{f}(\vec{x}) = f\left(\vec{x}_\beta g^{\beta\gamma}\vec{x}_\gamma\right)\hat{x}_\alpha$. This has some interesting consequences.

Particularly, one can stack the metric of such a function, similar to the manner in *App.* D.3, producing $\mathbf{f}(\vec{x}) = f(\vec{x}_\beta g_h^{\beta\gamma}\vec{x}_\gamma)\hat{x}_\alpha$. If the contravariant and covariant indicating indices are dropped, whilst allowing non-symmetric metric for later pairwise comparisons then the following equation can be considered: $\mathbf{f}(\vec{x}) = f(\vec{x}_\beta g^h_{\beta\gamma}\vec{x}_\gamma)\hat{x}_\alpha$. Moreover, the metric $g^h_{\beta\gamma}$ can then be expressed generally as the product of two matrices W^h_k and W^h_q, returning to standard matrix notation: $\mathbf{f}(\vec{x}) = f(\vec{x}^T(W^h_q)^T W^h_k\vec{x})\hat{x}$ and considering $K = W_k\vec{x}$ and $Q = W_q\vec{x}$, then a self-attention-like structure, $\mathbf{f}(\vec{x}) = f(Q^T_h K_h)\hat{x}$, is closely recoverable, and could be generalised for the value matrix, pairwise inner-products and normalisation factor[9]. Changing to a non-softmax activation function, which doesn't depend on other elements or components, reinforces this generalised symmetry consideration and partially motivates the discussion *App.* D.1.

Furthermore, one can make the apparent metric position dependent, having representational similarity follow from metrics over a non-linearly contracting and expanding space. This could be a meaningful avenue to explore, clustering regions of representation space or dispersing others. Hence, this symmetry formalism enables a recontextualisation which may have consequences for different comparisons between representations and potentially improved expressibility in attention. Therefore, this symmetry approach also allows the reinterpretation of self-attention-like operations within this symmetry formalism.

Overall, this formalism is highly versatile in categorising symmetry techniques within deep learning, potentially enabling improved cross-communications whilst also making clear alternative avenues to explore. The intention was to enable further comparisons and generalisation to more examples, using this approach to both audit existing functions into a categorised taxonomy and to use it as a definitional and generative method for producing functions and models collated by group structure. This can occur in parallel to understanding the ramifications of such symmetry definitions through representational geometry and mechanistic interpretability, optimisation and performance, parameter degeneracies and more.

5.2.1 Note on Representations

The following three paragraphs briefly discuss representation-theoretic additional considerations that may be important for taxonomisation, but are nuances that may complicate the utility of the overall taxonomy for general practice.

Moreover, these flavours can all be reformulated in terms of representations, which encompass and extend the equivariance/invariance formulae detailed below, and could be organised using a highest weight approach, such as indexing isotropy with Casimir operators. This can be used to label these representations and provide a more principled, primitive, and compositional framework. Hence, using a representation theoretic approach, for corresponding left and right actions, can add further nuance to the primary flavour categorisations notated, and likely representation theory may better organise foundational biases and their interactions. This should be undertaken, but is notationally suppressed for the sake of approachability, and is assumed as an implicit and vital part of the taxonomisation. Furthermore, considering the countless groups possible, and all the various possible representations, it is encouraged that general foundational bias principles are distilled down over particular families of groups defining primitives, e.g. orthogonal as opposed to particular instances $\mathrm{O}(5)$, $\mathrm{O}(8)$, $\mathrm{O}(100)$ etc. This generalised collation of biases may often offer better practical benefit and leveragability, rather than more niche instances of specific groups, so this is encouraged foremost. For example, it was shown in Bird [17] that the axis-anti/alignment generally persisted independent of the network width, which would indicate differing specific instances of the permutation family: S_{24}, S_{32}, etc. — this is the primary objective of this programme. Therefore, although a more fine-grained categorisation of primitives is possible, it may be advantageous in general to categorise them only by group and dissimilar representations.

Additionally, representations connected under a conjugated transform $\rho'(g) = \mathbf{A}^{-1}\rho(g)\mathbf{A}$ may have meaningfully differing inductive biases depending on $\mathbf{A}$, particularly whether there exists an element $g' \in \mathcal{G}$ for which $\mathbf{A} = \rho(g')$ is a representation. If this is not the case, then the resultant bias may be non-trivial and could be organised by what group $\mathbf{A}$ is a representation of. An example is that if a weight-decay regularisation is used, and two differing equivalent representations are chosen for a primitive composed with affine layers, then if $\mathbf{A}$ has column-wise or row-wise vectors which do not normalise to one, there may be meaningful compositional biases between L2 regulariser-affine-activation function interactions. A similar argument applies if $\mathbf{A}$ is not in the representation of the permutation defining an anisotropic activation function; then it will interact meaningfully with an L1 regulariser. This suggests that more than just irreducible representations may be relevant as foundational biases, and further investigation is required. This will also inflate the considerations of the foundational bias scheme; however, it again is likely beneficial to consolidate these into general principles for wider adoption, although specific instances could still be applied if desirable. An example of this is the permutations used in Bird [17] for rotated and non-standard-basis representations, where it was found that no observable differences in biases arose. Furthermore, there is an exponential growth in considerations when considering compositions; the desire is to distil general principles over constructions of only group-defined primitives.

Additionally, mismatching representations may be similarly useful if a suitable application required them. For example, an equivariance between a unitary spin-$\frac{1}{2}$ fundamental representation and an orthogonal spin-1 representation can be established. This would be a representation-generalised form of orthogonal primitives. One can form an activation function such as that in *Eqn.* 41, which uses the standard Bloch (Hopf-fibration) mapping. This Bloch/Hopf-activation function maps between a two-dimensional complex vector to a three-dimensional real vector: $\mathbf{f}_{\text{Bloch}} : \mathbb{C}^2 \to \mathbb{R}^3$, with equivariance

[9] Such a normalisation factor may be similarly applicable to standard isotropic activation functions too.

$\mathbf{f}_{\text{Bloch}}(\mathbf{U}\vec{x}) = \mathbf{R}(\mathbf{U})\,\mathbf{f}_{\text{Bloch}}(\vec{x})$, where $\rho_{SO}(g) = \mathbf{R}$ in a standard representation of $\mathrm{SO}(3)$ and $\rho_{SU}(g) = \mathbf{U}$ in a standard representation of $\mathrm{SU}(2)$, the matrix $\mathbf{R}$ is then parameterised by $\mathbf{U}$, given by $\mathbf{R}_{ij}(\mathbf{U}) = \frac{1}{2}\,\mathrm{tr}\left(\sigma_i \mathbf{U} \sigma_j \mathbf{U}^\dagger\right)$, or equivalently through the lie algebras. The Pauli matrices are given as a vector $\vec{\sigma}$ and the Bloch map $\mathbf{n} : S^3 \to S^2 \hookrightarrow \mathbb{R}^3$. These form an activation functional form class similar to isotropy, yet for mismatching representations. These Bloch/Hopf-primitives may find various applications, such as quantum mechanical modelling, where one may wish to convert between these quantities. Additionally, this Bloch map is non-linear and may contribute additional transforms to a network, particularly through projecting out the global complex phase resulting in degrees of freedom changing as $S^3 \to S^2$. This is related to the double cover property between representations defining the primitives. Similar mismatching representations defining primitives could be generated if practically necessitated.

$$\mathbf{f}_{\text{Bloch}}(\vec{x}) = \begin{cases} f\left(\vec{x}^\dagger \vec{x}\right) \mathbf{n}\left(\frac{\vec{x}}{\sqrt{\vec{x}^\dagger \vec{x}}}\right) & : \quad \vec{x} \neq \vec{0} \\ \vec{0} & : \quad \vec{x} = \vec{0} \end{cases} \quad \text{with} \quad \mathbf{n}\left(\vec{\psi}\right) = \frac{\vec{\psi}^\dagger \vec{\sigma} \vec{\psi}}{\vec{\psi}^\dagger \vec{\psi}} \tag{41}$$

Overall, it is suggested that, despite finer distinctions being possible through representation-theoretic principles, with foundational and compositional biases being additionally indexed by them, for the wider, general practicality of the taxonomy, it may remain primarily within group-theoretic definitions of primitives and general principles for their induced biases. However, niche cases may continue to utilise representation-theoretic considerations. This adds to why the group-theoretic approach is primarily showcased, with fewer primary flavours, alongside the notational approachability of this primitive-first programme.

6 Conclusion

This paper focuses on a novel case study of an isotropic functional form as a hypothesised better default inductive bias for deep learning. Current forms have been demonstrated in previous literature to produce task-unmotivated representational artefacts [17], which this work hypothesised may limit the networks' semantic expressibility. It is further argued that the current anisotropic functional forms may have detrimental effects on performance and learning through the predicted 'neural refraction', 'discrete semantics', and 'weight locking' phenomena. Removing such constraints from the model is also argued to unconstrain the representations from any particular basis. Hence, it is expected to produce a more natural activation representation based upon task necessities, rather than a structure induced by human-imposed functional forms. This may improve semantic structure and produce high-capacity embeddings, particularly important for applications discussed in *App.* D.

In isotropic networks, functional forms are promoted from the existing discrete permutation symmetry to a continuous orthogonal symmetry. This has substantial consequences for the form of almost every function in modern-day deep learning. This paper and appendices also outline a framework for connecting future work in these directions. Several preliminary activation functions, normalisers, optimisers, regularisers and operations are also described as a starting point. The tenets of reviewing such functional form choices are encouraged. Through comparison studies, this philosophy should, at the very least, reveal which characteristics of functional forms most significantly contribute to performance. This includes the role of symmetry breaking, which is one of the foremost concerns when developing this taxonomy, as it determines how various compositions may induce hierarchical interactions that aid or detract from learning.

A change to isotropic deep learning is argued to be generally advantageous, but may need substantial time for development as a mature alternative. New models and benchmarks may also require development to determine the practicality of this alternative approach. Additionally, better-optimised implementations that suitably leverage isotropy are preferable, as the ones described remain illustrative placeholders. The functions proposed so far are analogous to existing functions, which may not be inherently optimal for an isotropic network, even if they share superficial similarities. Therefore, empirical work on these placeholder functions will be presented in future papers, to not distract from the primary motivation for this shift to isotropic deep learning and the wider group-theoretic alternatives. The proposed ideas aim to stimulate the community's interest in conducting a directed search for better isotropic functions and determining whether this approach should be adopted in wider applications. The predicted pathologies may also offer a suitable falsifiable mode to validate the principles at this early stage.

Finally, other approaches to symmetry in deep learning are shown to be distinct from the approach proposed in this paper; however, an overarching symmetry formalism is also introduced to unify these disparate approaches and make clear other avenues to explore in a similar regard.

It is proposed that the breadth of the reformulations may constitute an alternative direction for deep learning: *Isotropic deep learning*, with the taxonomy of *Sec.* 5 extending this further. Connecting this to graph automorphisms yields axiomatic-like choices, forming distinct 'branches' of primitives and downstream models to consider. In general, it situates symmetry prior to neurons, rather than symmetry deduced from the neuron-defined computation graph — an ontological inversion of what is defined or deduced from a neural network. Extending this further may provide a complete axiomatic construction for deep learning. Additionally, new categories of primitives may be highly distinct from all that have come before, particularly with potential consequences on representations and learning. This may be leveraged to offer new approaches for deep learning in both practice and theory. This may require the development of various subdisciplines for the study of respective branches and any interrelatedness, such as the conjectured group universal approximation and bound theorems. Reanalysis of existing phenomena and results may also be undertaken to determine if they are predicated upon specific primitive choices.

In general, this taxonomic group classification may be an effective approach in categorising new functional forms and judging their interactions, foundational biases and symmetry-breaking phenomena. However, it is not a replacement for other analytical factors that may be considered in setting the functional form or after a group-defined functional form is fixed. For example, the identity group can generate such a broad repertoire of functional forms that is insufficiently distinguished by

group theory alone. Hence, other analytical tools, besides group-theoretic considerations, remain crucial to distinguish their foundational biases. Yet, a group-defined approach does enable a principled way to generalise over sets of primitives and is argued to be important in considering the useful computational maps that networks may leverage — such as in the orthant setup. Overall, it is argued that group theory provides a good foundation for defining initial primitive forms and an initial framework for categorising interactions. However, the categorisation of foundational biases is likely to outgrow the limits of group and representation-theoretic taxonomisation, alongside symmetry breaking. It may practically require an extension of the taxonomy beyond this, and group theory should be employed to extend the exploration, but not limit it.

Overall, this generates a considerable and novel design axis for the field of deep learning, which may be explored with the aim of producing better-performing models, gaining a better mechanistic understanding of their functions, discovering fundamental phenomena and results, and hopefully generalising to more applications.

6.1 Final Note on Philosophical Implications

Philosophically, this paper advocates for a shift in perspective on symmetry in relation to deep learning and an ontological shift in what could be considered a deep learning model, as well as the emergent consequences that stem from this.

This paper concerns the emergence of symmetry within deep learning itself and how it may, inherently and crucially, task-agnostically, act on a model's computation. This approach covers exploring the implications of this generally on models for all applications, contingent upon various choices of functional form definitions.

This is a markedly different assumption about the relationship between symmetry and deep learning compared to end-to-end model approaches, which leverage the established symmetries of the natural world using observations and arguments frequently emerging externally to the discipline, and extending these into deep learning for models to adhere to. These are extensions of a known external physical approach to symmetry and transferring into models. However, this paper proceeds from a drastically differing assumption regarding the emergence of symmetry, emphasising that it is internal to deep learning as well and has importance in its own right. It is making the argument and assumption that group-theoretic considerations are natively important characterising tools for the field, reinforced by it being already unintentionally set within the contemporary defining choices of primitives. Hence, it is not a perspective on emulating natural symmetries into a system, but considering that the functional form structure already within the system carries its own symmetry biases naturally attributable to and categorisable through symmetry — an externalist top-down versus internalist bottom-up perspective on symmetry for deep learning.

Hence, this constitutes a philosophical reorientation of symmetry, not just a tool for aligning models with the external world, but *also* as an internal, function-driven influence acting task-agnostically on representations, optimisation, interpretability, and more generally — the primary and generalised perspective shift this paper advocates for. Primarily, it is argued that these considerations may naturally arise internally, existing as important factors within the field, in addition to, but not requiring, motivation from the external world. Hence, it is contingent on an assumption that symmetry's importance as a categorising principle is not only imported but also native to deep learning.

Perhaps one of the most crucial aspects is the new ontology of what constitutes a deep learning approach, which this reframing provokes. It highlights that it may no longer be contingent on a computational system constructed upon a neuron-wise interconnectivity approach, but generalised across various group-defined primitive sets. This even generalises the notion of a neuron as an object generalised for higher-dimensional considerations, which is a downstream consequence after a symmetry definition is set. Definitions reverse from neurons preceding deducted symmetries, to symmetries defining neurons — and if symmetries are derived from automorphisms, then graphs precede these. It also contends that this is likely not limited to foundational biases stemming from parameter degeneracies, as a consequence contingent on current formalisms, such as affine layers plus a non-linearity; it also generalises the implications beyond this compositional structure, suggesting that even in such cases they do not need to be emergent phenomena solely predicated and formulated on degeneracies, but general function-driven consequences attributable atomically, such as neural refraction, and more in general compositions.

Hence, this differs from parameter symmetries, which identify computational equivalences under reparameterisations *deduced* when *assuming* the existing fixed set of primitives and the consequences therein. Instead, this work also changes the assumptions upon which those deductions are predicated by broadening the definition of primitives to a plethora of group-defined relations and considering the implications of these more broadly within the new ontology. Moreover, these function-driven implications are argued to exist for a model even where parameter-induced computational degeneracies may not. Hence, this is being considered as an exploratory approach to phenomena which may extend beyond those which are predicated on computational equivalences through parameter degeneracies, and therefore is not contingent on these to exist. For example, identity-defined functions may have similar foundational biases. Overall, the emphasis extends to more general ramifications, from all primitive definitions, both atomically and general compositional cases for networks. A trivial counterexample of how these extend past parameter degeneracies could be considering the diagonal basis-dependent and permutation-defined approximation of Hessian-like scalings for adaptive optimisers. This is a more trivial example, but similar constructions could be considered for activation functions and many other primitives, which may not depend on the specific parameter-symmetry composition structure for influence — such as in cases where a restricted linear map may not have the required closures, but still have phenomena dependent on the definition of the primitives used, for a single example: consequences of refraction can exist independent of degeneracy.

This also reframes several interpretability approaches, which may also be dependent on the existing set of primitives. Determining the semantics of representations may be mediated by these axiomatic-like choices, where artefactual structure may arise from primitive algebra alone. By broadening over differing group-defined sets, the geometry of representations may also alter. This may also affect the knowledge and deductions a model can make — broadening its epistemic horizon.

This interplay between imposed geometry of primitives and emergent consequences on representations and optimisations may constitute a 'no-free-geometry' consideration — reframing that observation of representation structure may be as much conditioned on the primitives as the data. Hence, care should be taken not to use it as evidence in support of the primitives circularly, but perhaps ascertain more fundamental and shared organisations for representations as insight into semantic relations and learning. This also motivates a shift in perspective on representations, from solely copying the external structure of reality to also being strongly structured by the non-derivable choices of functional forms inherent to the model.

Overall, this is a notably differing philosophy for symmetry in deep learning. It assumes that it already pre-exists and argues that it may be a constitutive and intrinsic property of deep learning, forming a leveragable, native, and foundational design axis. Philosophically, this is suggesting a pluralism redefinition for what constitutes deep learning by generalising it across various group-theoretic components constituting a model and determining and categorising their implications, such that they can be beneficially leveraged.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[2] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. URL https://arxiv.org/abs/1409.4842.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL https://arxiv.org/abs/1409.1556.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL https://arxiv.org/abs/1502.03167.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[8] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. URL https://arxiv.org/abs/1608.06993.

[9] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL https://arxiv.org/abs/1905.11946.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

[11] Benjamin F Logan and Larry A Shepp. Optimal reconstruction of a function from its projections. 1975.

[12] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[13] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. URL https://arxiv.org/abs/1606.08415.

[15] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

[16] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022. URL https://arxiv.org/abs/2209.10652.

[17] George Bird. The spotlight resonance method: Resolving the alignment of embedded activations. In *Second Workshop on Representational Alignment at ICLR 2025*, 2025. URL https://openreview.net/forum?id=alxPpqVRzX.

[18] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[19] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL https://www.sciencedirect.com/science/article/pii/089360809190009T.

[20] Chris Olah. Neural networks, manifolds, and topology — colah.github.io. https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/, April 2014. [Accessed 15-05-2025].

[21] Peter Foldiak and Dominik Endres. Sparse coding, Jan 2008. URL http://www.scholarpedia.org/article/Sparse_coding#:~:text=Sparse%20coding%20is%20the%20representation,subset%20of%20all%20available%20neurons.

[22] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[23] Taco S. Cohen and Max Welling. Group equivariant convolutional networks, 2016. URL https://arxiv.org/abs/1602.07576.

[24] Taco S. Cohen and Max Welling. Steerable cnns, 2016. URL https://arxiv.org/abs/1612.08498.

[25] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance, 2017. URL https://arxiv.org/abs/1612.04642.

[26] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical cnns, 2018. URL https://arxiv.org/abs/1801.10130.

[27] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

[28] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the symmetries of deep learning models and their internal representations, 2023. URL https://arxiv.org/abs/2205.14258.

[29] Derek Lim, Theo Moe Putterman, Robin Walters, Haggai Maron, and Stefanie Jegelka. The empirical impact of neural parameter symmetries, or lack thereof, 2024. URL https://arxiv.org/abs/2405.20231.

[30] David Lowe and D Broomhead. Multivariable functional interpolation and adaptive networks. *Complex systems*, 2(3): 321–355, 1988.

[31] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization, Nov 2017. URL https://distill.pub/2017/feature-visualization/.

[32] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations, 2017. URL https://arxiv.org/abs/1704.05796.

[33] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.

[34] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019. URL https://arxiv.org/abs/1803.03635.

[35] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules, 2017. URL https://arxiv.org/abs/1710.09829.

[36] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

[37] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

[38] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, September 2020. ISSN 1091-6490. doi: 10.1073/pnas.2015509117. URL http://dx.doi.org/10.1073/pnas.2015509117.

[39] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?, 2019. URL https://arxiv.org/abs/1805.11604.

[40] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

[41] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017. URL https://arxiv.org/abs/1607.08022.

[42] Yuxin Wu and Kaiming He. Group normalization, 2018. URL https://arxiv.org/abs/1803.08494.

[43] Pascal Mettes, Elise van der Pol, and Cees G. M. Snoek. Hyperspherical prototype networks, 2019. URL https://arxiv.org/abs/1901.10514.

[44] Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models, 2017. URL https://arxiv.org/abs/1702.03275.

[45] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[46] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

[47] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

[48] Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012. URL https://arxiv.org/abs/1212.5701.

[49] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. *Dive into deep learning*. Cambridge University Press, 2023.

[50] Jiaxuan Wang and Jenna Wiens. Adasgd: Bridging the gap between sgd and adam, 2020. URL https://arxiv.org/abs/2006.16541.

[51] Charles George Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, 1970.

[52] Roger Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322, 1970.

[53] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24 (109):23–26, 1970.

[54] David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24 (111):647–656, 1970.

[55] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[57] Ward Cheney and David Kincaid. Linear algebra: Theory and applications. *The Australian Mathematical Society*, 110: 544–550, 2009.

[58] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980. ISSN 00361429. URL http://www.jstor.org/stable/2156882.

[59] Francesco Mezzadri. How to generate random matrices from the classical compact groups, 2007. URL https://arxiv.org/abs/math-ph/0609050.

[60] Kai Hu and Barnabas Poczos. Rotationout as a regularization method for neural network, 2020. URL https://openreview.net/forum?id=r1e7M6VYwH.

[61] Wallace Givens. Computation of plain unitary rotations transforming a general matrix to triangular form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1):26–50, 1958. doi: 10.1137/0106004. URL https://doi.org/10.1137/0106004.

[62] Adelaide P Yiu, Valentina Mercaldo, Chen Yan, Blake Richards, Asim J Rashid, Hwa-Lin Liz Hsiang, Jessica Pressey, Vivek Mahadevan, Matthew M Tran, Steven A Kushner, Melanie A Woodin, Paul W Frankland, and Sheena A Josselyn. Neurons are recruited to a memory trace based on relative neuronal excitability immediately before training. *Neuron*, 83 (3):722–735, August 2014.

[63] Lingxuan Chen, Kirstie A Cummings, William Mau, Yosif Zaki, Zhe Dong, Sima Rabinowitz, Roger L Clem, Tristan Shuman, and Denise J Cai. The role of intrinsic excitability in the evolution of memory: Significance in memory allocation, consolidation, and updating. *Neurobiol. Learn. Mem.*, 173(107266):107266, September 2020.

[64] John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf.

[65] Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C. Love, Christopher J. Cueva, Erin Grant, Iris Groen, Jascha Achterberg, Joshua B. Tenenbaum, Katherine M. Collins, Katherine L. Hermann, Kerem Oktar, Klaus Greff, Martin N. Hebart, Nathan Cloos, Nikolaus Kriegeskorte, Nori Jacoby, Qiuyi Zhang, Raja Marjieh, Robert Geirhos, Sherol Chen, Simon Kornblith, Sunayana Rane, Talia Konkle, Thomas P. O'Connell, Thomas Unterthiner, Andrew K. Lampinen, Klaus-Robert Müller, Mariya Toneva, and Thomas L. Griffiths. Getting aligned on representational alignment, 2024. URL https://arxiv.org/abs/2310.13018.

[66] Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4738–4750. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf.

[67] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018. URL https://arxiv.org/abs/1802.08219.

[68] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas, Jun 2020. URL https://distill.pub/2019/activation-atlas/.

[69] Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups, 2021. URL https://arxiv.org/abs/2104.09459.

[70] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958. URL https://api.semanticscholar.org/CorpusID:12781225.

[71] R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107, Jun 2005. ISSN 1476-4687. doi: 10.1038/nature03687. URL https://doi.org/10.1038/nature03687.

[72] Katharine M Cammack, Thomas R Reppert, and Denise R Cook-Snyder. The simpsons neuron: A case study exploring neuronal coding and the scientific method for introductory and advanced neuroscience courses. *J Undergrad Neurosci Educ*, 20(1):C1–C10, December 2021.

[73] G Kreiman, C Koch, and I Fried. Category-specific visual responses of single neurons in the human medial temporal lobe. *Nat Neurosci*, 3(9):946–953, September 2000.

[74] Charles Sherrington. *Man on his nature*. 1940.

[75] Daniel J Graham and David J Field. Sparse coding in the neocortex. *Evolution of nervous systems*, 3:181–187, 2006.

[76] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. https://distill.pub/2018/building-blocks/, March 2018. [Accessed 02-08-2024].

[77] Charles G. Gross. Genealogy of the "grandmother cell". *The Neuroscientist*, 8(5):512–518, 2002. doi: 10.1177/107385802237175. URL https://doi.org/10.1177/107385802237175. PMID: 12374433.

[78] Charles E Connor. Neuroscience: friends and grandmothers. *Nature*, 435(7045):1036–1037, June 2005.

[79] Jerzy Konorski. Learning, perception, and the brain: Integrative activity of the brain. an interdisciplinary approach. *Science*, 160(3828):652–653, 1968. doi: 10.1126/science.160.3828.652. URL https://www.science.org/doi/abs/10.1126/science.160.3828.652.

[80] H B Barlow. Summation and inhibition in the frog's retina. *J Physiol*, 119(1):69–88, January 1953.

[81] Simon Thorpe. Local vs. distributed coding. *Intellectica*, 8(2):3–40, 1989.

[82] J. Y. Lettvin, H. R. Maturana, W. S. McCulloch, and W. H. Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11):1940–1951, 1959. doi: 10.1109/JRPROC.1959.287207.

[83] H. K. Hartline. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *American Journal of Physiology-Legacy Content*, 121(2):400–415, 1938. doi: 10.1152/ajplegacy.1938.121.2.400. URL https://doi.org/10.1152/ajplegacy.1938.121.2.400.

[84] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943. ISSN 1522-9602. doi: 10.1007/BF02478259. URL https://doi.org/10.1007/BF02478259.

[85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[86] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013. URL https://arxiv.org/abs/1311.2901.

[87] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. URL https://arxiv.org/abs/1312.6199.

[88] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization, 2015. URL https://arxiv.org/abs/1506.06579.

[89] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017. URL https://arxiv.org/abs/1702.03118.

A Beyond Isotropic Activation Functions

Activation functions are predominantly explored in this paper, showing how the isotropic functional form opens up a wealth of new functions to explore and design. However, isotropic deep learning principles are not limited to activation functions; a network is not isotropic until all constituent functions are reformulated using the provided conditions. Despite Bird [17]'s results, which empirically demonstrate representational anisotropies transforming under alterations in the activation function, it is also hypothesised that other primitives incur a similar basis-dependent representational alignment. It is expected that this is systematic to contemporary deep learning, as stated. This hypothesis is used to broaden the scope of the isotropic deep learning case study, encouraging an overhaul of almost all functional forms within modern-day deep learning. A preliminary audit of several functions is undertaken, then generalisations are proposed for isotropy. Similar can be replicated for various other symmetries. A systematic and extensive auditing is encouraged to establish symmetry categorisations and examine respective inductive biases.

A non-exhaustive list of functional forms requiring reformulation includes initialisers, normalisers, regularisers, operations, optimisers, losses, and gradient clipping. This section briefly summarises some of these, highlighting the standard anisotropies present in current forms. Isotropic reformulations will follow. These directions are currently incomplete, and future work will be required to develop and refine these functional forms through empirical benchmarking.

A.1 Normalisers:

Normalisation of the activations within deep learning is implemented frequently. This may be to prevent exponential gradient growth or decay, produce faster solution convergence, or smooth the loss landscape [39]. Initially, it was thought to primarily aid in reducing internal covariate shift [5]. In this section, their algebraic properties are analysed. Throughout the following, functional form anisotropy is defined over an arbitrary distribution, in accordance with algebraic or probabilistic equivariances, as opposed to anisotropy in any particular activation distribution — the latter is expected to still occur in isotropic networks due to symmetry breaking.

There have been many proposed normalisation techniques, including but not limited to "Batch Normalisation" by Ioffe and Szegedy [5], "Layer Normalisation" by Ba et al. [40], "Instance Normalisation" by Ulyanov et al. [41] and "Group Normalisation" by Wu and He [42]. The mathematics of each of these will be briefly outlined below. Similar approaches can be taken for other normalisations not listed. When the functions do not meet the isotropic requirements, it is not a suggestion that they are inherently suboptimal, as they were designed for purposes other than the principles stated for Isotropic deep learning. This analysis examines whether current normalisations can be classified as Isotropic deep learning methods and whether they introduce further incidental inductive biases in the representations beyond the intended distributional shifts.

Batch-normalisation [5] consists of normalising every element of the activation vector based upon mean and standard deviation statistics across a (mini-)batch, to produce a consistent activation distribution to train from. This was initially argued to reduce covariate shift, which otherwise may reduce learning due to phenomena such as saturation of activation functions. Batch normalisation also acted as a regulariser due to noise within the statistics and also enabled a higher learning rate, alongside fewer constraints on parameter initialisation. Expressed in multivariate form, *Eqn.* 42 shows the batch-normalisation operation, with trainable parameters $\vec{\gamma}$ and $\vec{\beta}$, where $\mathbb{E}_B\left[\cdots\right]$ indicates an average over the batch.

$$\mathbf{f}\left(\vec{x}\right) = \sum_{i=1}^{N}\left(\vec{\gamma}\cdot\hat{e}_i\right)\frac{\vec{x}\cdot\hat{e}_i - \mathbb{E}_B\left[\vec{x}\cdot\hat{e}_i\right]}{\sqrt{\epsilon + \mathbb{E}_B\left[\left(\vec{x}\cdot\hat{e}_i - \mathbb{E}_B\left[\vec{x}\cdot\hat{e}_i\right]\right)^2\right]}}\hat{e}_i + \vec{\beta} \tag{42}$$

Despite the appearance of many basis terms, $\hat{e}_i$, Batch Normalisation could be isotropic under strong and very restrictive assumptions in representation distribution — but is generally not algebraic nor probabilistically equivariant to orthogonal group actions. For example, if the activation distributions were a perfect normal distribution, with suitable isotropic initialisation of $\vec{\gamma}$ and $\vec{\beta}$, then batch-normalisation across multiple activations would produce an isotropic distribution, due to the product of zero-mean normal distributions being isotropic. However, even within isotropic deep learning, the activations are not expected to be isotropic nor constrained to be a zero-mean normal distribution. Therefore, a per-coordinate rescaling would likely incur anisotropy counterintuitively. This is because covariance is neglected for efficiency, inducing a basis dependence and anisotropy. This approach also makes batch normalisation sensitive to the mini-batch size and also *not* isotropic in any degree.

The dependence on mini-batches can be reformulated in an alternative manner. One can represent batch normalisation through a matrix of activations: $\mathbf{X}\in\mathbb{R}^{b\times n}$, where b represents the batch size and n represents the number of features. The mean-statistic consequently becomes $\vec{\mu}_j = \mathbf{X}_{ij}\vec{1}_i/n$, in the standard basis and using Einstein summation convention for compactness. With subtraction of the mean, $\mathbf{X}'_{ij} = \mathbf{X}_{ij} - \vec{\mu}_j\vec{1}_i/n$, and then normalisation using $\sigma_j^2 = \mathbf{X}'_{ij}\mathbf{X}'_{ij}$ $(+\epsilon)$. One can see that the resultant manifold of possible representations is first constrained to a plane orthogonal to $\vec{1}_i$, a $\mathbb{R}^{(b-1)\times n}$ space, then normalisation approximatly produces an $\mathcal{S}^{(b-2)}\times\mathbb{R}^n$ space (if ϵ is ignored), embedded in the original $\mathbb{R}^{b\times n}$ space. The parameters $\vec{\gamma}$ amd $\vec{\beta}$ produce a different embedding. Consequently, one can see that a single sample activation space is preserved at $\mathbb{R}^n$. However, across the batch, the space is $\mathcal{S}^{(b-2)}\times\mathbb{R}^n$. This demonstrates how the stochasticity in batch sampling results in a change to the batched-activation space, producing regularising stochasticity that affects gradients. Despite batch normalisation being a function $\mathbf{f}:\mathbb{R}^{b\times n}\to\mathbb{R}^{b\times n}$ for its extrinsic space, the effect on its intrinsic manifolds will be denoted $\mathbf{f}:\mathbb{R}^{b\times n}\to\mathcal{S}^{b-2}\times\mathbb{R}^n\hookrightarrow\mathbb{R}^{b\times n}$. This makes the matrix-valued nature of the function evident.

Layer-normalisation [40] consists of a similar approach in normalising activations; however, the statistics are calculated differently. Layer normalisation acts sample-wise within a batch, removing the dependence on mini-batch stochasticity and batch size, thereby returning the normalisation to a vector-valued form. Alongside other properties, this sample independence has made it a popular choice as a normaliser. It is given by the function in *Eqn.* 43, where $\mathbb{E}_i\left[\cdots\right]$ is the expectation over index i occuring in the basis vectors $\hat{e}_i$.

$$\mathbf{f}\left(\vec{x}\right) = \sum_{i=1}^{N}\left(\gamma\frac{\vec{x}\cdot\hat{e}_i - \mathbb{E}_i\left[\vec{x}\cdot\hat{e}_i\right]}{\sqrt{\epsilon + \mathbb{E}_i\left[\left(\vec{x}\cdot\hat{e}_i - \mathbb{E}_i\left[\vec{x}\cdot\hat{e}_i\right]\right)^2\right]}} + \beta\right)\hat{e}_i \tag{43}$$

Despite this function producing $\mathbf{f}:\mathbb{R}^n\to\mathbb{R}^n$, the intrinsic dimensionality of an $\mathbb{R}^n$ dimensional object is mapped to an $(n-2)$-sphere, $\mathcal{S}^{n-2}$, embedded within an $\mathbb{R}^n$ space: $\mathbf{f}:\mathbb{R}^n\to\mathcal{S}^{n-2}\hookrightarrow\mathbb{R}^n$. This is similar to batch normalisation, but it directly affects the representational geometry of individual samples. This loses two degrees-of-freedom in the *activation* vectors, and these degrees of freedom are not reintroduced simply by the two degrees-of-freedom gained in the *parameters*: γ and β. The latter affect the manifold globally by altering the embedding: $\mathbf{f}\left(\vec{x}\right)\cdot\vec{1} = n\beta$ (an equation for a hyperplane with normal $\hat{1}$) and $\left\|\mathbf{f}\left(\vec{x}\right) - \beta\vec{1}\right\|_2 = \gamma$ (constraining the vector norm within the hyperplane). Therefore, gaining two degrees of freedom in the parameters does not constitute a replacement of representational degrees of freedom. These are global degrees of freedom in the embedding vs local degrees of freedom in the structure of the distribution. By definition, this cannot be isotropic due to a breakdown of rotational equivariance when rotating into the plane's normal direction. It exhibits a broken equivariance: $\mathrm{O}\left(n-1\right)\subset\mathrm{O}\left(n\right)$, additionally with a translation right-invariance by $+\lambda\vec{1}$ and scale right-invariance $\lambda\mathrm{I}_n$ connected to how the representational degrees-of-freedom are lost. Furthermore, the implementation of β and γ is basis-dependent. Interestingly, any weighted mean, and its corresponding $\beta\hat{n}$, could be chosen to have a similar effect (ignoring the effect of other primitives on the model). This includes the extreme of one-hot weighted means — this may be a surprising insight yielded by representational geometry considerations.

The success of layer normalisation can be reinterpreted as a map with the loss of information in two directions, an information bottleneck much like an autoencoder, which can be optimised to remove redundant data. This may explain an additional benefit of layer normalisation in classification networks. This principle could be generalised by reinterpreting layer-norm as a sequence of three isotropic layers producing a small bottleneck: $\mathbb{R}^n\to\mathbb{R}^{n-2}\to\mathbb{R}^n$. If an isotropic activation function is applied on $\mathbb{R}^{n-2}$, such as $\mathbf{f}\left(\vec{x}\right) = \vec{x}/\sqrt{\vec{x}\cdot\vec{x}} = \hat{x}$, or the implicit normalisation properties of isotropic-$\tanh$ are used, then one can largely recreate the form of layer-norm architecturally. However, the weight and bias parameters of $\mathbb{R}^{n-2}\to\mathbb{R}^n$ layer now act as a basis-free substitute for γ and β. This basis independence is essential under the isotropic framework's approach to reducing unintentional representational inductive biases. One can also further constrict the bottleneck if desirable. This will be termed *isotropic layer-norm*, and could be used as a drop-in replacement within existing classification models, which may especially benefit from removing redundant directions. It features a probabilistic equivariance to orthogonal group actions, and is spontaneous symmetry broken into an algebraic equivariance to a factored-orthogonal group action. This approach also blurs the distinction between what constitutes an activation function and a normaliser, perhaps the latter may be distinguished as a function which reduces (projects out) representational degrees-of-freedom and/or contains a time-like or data-like probabilistic relation.

Instance normalisation [41], arose in convolutional neural networks for style transfer. It is defined in *Eqn.* 44, where the basis-dependence is expressed through indices for brevity, with notation aligning with Ulyanov et al. [41].

$$\mathbf{f}\left(\vec{x}\right)_{tijk} = \gamma_i\frac{x_{tijk} - \left[\mathbb{E}_{hw}\left[x_{tihw}\right]\right]_{ti}}{\sqrt{\epsilon + \left[\mathbb{E}_{hw}\left[x_{tihw} - \left[\mathbb{E}_{hw}\left[x_{tihw}\right]\right]_{ti}\right]^2\right]_{ti}}} + \beta_i \tag{44}$$

Analysing the isotropic properties of this normalisation is made difficult due to the use of convolution. The axiomatic-like approach to isotropy argues that the symmetries arise from architectural symmetries of the underlying model, discussed briefly in *Sec.* 5 and *App.* E.1. This approach suggests that an orthogonal automorphism can exist channelwise. Hence, if one chooses an isotropic equivariance relation, it would be restricted to rotations within this linear subspace, indexed per spatial pixel, of the entire activation space. Hence, isotropic forms can be applied over the channel-wise vectors, $\mathbb{R}^C$ for C channels, to which the algebraic rotational equivariance applies.

This would suggest that instance normalisation could have an algebraic factored-isotropic equivariance, particularly when $\beta_i = 0$ and $\gamma_i = \gamma$, but it is not isotropic in its current form, nor factored-orthogonal. Similarly, one could reinterpret this as a linear single-layer-like transform, with standardisation followed by $\mathbf{W} = \mathrm{diag}\left(\vec{\gamma}\right)$ — highlighting its current standard-basis dependence and standard permutation anisotropy. One could apply probabilistic invariances to β_i and γ_i for isotropy and apply them to a weighted average as well.

Group normalisation [42] can be understood as a middle ground between Layer normalisation and Instance normalisation, as it is argued that channels are not statistically independent. Hence, the statistics are computed along these groups of channels. Since it is a middle ground between both Instance normalisation and Layer normalisation, arguments for its anisotropy are trivially extended.

A.1.1 Potential Isotropic Normalisers:

Alongside the isotropic layer norm proposed above, several other formulations can be derived. These consist of a batch-norm-like approach, to be termed 'Chi-normalisation', a time-like normalisation, and a simpler isotropic layer norm than proposed above. Normalisations are the most analogous to activation functions in terms of symmetry equivariance constraints: $[\mathbf{R}, \mathbf{f}] = 0$, per $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$. This makes them a reasonably straightforward candidate for reformulation. Despite this, due to the presence of parameters, a probabilistic condition must also apply across their initialisation distributions, and is implicit in all subsequent definitions. This is discussed further in *Sec.* 5.2 and then *App.* A.2. The full possibilities of normalisation within isotropic deep learning are not limited to these few suggestions.

Isotropic deep learning is also suggested to make continuous directions meaningful, and magnitudes indicate the degree of that meaning. Large growing magnitudes may destabilise training in the networks, so it is sensible to normalise over the magnitude — analogous to existing normalisation techniques.

A simplified layer-normalisation than the one stated above is: $\mathbf{f}(\vec{x}) = \gamma\hat{x} + \vec{\beta}$, where $\hat{x}$ is the unit-normalised vector of $\vec{x}$. The increased parameters and collapse of redundant directions may make the previous form preferable. Probabilistic considerations should be made for the parameters.

Second, it is termed a 'Chi-normaliser' and is computed over a batch, which is acknowledged to have its drawbacks. After a linear transform is applied to prior activations, it could be assumed that the resulting activation distribution is approximately a multivariate normal distribution. Therefore, one could subtract the mean of the distribution across a batch and then normalise the magnitudes. The magnitudes of an n-dimensional, standard multivariate normal distribution follow the Chi distribution, shown in *Eqn.* 45 with its mean and variance, for $x \geq 0$ and Γ being the gamma-function. The probabilistic considerations also apply to the data in such instances. This enables the dataset to perform a spontaneous symmetry breaking of the network.

$$f(x;n) = \frac{x^{n-1}e^{-\frac{x^2}{2}}}{2^{\frac{n}{2}-1}\Gamma\left(\frac{n}{2}\right)}, \quad \mathbb{E}\left(f(x;n)\right) = \sqrt{2}\frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}, \qquad \mathrm{Var}\left(f(x;n)\right) = n - 2\mu. \tag{45}$$

Therefore, the mean and standard deviation statistics could be estimated over the batch, and the magnitude distribution normalised using these. Then, the scale of the resultant chi-like distribution must be standardised. This is achieved by dividing by the mean of the distribution. The extra factor of $\sqrt{n}$ is then absorbed into the scaling parameter γ. For batched activations $\mathbf{X} \in \mathbb{R}^{b\times n}$, with elements indexed and using Einstein summation convention, the Chi-normalisers is given by *Eqn.* 46 and *Eqn.* 47. This form may require further refinements, yet the Chi-like normalisation approach is proposed as a potentially useful new form of normaliser.

The appearance of $\vec{1}$ applies over the batch, so it does not result in an explicit sample-wise symmetry-breaking. One could experiment with other probabilistic-invariant weightings in the average of *Eqn.* 46 as a fibre-bundle-like consideration over the samples if practical. Similar for the $\beta\hat{n}$ instead of $\beta\hat{1}$.

$$\mathbf{X}'_{ij} = \mathbf{X}_{ij} - \frac{\mathbf{1}_{is}\mathbf{X}_{sj}}{n} \tag{46}$$

$$\mathbf{f}(\mathbf{X}') = \frac{\gamma\mathbf{X}'_{ij}}{\sqrt{\|\mathbf{X}'_{sm}\mathbf{X}'_{sm} + \epsilon\|}} + \vec{1}_i\vec{\beta}_j \tag{47}$$

One may reshape the width of the chi-like distribution, using a further statistic, such that it follows the intended standard chi-distribution more closely. Further normalisations may be non-trivial due to ensuring positive domain operations, which subtraction violates. This direction change would be highly undesirable, but the magnitudes can be rectified using $\max(0, x)$ or an alternative.

Additionally, projecting onto the surface of a hypersphere by taking a vector norm may act as a normaliser, but it loses the magnitude degree of freedom. This has been implemented on output layers [43] and may be explored in terms of an alternative to one-hot (local coding) outputs for isotropic networks.

Finally, one may estimate magnitude-normalising statistics by a moving average over training steps [44]. This is a time-like normalisation, which could also be computed over the batch if desirable. This can arguably result in stale statistics due to the changing representations by optimisation, but results may differ from previous attempts by strictly normalising the magnitude. It is *not* proposed that gradients should be propagated through time; statistics could be assumed to be approximately constant for efficiency. Storing such statistics is negligible in terms of memory requirements, as they would only be scalars per normalisation layer. These can be updated such that they follow an exponential weighted average. Consequently, more recent activations would contribute more to estimating the statistics, reducing the stale statistics problem. This direction would need considerably further exploration. Similar stale statistics are also found to be beneficial in optimisers such as in *App.* A.3, so may not inherently be a problem if other aspects of the normalisation are well-constructed.

These preliminary normalisers are suggested as a starting point for testing and building new isotropic normalisers.

A.2 Initialisers:

Parameter initialisation may also have unintended consequences for representational geometry. For example, in the extreme case, if one initialises all weights as rank-1 matrices, then we may expect both worse-performing and slower-learning networks due to lower expressivity and difficulties with gradient flow. Therefore, initialisation considerations are essential for representational geometric inductive biases.

Producing Isotropic constraints on initialisation requires analysis of how the isotropic symmetry is said to 'derive' from the architecture. This is discussed briefly in *Sec.* 5 and *App.* E.1. In effect, under the construction discussed, an arbitrary

architecture is said to be invariant to continuous rotations of its nodes, after fields are assigned to them, but before functional forms are introduced to the architecture. Only the initialisation *distribution* is made left/right-invariant to a symmetry under this construction. Thus, sampling the parameters does produce a framework-acceptable spontaneous symmetry breaking. Moreover, this spontaneous symmetry breaking is required for network learning and functionality in both parameter and activation distributions. The graph-automorphisms only constrain forms of the probability distribution through the chosen symmetry. Initialisers are also chosen to allow full-rank parameters.

The set of n nodes with field $\mathbb{F}$ assigned to them is said to form a $\mathbb{F}^n$ representation space. This space has various applicable automorphisms, but is chosen to be invariant to $\mathrm{O}(n)$ group actions for this case study into isotropic deep learning. Following this approach, the initialisers producing parameters which transform between sequential spaces should also be invariant to rotations of the underlying spaces. Therefore, a relation such as $\forall \mathbf{R} \in \mathrm{O}(n)$ and $\forall \mathbf{T} \in \mathrm{O}(m)$ then: $\mathbf{R}\mathbb{P}_{\mathbf{W}}\mathbf{T} = \mathbb{P}_{\mathbf{W}}$ is desirable, for parameter $\mathbf{W} \in \mathbb{R}^{n\times m}$ and probability distribution $\mathbb{P}_{\mathbf{W}}$ over $\mathbb{R}^{n\times m}$. This is displayed more formally in *Eqn.* 48. In this case, the initialiser is invariant to the same symmetry as the equivariance in the nodes, and does not induce standard-anisotropic artefacts by an unintended inducive bias.

$$\forall \mathbf{W} \in \mathbb{R}^{n\times m}, \forall \mathbf{R} \in \mathrm{O}(n), \mathbf{T} \in \mathrm{O}(m) \quad \mathbb{P}(\mathbf{RWT}) = \mathbb{P}(\mathbf{W}) \tag{48}$$

Many such initialisers could be formulated; in particular, two will be suggested: semi-orthogonal [45] and multivariate normal. If the product measure for $\mathbb{R}^{nm}$ is composed of independent standard normal distributions, $\mathbb{R}$, then the overall distribution is rotationally symmetric, meeting requirements. Orthogonally, one could draw two elements *uniformly* from groups $\mathrm{O}(n)$ and $\mathrm{O}(m)$ and use their $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{m\times m}$ matrix representations respectivly. These distributions could then be joined together using a $\Lambda \in \mathbb{R}^{m\times n}$ matrix drawn uniformly from rectangular-diagonal matrices — however, this could be restricted to the δ_{nm} matrix, an identity matrix padded with appropriate zeros to be $n \times m$ in shape. The distributions joined using $\mathbf{W} = U\Lambda V$ would then be probabilistically left/right invariant to orthogonal group actions, and thus considered isotropic. This list of possible initialisers is not exhaustive.

This leaves some free parameters available within these isotropic distributions: σ for the multivariate normal and a 'gain' factor for orthogonal. These could be chosen using a similar approach to Glorot and Bengio [46] for isotropic deep learning. This could ensure that the expectation of magnitudes between layers remains relatively constant and provides an appropriate constraint on gradients in isotropic networks. Future work would need to establish how these values should be set in line with the principles of Isotropic deep learning.

A.3 Optimisers:

Both stochastic gradient descent and momentum variants do not contain a basis dependence in their formulations; however, many adaptive optimisers do. For example, Adagrad by Duchi et al. [47] shown in *Eqn.* 49, AdaDelta by Zeiler [48] shown in *Eqn.* 50 and ADAM by Kingma and Ba [7] shown in *Eqn.* 51, all use approximations which are basis-dependent. A basis dependence within the optimiser will likely optimise preferentially in specific directions, producing an anisotropic effect on parameters. This anisotropic preference in parameters then shapes the distribution of activations through the network. Hence, it is expected that anisotropy in an optimiser results in a form of representational bias too. Since this is an indirect effect, such a bias may be non-trivial, obscuring its basis-dependent effect. An isotropic optimiser would remove these biases. They also can be grouped, and defined, through similar relations to those found in *Sec.* 5.2.

In the following equations, parameters at time-step t are indicated by θ, a per-coordinate gradient at time-step by g_t, and learning rate η — broadly consistent with Zhang et al. [49] notation when comparing the algorithms.

Adagrad optimiser:

$$\begin{aligned} \theta_{t+1} &= \theta_t - \frac{\eta g_t}{\sqrt{s_t + \epsilon}} \\ s_{t+1} &= s_t + g_t^2, \quad s_0 = 0 \end{aligned} \tag{49}$$

AdaDelta optimiser:

$$\begin{aligned} \theta_{t+1} &= \theta_t - g'_t \\ g'_t &= \frac{\sqrt{\Delta x_{t-1} + \epsilon}}{\sqrt{s_t + \epsilon}} g_t \\ \Delta x_t &= \rho \Delta x_{t-1} + (1-\rho) g_t'^2, \quad \Delta x_0 = 0 \\ s_t &= \rho s_{t-1} + (1-\rho) g_t^2, \quad s_0 = 0 \end{aligned} \tag{50}$$

ADAM optimiser:

$$\begin{aligned} \theta_{t+1} &= \theta_t - \eta g'_t \\ g'_t &= \frac{\tilde{v}_t}{\sqrt{\tilde{s}_t} + \epsilon} \\ v_t &= \beta_1 v_{t-1} + (1-\beta_1) g_t, \quad v_0 = 0 \\ s_t &= \beta_2 s_{t-1} + (1-\beta_2) g_t^2, \quad s_0 = 0 \\ \tilde{v}_t &= \frac{v_t}{1-\beta_1^t} \\ \tilde{s}_t &= \frac{s_t}{1-\beta_2^t} \end{aligned} \tag{51}$$

Within *Eqn.* 49, this can be seen through the coordinate-wise accumulated squared-gradient value. Similar is true for *Eqn.* 50 and *Eqn.* 51. Thus, a decomposition along the standard basis is used in this accumulation, resulting in a diagonal approximation that introduces anisotropy through its implementation.

However, in Wang and Wiens [50] their optimiser is ADAM-like with isotropic definitions. This AdamSGD algorithm is displayed in *Eqn.* 52. This may be a starting point for a more optimal isotropic adaptive optimiser.

$$\begin{aligned} \theta_{t+1} &= \theta_t - \eta_t m_t \\ \eta_t &= \eta \sqrt{\frac{1-\beta_2^t}{v_t / \dim \theta}} \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) \left\| g_t \right\|_2^2, \quad v_0 = 0 \\ m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad m_0 = 0 \end{aligned} \tag{52}$$

Alternatively, a return to an inverse-Hessian approximating quasi-Newton algorithms such as BFGS [51, 52, 53, 54] or L-BFGS [55], may be another approach. Gradient clipping is another operation that requires consideration, as its anisotropy can produce a representational bias. The author is actively researching both directions.

A.4 Operations:

Several new operations can also be defined within the Isotropic framework; these include min-like and max-like functions, as well as a new multiplication operation. A link demonstrating these is available at `https://www.desmos.com/calculator/ttmvis7av4`.

The standard minimum function and maximum function are displayed in *Eqns.* 53 and 54 respectively, for a function $\mathbf{f} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$.

$$\mathbf{f}\left(\vec{x}, \vec{n}\right) = \min\left(\vec{x}, \vec{n}\right) = \sum_{i=1}^{N} \min\left(\vec{x} \cdot \hat{e}_i, \vec{n} \cdot \hat{e}_i\right) \hat{e}_i \tag{53}$$

$$\mathbf{f}\left(\vec{x}, \vec{n}\right) = \max\left(\vec{x}, \vec{n}\right) = \sum_{i=1}^{N} \max\left(\vec{x} \cdot \hat{e}_i, \vec{n} \cdot \hat{e}_i\right) \hat{e}_i \tag{54}$$

These functions are applied elementwise to components of $\vec{x}$ and $\vec{n}$ as indicated by the sum over standard basis directions, $\hat{e}_i$. This basis dependency is an example of an inductive bias which affects representations. These functions have two arguments, which require generalisation of the framework to multi-argument cases. One way to generalise such an isotropic equivariance is by applying the group action to both arguments in a modifed equivariance relation: $\mathbf{R}\mathbf{f}\left(\vec{x}, \vec{n}\right) = \mathbf{f}\left(\mathbf{R}\vec{x}, \mathbf{R}\vec{n}\right)$, for all $\mathbf{R} \in \mathrm{O}\left(n\right)$. Accordingly, one can define *Eqns.* 55 and 56 as functional forms which follow this relation. In the case where $\vec{n} \cdot \vec{x} = 0$, the operation should be defined as the identity map.

$$\mathbf{f}\left(\vec{x}, \vec{n}\right) = \min\left(\operatorname{sign}\left(\vec{x} \cdot \vec{n}\right), \left|\frac{\vec{n} \cdot \vec{n}}{\vec{x} \cdot \vec{n}}\right|\right) \operatorname{sign}\left(\vec{x} \cdot \vec{n}\right) \vec{x} \tag{55}$$

$$\mathbf{f}\left(\vec{x}, \vec{n}\right) = \max\left(\operatorname{sign}\left(\vec{x} \cdot \vec{n}\right), \left|\frac{\vec{n} \cdot \vec{n}}{\vec{x} \cdot \vec{n}}\right|\right) \operatorname{sign}\left(\vec{x} \cdot \vec{n}\right) \vec{x} \tag{56}$$

These operations are a suggestion for such an isotropic alternative, but may not be optimal. They clip vectors which cross a hyperplane boundary defined by the choice of $\vec{n} \in \mathbb{R}^N$; the hyperplane equation is: $\vec{x} \cdot \vec{n} = \left\|\vec{n}\right\|_2^2$. The minimum function preserves the coordinates of all samples on the origin side of the hyperplane and projects all other coordinates onto a hyperplane. This projection is carefully chosen so that the projected point remains on a line passing through the origin and the original coordinates. Consequently, neural refraction does *not* occur on the plane boundary for origin-intersecting lines, like it does in the standard functions. The maximum function computes the opposing case, where points are kept constant on the far side of the hyperplane, which does not include the origin. In this case, points within the other sector are similarly projected onto the hyperplane.

Similarly, Hadamard multiplication, also termed elementwise multiplication, is inherently basis-dependent. This can be seen through its basis dependence, through $\hat{e}_i$, in *Eqn.* 57. This is used in settings such as the LSTM gates mechanism [56], also one can reinterpret the dropout-mask in such a form [2], alongside many more applications.

$$\mathbf{f}\left(\vec{x}, \vec{n}\right) = \vec{x} \otimes \vec{n} = \sum_{i=1}^{N} \left(\vec{x} \cdot \hat{e}_i\right) \left(\vec{n} \cdot \hat{e}_i\right) \hat{e}_i \tag{57}$$

One interpretation of *Eqn.* 57 is that it scales each component of $\vec{x}$ by each component of $\vec{n}$, such that each axis in the standard basis is rescaled. This is equivalent to premultiplying $\vec{x}$ with a diagonal matrix $\mathbf{W} = \operatorname{diag}\left(\vec{n}\right) \in \mathbb{R}^{N \times N}$, where diag decomposes $\vec{n}$ along the standard basis and puts these components along the diagonal of a square matrix. This axis-wise rescaling is anisotropic, particularly a two-component algebraic permutation equivariance similar to before.

To reproduce a similar behaviour, one could use isotropic-multiplication shown in *Eqn.* 58. This rescales the component of $\vec{x}$ which lies in the $\vec{n}$ direction by an amount determined by $\left\|\vec{n}\right\|$.

$$\mathbf{f}\left(\vec{x},\vec{n}\right)=\vec{x}+\left(\left(\|\vec{n}\|-1\right)\vec{x}\cdot\hat{n}\right)\hat{n} \tag{58}$$

These are just examples and may not be a simple drop-in replacement for existing operations, due to the differences in how the operations act. Initial anisotropic operations rescale in multiple axes/hyperplanes at once, whereas the isotropic operations only act in single directions. This may make them suboptimal as gate mechanisms, such as in LSTMS, since they can only collapse the representations in one direction at a time: $\mathbf{f}:\mathbb{R}^N\times\mathbb{R}^N\to\mathbb{R}^{N-1}\hookrightarrow\mathbb{R}^N$. These single-direction gates may be limited in how they reshape the representations and have more significant computational cost; therefore, further development is undoubtedly needed. These specific implementations may require a custom isotropic operation that fully replicates their desired effect in an isotropic and basis-independent manner.

Finally, there may be some instances where representations must be clipped into a hypercube shape. This is *not* an isotropic operation, nor basis independent; however, this formulation does reduce neural refraction, so it is included for convenience. Using *Eqn.* 59 results in substantial neural refraction along the boundary of the hypercube: lines passing through the origin are projected across the boundary, substantially changing direction.

$$\mathbf{f}\left(\vec{x}\right)=\max\left(\min\left(\vec{x},1\right),-1\right)=\sum_{i=1}^{N}\max\left(\min\left(\vec{x}\cdot\hat{e}_i,1\right),-1\right)\hat{e}_i \tag{59}$$

Using *Eqn.* 60 achieves the same result without neural refraction at the boundary. Any coordinate which is outside the boundary is projected back onto the boundary, along a line between the origin and the original coordinate. As a result, direction is preserved. An affine transform can recenter this box at arbitrary coordinates and reshape the box due to the linear transform.

$$\mathbf{f}\left(\vec{x}\right)=\min\left(\frac{1}{\max_i\left(\vec{x}\cdot\hat{e}_i\right)},1\right)\vec{x} \tag{60}$$

B Quasi-Isotropic Functional Forms

A middle ground, following from *Sec.* 4, balancing the predicted problems of anisotropy whilst enabling representation compression, not just through the bias, is to relax the hard isotropy condition and introduce slight symmetry breaking in many directions. This can be achieved using many small perturbations to the direction unit vector only, producing a softer symmetry breaking. Then the network has many distinguished vectors, a subset with which it may align its representations in a task-dependent manner. Therefore, it does not favour a particular basis, but still introduces some desirable consequences of anisotropy, for problems such as classification. If one further restricts the functions from featuring dynamic refraction, it limits detrimental anisotropic effects. This recovers an arguably more principled and slightly increased basis-independent form of anisotropy.

One method is to apply a non-linearity based on rounding the vector's directions. This is shown in *Eqn.* 61, where $[\cdot]$ indicates the rounding operation and $\phi(\vec{x}) \neq \vec{x}$. In fact, the anisotropic perturbation may be implemented as simply as: $\phi(\vec{x}) = \beta\vec{x}$ for $\beta \neq 1$. The overall angular term is approximately unit-normed, but can be trivially modified to be exactly norm-1.

$$\Phi(\hat{x}; \alpha) = \frac{[\alpha\hat{x}]}{\alpha} + \phi\left(\hat{x} - \frac{[\alpha\hat{x}]}{\alpha}\right) \approx \hat{x} \tag{61}$$

This produces a quasi-isotropic functional form shown in *Eqn.* 62, with an isotropy-breaking parameter α. Slight anisotropic refraction is added, independent of magnitude, so it is predictable and thus extrapolatable to the network. Due to the angular rarefaction and compression by the proposed non-linearity, representation over- and underdensities may then occur, where semanticity may begin to be assigned. However, for $\alpha \to \infty$, isotropy is continuously reintroduced and could be an optimisable parameter. Such an approach may be beneficial to contrastive learning methods [36, 37].

$$\mathbf{f}(\vec{x}) = \sigma(\|\vec{x}\|)\,\Phi(\hat{x}) \tag{62}$$

Similarly, one could construct *Eqn.* 63 as another form of quasi-isotropic activation functional forms. In this case, a finite set of unit-vectors $\hat{b}_i$ is distributed over a lattice across $\mathcal{S}^{n-1}$ in $\mathbb{R}^n$ and a discrete rotation group transforms between various of these unit-vectors. This method may be less computationally efficient than the aforementioned method due to the numerous dot-product evaluations; however, it remains faithful to the Ψ_n hierarchical group structure. Moreover, one can recover a B_n symmetry without neural-refraction with this method, which may be desirable.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}) = \frac{\sigma(\|\vec{x}\|)\,\hat{x}}{\max\limits_{\hat{b}_i}\left(\hat{b}_i \cdot \hat{x}_i\right)} \tag{63}$$

It is likely that many other functional forms could be considered.

B.1 Parameterised Probabalistic-Isotropy

Following from *Eqn.* 63, an alternative activation functional form can be constructed which appears similar but uses trainable $\hat{b}_i$ vectors, which notationally can be stacked into a matrix $\hat{\mathbf{Z}}_{ij} \in \mathbb{R}^{m\times n}$.

Following an appropriate initialisation of trainable parameters $\hat{\mathbf{Z}}_{ij}$, as discussed in *App.* A, the functional form can become weakly-isotropic. Yet it can undergo spontaneous symmetry breaking to reproduce desirable behaviours of anisotropy whilst preventing phenomena like neural refraction. Whether the denominator terms $\hat{x}$ and $\hat{\mathbf{Z}}$ should be unit-normalised can be evaluated, as well as the specific use of a max function.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}) = \frac{\sigma(\|\vec{x}\|)\,\hat{x}}{\max\limits_{i}\left(\hat{\mathbf{Z}}_{ij}\hat{x}_j\right)} \tag{64}$$

C Stochastic Isotropy — Producing Immediate Anisotropic Analogues

One method to approximate isotropy with current functions is to stochastically choose a basis on which the anisotropic function operates. This enables anisotropic functions to be used in an isotropic network, without inducing a representational alignment to an arbitrary basis. The example is a training-time probabilistic condition; each batch exhibits spontaneous symmetry breaking, which 'averages out' over multiple batches. This approach can be generalised, and its overall utility assessed.

For example, current (anisotropic) dropout by Srivastava et al. [2] appears to privilege the basis anti-aligned with the standard basis, thereby maximally preserving information when a direction of the standard basis is collapsed. So it is expected to incur an arbitrary basis dependence in the representations. However, anisotropic dropout can be applied to a stochastically chosen basis. This randomness is hypothesised to prevent a representational-anisotropy induced by an arbitrarily chosen basis.

This can be achieved by producing a basis, uniformly drawn from the layer's orthogonal symmetry: $\mathbf{B} \sim \mathrm{SO}(n)$[10]. There are several methods to produce a uniform random matrix, each with varying computational costs. These include the exponentiation of a Lie generator scaled by an appropriately drawn random variable, the Gram-Schmidt procedure [57], and many others [58, 59]. The below matrix-multiplication procedure is computationally cumbersome; simpler formulations may be desirable. Hence, the following proposed forms are only a starting point for converting anisotropic functions directly to probabilistically-isotropic forms. In practice, isotropic functions should be constructed from the ground up rather than merely analogous functions converted from existing anisotropic ones. Therefore, this remains a placeholder, but with some interesting extensions.

For the example of standard dropout, shown in *Eqn.* 65, it can be made stochastically-isotropic by including the basis-transform shown in *Eqn.* 66. Where $\vec{x}$ is the activation vector, with normalisation factor S_a and S_{si}, dropout-mask $M_i = \vec{M} \cdot \hat{e}_i$ and standard-basis vectors $\hat{e}_i$.

$$\vec{x}' = S_a \sum_{i=1}^{N} M_i \left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i \tag{65}$$

$$\vec{x}' = S_{si} \sum_{i=1}^{N} \left(\mathbf{B}\vec{M} \cdot \hat{e}_i\right) \left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i \tag{66}$$

Similar formulations, such as RotationOut by Hu and Poczos [60], generally improved performance when the basis is stochastically rotated. However, this method remains stochastically anisotropic as the rotations are generated through Given's rotations [61], which is not uniform over the space of orthogonal matrices — a necessity for full stochastic-isotropy. Nevertheless, the implementation by Hu and Poczos [60] is somewhat encouraging.

This procedure can be generalised and applied to any existing anisotropic function. Yet, as stated, it is generally preferable to construct an isotropic function from first principles rather than relying on stochastic isotropy, which may be computationally costly.

C.1 Considering Correlating the Stochastic-Isotropy

Correlating the random bases in time would be a curious extension, particularly to stochastically isotropic dropout. This may produce a time-like structure in the embedded activation distribution of a network. This also constitutes an initialisation-based symmetry breaking, since a random walk may not evenly cover the space in practice and its starting point may bias training. Nevertheless, this is a suggested extension that is considered interesting and informative.

If one imagines a random walk in the Lie-algebra space for *rotation* matrices (no mirror action): $\mathrm{SO}(n) \ni \mathbf{R}^{(\mathbf{t}+\mathbf{dt})} = \mathbf{R}^{(\mathbf{t})}\delta\mathbf{R}$, with $\delta\mathbf{R} = e^{\mathfrak{r}\cdot\vec{n}}$, with $\mathfrak{r}$ being the corresponding (normalised) anti-symmetric generators for rotations and $\vec{n} \sim \mathcal{N}(\vec{0}, \sigma \mathrm{I}_n)$ with $0 < \sigma \ll 1$. This procedure results in a random walk of the rotation matrix at each time step.

Following this, a time-correlated Bernoulli distribution can be defined. Beginning with $\vec{D}^{(0)}$, divide up the layer of neurons into two sets: inactive $I_n = \left\{i | \vec{D}_i^{(n-1)} = 0\right\}$ and active $A_n = \left\{i | \vec{D}_i^{(n-1)} = 1\right\}$. Then we have two hyper-parameters: the standard dropout probability λ and an overlap probability Γ, such that $|A_n|\, q + |I_n|\, \Gamma = (|A_n| + |I_n|)\, \lambda$ — where q is not a free parameter. If $|A_n| = 0$ or $|I_n| = 0$, then temporarily define $q = \Gamma = \lambda$. If not, one must prevent unnormalised probabilities as shown in *Eqn.* 67.

$$\Gamma = \max\left(0, \max\left(\lambda + \frac{|A_n|}{|I_n|}\left(\lambda - 1\right), \min\left(1, \min\left(\lambda + \frac{|A_n|}{|I_n|}\lambda, \Gamma\right)\right)\right)\right) \tag{67}$$

Leading to $q = \lambda + \frac{|I_n|}{|A_n|}\left(\lambda - \Gamma\right)$. Then use one Bernoulli function across all active neurons using $\mathbb{R}^{|A_n|} \ni \vec{D}_{(A)}^{(n)} \sim \mathrm{BernoulliDist.}^{|A_n|}(q)$ likewise for inactive neurons $\mathbb{R}^{|I_n|} \ni \vec{D}_{(I)}^{(n)} \sim \mathrm{BernoulliDist.}^{|I_n|}(\Gamma)$. Therefore, correlating the inactive 'neurons'[11] across the time steps, whilst still introducing a degree of random dropout. Thus, the 'basis of dropout' undergoes a random walk at every time step, and 'neurons' are randomly chosen to be dropped from the network, with a differing likelihood if they were just previously dropped. The coherence time can be adjusted through Γ for the specific time-dependent task needed.

[10] Generating $\mathrm{SO}(n)$ may be computationally simpler than generating $\mathrm{O}(n)$ due to the former's connected nature. Moreover, in either case, there is no effect for dropout.

[11] These 'neurons' do have a stochastic drift in their definition due to the random walk. In general, individual 'neurons' are an ambiguous concept in an isotropic network.

This creates a link between the stimulus's presentation time to the network and the 'neurons' it alters, such that stimuli presented in a smaller time window perturb a similar subset of the network's neurons. This may produce an encoding qualitatively similar to that found in human cognition, where neurons are thought to go through excitability cycles of slightly differing frequencies and phases. When the excitability is higher, information (engrams) preferentially encodes upon those neurons [62, 63]. As groups of 'neurons' begin to decohere, some overlap remains, such that memories are interlaced if they occur within a temporal window of coherence. *This potentially gives neural networks using isotropic dropout an advantage in time-series data*. However, this is not suggested as a model of such neurological processes, only a similar behaviour in deep learning, which is made possible through isotropic choices. Similar correlations could be considered for anisotropic dropout, likely to have a similar resultant effect.

D Potential Applications

Besides the proposed general applicability of the isotropic modifications, the following are some areas where they may yield significant performance benefits or enable desirable network behaviours.

D.1 Isotropy In Transformers

It is argued that isotropic deep learning may be a more appropriate inductive bias for deep learning. However, there may also be some architectures which are particularly enhanced by its inclusion. One of these is the self-attention step of transformers [10], where isotropic-$\tanh$ may be of particular benefit, in replacing the softmax operation [64].

Softmax is defined through elements being bounded between zero and one, $\mathbf{f}(\vec{x}) \cdot \hat{e}_i \in [0, 1]$ and summing to one. Consequently, it is non-negative, and there are regimes where this may be a limiting factor. It forms a $n-1$-simplex embedded in the $\mathbb{R}^n$ space, normal to $\vec{1}$, therefore a degree-of-freedom is also lost: $\mathbf{f} : \mathbb{R}^n \to \triangle^{n-1} \hookrightarrow \mathbb{R}^n$.

It has been shown that representations can exist in an antipodal superposition [16], particularly when stimuli do not tend to coexist in samples; thus, antipodal arrangements can exist with minimal interference. Such a stimulus may be a continuous quantity, but its two extremes are mutually exclusive. Many of these semantics are present in the real world, such as daytime-to-nighttime, motion towards or away, and smiling versus frowning. These could be represented through a zero-to-one scale; however, a $[-1, 1]$ scale may be a better representation, with zero as a better neutral middle point. This is because, in the linear features hypothesis, the magnitude often indicates the strength of the stimulus's presence. In this case, the negative of a semantic direction may be equally meaningful and present in varying amounts. It may be expected that enabling this behaviour within the self-attention step is favourable.

Moreover, the sum-to-one case may not always be desirable: it always encourages a change to the semantics when considering the residual-step-modification. **This may encourage a semantic correction to an activation in transformers, even when it is inappropriate, or optimisation may force the existence of a near-zero value vector to prevent corrections.** The residual step only encourages an identity transform independent of the activation, in the linear layer, rather than the identity dependent on a particular activation.

The self-attention step compares the pairwise similarities between several vectors grouped into the so-called 'keys' and 'queries'. The degree of similarity then affects how much of another semantic is expressed: the 'values'. However, the softmax layer is basis-dependent and prevents a negative expression of these value semantics.

A more suitable choice may be isotropic-$\tanh$. In analogy of its sum-to-one constraint, its vector-magnitude is at maximum one, $0 \leq \|\mathbf{f}(\vec{x})\| \leq 1$, whilst elementwise its values are $-1 \leq \mathbf{f}(\vec{x}) \cdot \hat{e}_i \leq 1$. Hence, it can express a negative of the value semantic, or any scaling of it between -1 and 1. This suggests that isotropic-$\tanh$ may be an appealing drop-in replacement for softmax in the attention step, at least conceptually. Its continuous rotational symmetry may also offer an advantage, since the underlying pairwise similarity of self-attention $QK^T = \vec{x}^T W_Q^T W_K \vec{x} \hat{=} \vec{x}^T W'_{kq} \vec{x}$ is also basis-independent in $\vec{x}$ for isotropic initialisations of W, which somewhat aligns with the principles of isotropic deep learning. Hence, removing further bases may enable a more even interpolation between, and perturbation to, the value vectors. Hence, an isotropic adaptation to a self-attention may appear as shown in *Eqn.* 68, which will be explored in future work.

$$\text{Attention}(Q, K, V) = \text{Isotropic-Tanh}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{68}$$

However, this does not make transformers 'isotropic' as a whole, since there are many further anisotropic steps present. Nevertheless, single-layer isotropic adaptations remain compatible with a larger anisotropic network or radial-basis network. Therefore, one may hybridise these approaches if appropriate. In general, isotropy may not be applicable to current transformers, due to their likely selection upon anisotropic primitives. Hence, a ground-up approach may be generally required, whilst borrowing concepts from the established transformers.

D.2 Real-Time Dynamical Network Topology

An appealing feature of isotropic deep learning is the relation displayed in *Eqn.* 14, showing that due to rotational equivariance, a rotation to one weight matrix can be counteracted with the inverse-rotation of another, preserving the network's function. Consequently, a gauge freedom is formed due to functional forms commuting with symmetries. With gauge freedom, a particular gauge that expresses the weights beneficially without affecting network functionality can be chosen.

One such gauge expresses the parameters in a basis with a magnitude ordering of the singular values for the matrix rows/columns. One could then set a threshold for the singular value, and bias, to determine if each corresponding direction in such a matrix has a meaningful contribution to the overall functionality. If it is deemed to have negligible value, it can be pruned with little adverse effect on the network.

Moreover, ζ latent neurons can be included, with zero-initialised singular values fully connected to existing neurons. These do not impact performance, but enlarge the activation space and parameters available. Since the Jacobians of the isotropic activation functions are not strictly diagonal, these latent neurons may be rapidly trained if required. Therefore, the otherwise static, fully connected network is now dynamic, growing and shrinking in response to task-necessitated demand, with minimal impact on performance.

This is enabled through an isotropic functional form. It poses an interesting research direction, enabled by the continuous rotational symmetry available. Transfer learning and task-swapping may become more straightforward. The network may grow to accommodate new tasks, or prune to optimise the model and stabilise computation. For example, it may stabilise by removing near-negligible, but sometimes significantly non-zero effects, which may be maladaptive due to their infrequent

effect. Output and input neurons could also be appended and removed in such a way, allowing for real-time changes to a dataset, or even training on multiple datasets. Such a procedure could be extended to convolutional networks, allowing a dynamic number of kernels. Similar possibilities may exist for other architectures.

It appears that it may sidestep the Lottery Ticket Hypothesis [34] in choosing the optimal network size before training. Due to the computational cost, this does not need to be computed at every step; instead, it can be performed periodically and layerwise.

This could offer substantial insight into how parameters may be shared between tasks in real-time. For example, the author postulates that if a new dataset is introduced partway through training on a different dataset, there might be a short-term increase in parameters until the network's parameter-sharing begins, followed by a pruning phase until a more compact architecture is reached. Questions such as these could motivate the development of a subfield offering insight into these research avenues. These network dynamics may be incredibly insightful. Overall, this enables task-driven, real-time neural plasticity in deep learning. Other continuous symmetry-primitives may enable similar approaches.

D.3 Multi-Headed Layers

Although not limited to isotropic deep learning, producing 'multi-head' feed-forward layers may be desirable, enabling perturbative-like corrections to activations at each layer. This could be achieved by summing over a series of activation functions in each layer. An example of this is shown in *Eqn.* 69 or a more general construction in *Eqn.* 70, for weight matrices $\mathbf{W}^j$ and $\mathbf{W}'^j$, biases $\vec{b}^j$ and $\vec{b}'^j$ and an activation function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$.

$$\vec{x}^{l+1} = \sum_j \mathbf{f}\left(\mathbf{W}^j \vec{x}^l + \vec{b}^j\right) \tag{69}$$

$$\vec{x}^{l+1} = \sum_j \mathbf{W}'^j \mathbf{f}\left(\mathbf{W}^j \vec{x}^l + \vec{b}^j\right) + \vec{b}'^j \tag{70}$$

This is effectively a sum over several feed-forward layers, which could increase the expressibility of a layer. It may be especially beneficial for isotropic functions, such as isotropic-$\tanh$, where each sum produces a further perturbative 'correction' to an output vector. A scaling could be enforced if an ordering of such perturbative corrections was desired.

Furthermore, one could consider an alternative, but similar, structure based upon a fibre-bundle symmetry extension over these various heads. Each layer could constitute a base space, and the multiple heads could be considered a fibre, or vice versa. This can enable a definitional form for matrix-valued functions. Additionally, in such a case, one may also consider the heads as a vector representation over each base neuron, and restructure the parameter maps accordingly for computation between various vector-valued-like neurons. This generalisation may find applicability in certain specialisms, whilst the general structure above may have broader applicability.

D.4 Isotropic Representations May Aid Semantic Alignment

An emerging interdisciplinary field of semantic alignment [65] are trying to produce comparable representations between deep learning and the brain. The author believes it is worthwhile to investigate how the representations generated by the Isotropic Deep Learning approach can aid in achieving this objective. This is because anisotropic functional forms have been shown to create representational structure due to functional forms; this structure is not a naturally emerging consequence of the data [17]. These artificial structures may be detrimental to representational alignment objectives. Removing anisotropy may help with alignment methods that use continuous rotation-like, such as in the work of Williams et al. [66], since the inductive bias of isotropy is equivariance to continuous rotation. However, this connection is largely speculative, but is included as a point of discussion and potential research avenue for isotropic deep learning. For isotropy, this may provide a testable route for the hypothesis that isotropic deep learning forms more 'natural' semantic structure in representations.

However, this is not to suggest that the brain is likely to be isotropic, especially since the approach produces delocalised functional forms. In isotropic networks, neurons instead act as a collective and are arbitrarily decomposable into any set of individual neurons, due to the gauge invariance. Consequently, there is likely no identifiable and agreeable definition of a neuron in an isotropic network. This is fundamentally incompatible with the brain's structures. On the other hand, observations of distortion in deep learning due to anisotropy do not imply that the brain also produces anisotropic and discrete features when time-averaging its neuron firings. Therefore, despite the incompatibility of functional forms with biological neuron behaviour, their representations may have substantial similarity, and it may be possible to produce better alignment between their respective activation distributions once anisotropic-incurred structures are removed from deep learning.

A similar approach may be extendable to inferring meaning from languages where a large corpus is available. If isotropy produces a continuous representation, free from basis distortions, then one may expect a more structured and interpolatable semantic structure. One may speculate whether an approximately language-agnostic structure may develop in a similar analogy to representational alignment — the latter field is arguably assuming (and often encouraging) an approximately model-agnostic representation structure. Hence, there may be a chance that representations free from artificial anisotropies may aid in deducing an alignment between known and unknown vocabulary. This may be an interdisciplinary application of Isotropic Deep Learning. Smaller-scale empirical approaches could begin with a hybrid of the two methods to establish whether embedded activation representations can be aligned between other simpler biological systems' neural activity or vocalisations, and similar domained deep learning models. This may give insight into the approximate semanticity of some of these. Similar can be tried for various models within deep learning, determining if convergence onto one, or several, universal-like representations occurs within a similar domain.

Despite this, the success of ensemble models may limit this alignment. Ensemble models use diverse individual models to collectively approximate a task solution. The model diversity would suggest that there are more minima than those connected through a permutation or continuous symmetry, which would not yield diverse individual solutions, as they are functionally identical. Therefore, there are likely functionally diverse models with substantially different internal representations. This may challenge assumptions of a universal and comparable representational structure for semantics. Despite this, they may all produce different approximations to a universal representation. Substantial testing will elucidate this, and the basis-undistorted representations produced by isotropic networks may be particularly beneficial in such an endeavour.

E Comparisons With Geometric Deep Learning Approaches

E.1 Distinction from Equivariant Networks

Geometric Deep Learning and this Primitive-First approach both consider deep learning through the use of symmetries and group-theoretic lenses. Naturally, due to this, several implementation, formulaic and terminological convergences occur, and such interdisciplinary considerations and understanding of their consequences may be practically advantageous in general. For example, similarities include the (algebraic) invariant and equivariant relation in their definition, which extends to other group-theoretic usage, representation theory, Lie Groups, Gauges, etc., which can make them appear similar due to a incommon formalism. Yet, these perspectives on symmetry in deep learning remain distinct in their intended purpose, the resultant consequences for applications, their broadly different regimes of consideration, their independent motivation for such symmetry principles, and their differing potential impacts and implications for the field. Several places in this work already briefly outline overlaps and dichotomies, particularly regarding taxonomic unification. This section aims to provide an extended outline of the parallels and divergences between these philosophies, establishing a holistic picture of these complementary considerations of symmetries in deep learning.

Primarily, the guiding philosophy of Geometric Deep Learning is to ascertain which symmetries (specifically the *algebraic* symmetries in the taxonomic terminology) are inherent to a given dataset and in the intended application. Using this, one then constructs models which are guaranteed to respect this data-derived structure. A clear example of this approach is the broad array of equivariant networks that have been developed, and several of these in particular will be discussed further below to compare similarities and differences. These have required the construction of individual maps to further this model-scale equivariance when composed together. This does create some overlap, specifically within algebraic considerations, but it is primarily a top-down approach — it starts with model-scale constraints and recursively applies them downwards to ensure it is retained over all the functions and their compositions.

For primitive-first reformulations, it starts with determining the implications of primitive functions as foundational biases and their compositional biases, which are expected to be hierarchical in nature. This is its initial line of enquiry before leveraging such findings generally. This is not just pertinent to activation functions but extends over primitives more generally: optimisers, normalisers, operations, initialisations and many more foundational maps. These can be collated into sets defined by particular symmetries through a taxonomised system of three generations and three flavours to indicate the strength/degree and type of symmetry categorising each map — this merges into the broader taxonomic approach. In this, Geometric Deep Learning occupies the algebraic sector in furtherance of model-scale symmetries. This is not the sole constraint nor scale-remit of the primitive-first approach, for which the whole taxonomy is relevant at all scales. This approach is the suggestion that these primitive characterising symmetries may have important implications internally for networks. Their form may have direct and indirect implications for the internal dynamics of networks, through representations and learning dynamics, which, once suitably understood, can be leveraged in more general applications. It is hypothesised that such symmetries may be important and generalising analytical quality of many primitives for each set[12]. Hence, the development and investigative audit of all such primitives and their consequences, followed by the rebuilding of general architectures for applications, is the primary guiding principle for this work. Hence, it constitutes a bottom-up approach which is less predicated on specific data-driven structures.

To extend this brief comparative analysis, this section will continue with a discussion and outline of the several key approaches within Geometric Deep Learning, namely Equivariant Group-Convolutions [23] and discussion of Steerable-CNNs [24], Harmonic networks [25] and Spherical-CNNs [26]. Following this is a summary of the similarities between these methods and the primitive-first approach — namely, one can produce some alignments when considering only algebraic symmetries of the taxonomy. Finally, crucial differences will be highlighted to demonstrate the distinctiveness of these approaches in general, particularly in terms of symmetry in deep learning. Overall, this indicates distinct but potentially complementary group-theoretic approaches.

An outline of Cohen and Welling [23]'s Group Equivariant Networks is to utilise a specific symmetry, particularly the one expressed in the underlying task's data-structure domain, and ensure the network as a whole respects the task-relevant symmetry through use of a modified convolution operation: Group Convolutional Neural Networks (G-CNNs). This is generalising the traditional translation equivariance of the convolution operation (ignoring edge effects) to instead be equivariant to a general discrete group $\mathcal{G}$. This symmetry group is chosen a priori by considering the given dataset, so the approach is to leverage *the known symmetries of the task as a strong and constraining inductive bias* to ensure accurate desired solutions. Consequently, this symmetry-respecting constraint is applied end-to-end over the *whole* network architecture. This can have a wealth of benefits, including increased weight-sharing efficiency, physically accurate modelling, and a resultant increased expressive capacity.

A summary of this design procedure is: identify if the specific task has its data distributed over a particular linear base space and if it is expected that applications upon this data are expected to follow a symmetry of that space. If so, then the data can be 'lifted' onto a symmetry group acting on that space, and an associated equivariant model can then be used to achieve the intended application. The connection to symmetry can be denoted for data and general activations by the following map: $f : \mathcal{G} \to \mathbb{R}^n$ — every element within the group is assigned an n-dimensional vector by f, such that the resultant vectors are interrelated through the intended group action. For Cohen and Welling [23], their convolution operation then preserves group-structure in its convolution map for discrete symmetries, producing new features that are still structured over the group. These can then be stacked to form a model which respects the end-to-end symmetry.

[12] Although it is recognised that other analytical qualities specific to individual implementations also contribute, so to some extent could be considered an 'effective theory'.

The group-convolution is implemented as a modification of the classic discrete convolution operation: by applying the filter over the group, as shown in *Eqn.* 71. In *Eqn.* 71, k indexes the filter ψ and notably the sum is over the base space $\mathcal{X}$ in the first layer: $h \in \mathcal{X}$, not $h \in \mathcal{G}$. Then, the resultant equivariant group convolution respects the symmetries of the task at every layer, given by the aforementioned data structure $f : \mathcal{G} \to \mathbb{R}^n$. A naive implementation results in augmenting the number of filters to accommodate every action of the group; however, crucially, these are shown to be related through permutation, so can be achieved more efficiently in practice by an indexing that exploits the group's structure. For more details of precise implementation, see Cohen and Welling [23].

$$[f \star \psi](g) = \sum_{h \in \mathcal{G}} \sum_{k} f_k(h)\, \psi_k\left(g^{-1}h\right) \tag{71}$$

Overall, this adapts the convolutional operation and padding to respect the symmetries in the underlying data structure. It is also shown that several existing primitives commute with these group actions. Particularly, the existing elementwise non-linearities commute with the considered group actions, and hence must be retained in their current functional form to ensure end-to-end adherence to the symmetry. In other cases, only small modifications to other primitives, such as those specified for normalisations, need to be made, which still enables them to retain their current functional form. Hence, in this approach, primitive reformulations are potentially detrimental to breaking the end-to-end equivariance of the construction — in this case, retaining the current elementwise form is important for commuting with the group action. In short, the current primitives are retained.

In the subsequent works of Cohen and Welling [24], Worrall et al. [25], and Cohen et al. [26], considerable progress is made in developing models capable of a greater range of equivariance symmetries through extending the architectures and tools. In Cohen and Welling [24], the authors build upon earlier work by creating steerable capsules, in which vectors transform under irreducible representations of the discrete group. These steerable filters are constructed as linear combinations of base filters, resulting in a more parameter-efficient design. In [26], they generalise these concepts for images over a spherical shell, $\mathcal{S}^2$, and lift it to an $\mathrm{SO}(3)$ continuous symmetry equivariance, using a Fourier transform-like method. Through these and others, networks are made *algebraically* equivariant to discrete group transformations and extended to specific continuous group transformations. This subfield is highly active and rich with many other successful discoveries along the same vein of data-driven symmetry approaches for the model scale. The key differences can be clarified with the stated examples.

Returning to Worrall et al. [25], the authors use the steerable filters to construct equivariance to continuous patch rotation using finite filters constrained onto circular harmonic functions exhibiting the desirable rotational equivariance. This develops into complex-valued activations and maps, where the initial input data becomes the real part in the complexification. To maintain the rotational equivariance in the harmonic network, an activation function is introduced which must act upon the complex-valued activations but is also constrained to ensure rotational equivariance. The result is an activation function which acts on the absolute value of the complex number elementwise. This instance of an activation function which applies over the absolute value, can be considered to constitute a single activation function of the algebraic $S_n \times \mathrm{U}(1)$ branch, and if expressed multivariately could be manipulated into the functional form such as *Eqn* 29. This indicates that application-driven instances of primitives do occur and could be retrospectively classified through the taxonomic considerations of this paper, and particularly strictly as algebraic generations. However, these also emerged in the context of a primitive that must further the broader model's constraints, so consideration of their foundational biases is not discussed in relation to its wider impact on general networks outside of model-scale equivariance, which contributes further divergences in philosophy.

Building upon the Harmonic network, Thomas et al. [67] utilise a more generalised instance intended for geometric tensors which enables equivariance to local rotation, translation and permutation, extending the prior work of Harmonic networks for various geometric tensors. As a necessity, the non-linearity must act on the tensors in a manner which is not changed under the specified group actions, for rank-1 tensors this is manifestly a form of norm-based activation function instance to ensure equivariance, drawing some parallels with instances of isotropic activation functions but only for tensor objects, meanwhile for rank-0 (scalars) tensors this reduces to the standard elementwise activation function of the typical anisotropic forms and higher rank tensors are similarly demonstrated. These choices ensure that the non-linearity does not break the equivariance to these actions.

Overall, the shared language of group theory emerged naturally in both approaches as a result of their respective objectives. One uses it in ensuring model-scale adherence to a specified algebraic symmetry group lifted from the data, whilst the other considers sets of primitives with respective function-driven foundational and compositional biases. These do sometimes constitute overlapping considerations and potential for shared tooling, mostly limited to instances of activation functions for specific algebraic symmetries. Yet, they remain tangent in both their motivating purpose, particular versus general applications and the consequences of symmetry in a network. Although the primitive reformulation of deep learning is of geometrical and deep learning construction, it does not appear to sit cleanly into the current field of geometrical deep learning [27]. Instead, it is constructed around the geometry of embedded representations, altering the internal symmetries of general networks rather than a network-wide externally applied symmetry instilled by a predominantly task-driven inductive bias. In many ways, the primitive first approach could be considered the consequences and leveraging of symmetry *breaking* in many circumstances, where a network is not enforced to be end-to-end symmetric and instead these actions can be thought to act on the network and influence its behaviour in unintentional and peculiar ways. These are then conjectured to reemerge as a plethora of scattered phenomena discussed. A further discussion of the differences is provided below.

There are several more distinct differences in the approach, particularly concerning where symmetries arise, the distinct inductive biases considerations, and how these may influence vector spaces. These are further detailed below.

One clear difference of the primitive-first approach is that it is enforced in the internal representation spaces and not necessarily preserved through the transformations between the chain of vector spaces within a network. *In effect, not only is symmetry breaking allowed and represents a crucial aspect of study*. This is a difference between the more externally imposed and the internal consequences of symmetry, the symmetries which trivially commute with a network and those which may non-trivially interact with it. As a result, the symmetries themselves and the effect on representations and optimisation may substantially differ.

For the primitive-first approach, a set of primitives can be chosen network-wide and defined by a particular relation, which would be typically constructed from a *family* of symmetry groups, e.g., orthogonal for isotropy. These primitive sets may also be defined through closure or statistical symmetries *not just the algebraic variety*. In this approach, the particular layers then acquire a specific instance of this family (and importantly, a certain representation of that instance) to be constrained under. In practice, a general principle of O family symmetry enforcement may be chosen for the primitive functional class, which every element of this class then has an equivariance to a specific symmetry from this family. For example: A layer's activations form an $\mathbb{R}^l$ linear vector space, which is transformed through a function $\mathbf{f} : \mathbb{R}^l \to \mathbb{R}^l$. As a consequence, a representation of $\mathrm{O}(l)$ could be used to define $\mathbf{f}$'s functional form, but perhaps $\mathrm{O}(l')$ for another layer's primitives. Therefore, per layer, a primitive of a specific symmetry from a specified symmetry family is used, whose functional forms are in/equivariant to its actions. In contrast, an equivariant network is made equivariant to a specific instance of a symmetry class, which is task-necessitated. Hence, generally, isotropy would be concerned with a symmetry family rather than a particular instance of a symmetry group. In isotropic deep learning, the network as a whole does not need to respect a specific symmetry, but may only do so if desirable. The primitive-first approach is, in many regards, the investigation of the symmetries and symmetry-breaking both compositionally and via primitives individually, including interacting and breaking the symmetries of the underlying graph's connectivities.

There is also a substantial distinction in the nature of bias being manipulated/leveraged. In particular, the addition and removal/control of inductive biases stemming from data or functions, respectively. For equivariant models, this is the addition of highly-specific task-aligned inductive bias, informed by the application and hard-coded into the architecture for equivariant networks to increase efficiency, leverage symmetry structure for generalisation, and constrain the solution to a known hypothesis space. It's the highly targeted addition of strong data-driven inductive biases. Whilst primitive reformulations, and in particular isotropy, are generally the removal, control and understanding of a usually unintended inductive bias across general task categories, e.g. reconstruction, classification, next-time step prediction, etc. For example, the removal of basis-dependent anisotropies of the current formalism. Other primitive forms would have their own representational inductive biases.

Thus, isotropy is motivated as a minimal inductive bias, removing anisotropic biases that may induce undesirable function-driven representation structures, and hence pitches the reformulation as a new default for broader applicability. The consideration of function-driven biases has potentially near-universal design benefits to consider. The aim is primarily understanding, then reducing or leveraging, unintended inductive biases emerging from primitives, potentially allowing for a less arbitrary and maybe more natural and well-suited representational geometry — particularly free from function-driven undesirable structure. These function-driven consequences generally form task-agnostic inductive bias [17]. However, in principle, one could still use a priori task-specific knowledge for a problem where a strong constraint should be added through primitives. This highlights their substantially differing considerations, often the addition of data-driven, highly targeted inductive biases compared to the general understanding, reduction or control of inductive biases emerging from primitives across many tasks.

Moreover, this can have very different consequences for the vector spaces themselves. The approach to equivariant networks modifies the construction of maps between, and may constrain the resultant dimensionality of, vector spaces to enforce a global symmetry. Whereas, new primitives such as the isotropy family would largely leave the vector space construction unchanged, affecting only certain transforms between them — more of a like-for-like replacement in many general network scenarios. Even without such alterations made directly to the vector space to accommodate the group structure, equivariant networks must preserve pose information throughout the network, resulting in representational degrees of freedom that explicitly indicate solely this pose, which then transforms appropriately under actions. This can reduce some freedom in the representation semantics and construction. In contrast, these representational and construction constraints are not innate to primitive reformulations — representations can adapt to represent all manner of semantics without being constrained by construction to preserve pose. Hence, typically, primitive reformulations would leave the vector space construction unchanged, besides potentially differing interpretations. A specific symmetry is not enforced globally, but instead a family of symmetries is applied only locally per representation space. Therefore, in many scenarios, it can act more as a drop-in replacement whilst still reshaping representational geometry. However, other methods of leveraging these biases may also encourage specific constructions. Principally, isotropy is just elevating the existing discrete inductive bias in functional forms to a continuous one. It leaves the architecture topology and, consequently, vector space construction unchanged, giving it broader applicability if reformulations are deemed leveragable. Similar considerations also apply to the other symmetry forms of primitives.

Several of these arise from the fundamental difference in how symmetry is considered to emerge in primitive-reformulations, and these are from the underlying graph's connectivity. Primitives are grouped by the symmetries they respect, and one could consider the choice of primitive constrained by the actions which the underlying graph inherently supports. Choosing from those which are broken by the graph may have significant distinguishing consequences. Hence, in many regards, the choice of symmetry could be considered to derive from the graph's connectivity as opposed to data-driven. This foundational approach to deriving the applicable symmetries, used in the selection of primitives, will be outlined below and is being substantially developed for future publication. It also indicates that, from a primitive-reformulation perspective, both the taxonomised sets

of primitives and the underlying directed graph could be considered foundational.

The overall intention is to produce an axiomatic-like approach to the selection of primitives defined by groups available by the graph, and this can be used to generalise deep learning principles over differing maps whilst retaining the characterising construction of deep learning models as generalised neuron-like objects with interconnectivity. This then makes explicit the axiomatic-like choices made when selecting functional classes with associated symmetries and hence implications. From this picture, the rotational symmetry defining isotropy can be argued to emerge from the architecture itself by leaving the connectivities invariant, alongside a selection of many other *continuous* symmetries which achieve the same invariance. In this work, isotropic deep learning is predominantly discussed in terms of fully connected feed-forward architectures, of an arbitrary number of hidden layers and an arbitrary number of neurons per hidden layer. However, in future work, it will be explained that these symmetries can both arise and be broken when considering arbitrary graph structures, resulting in products of various groups.

This can be achieved by endowing an arbitrary directed graph with nodes that acquire continuous activations, and one can then examine the actions which leave the most general computational functional class, contingent upon this construction, closed and invariant. Using this, one can also produce a formal layer definition to separate vector spaces systematically and indicate how successive maps influence representations. Here, the continuous rotational symmetry defining isotropy, among others, can be selected from or further broken down into primitives defined by a subgroup, such as discrete permutation symmetries characterising modern deep learning. Overall, this graph-theoretic construction indicates which symmetries are fundamentally available by the given graph, and then can be chosen to strengthen these into closures/statistical/algebraic relations, yielding more constrained functional forms. This can be applied to functional forms across the board, including activation functions, initialisations, normalisations, regularisers, optimisers, etc. Although it is strongly acknowledged that architecture's performance may be predicated upon anisotropic primitives, which results in a selection, and therefore, there may exist circular implications and potentially new architectural considerations for specific primitives/tasks — these may be investigated under general graph-primitive interactions. However, overall, this construction is fundamentally graph-driven, as opposed to data-driven, in the emergence of symmetry. Hence, the primitive reformulations can be thought to be derivable from graph-theoretic argumentation, a distinct and independent source that motivates the group-theoretic approach.

Overall, this is a substantial difference from the task-dependent symmetry informing the development of a specific architecture, as in equivariant networks. This divides the approaches into two distinct design axes, particularly regarding the origin of their symmetry, the beneficial leverage they offer, and their relation to architecture. Some overlap may occur, but primitive reformulations could be considered predominantly the consequences of function-driven symmetries arising from architectural constraints on functional forms. In contrast, equivariant networks are a symmetry of the underlying data structure that influences the architecture.

This is not to say that one is better or more principled, or that they are even particularly parallel techniques; they are constructed for differing purposes, they are differing design axes, different implications for the field and differing lines of enquiry; however, they do remain unifiable under the broader taxonomy. In some regards, they represent differing philosophical approaches to the notion of symmetry in deep learning, one focusing on the model scale recursively downwards and the other on the implications of primitives upwards. Geometric deep learning is predominantly for respecting a specific symmetry in solutions present in a particular task, such as in many physics-related problems, while the other controls function-driven implications on general networks. Specifically for isotropy, this is the removal of a basis dependence that may arbitrarily and anisotropically affect the distribution of representations in all problems, motivating its universal adoption in deep learning. Additionally, isotropy is a proposal of basis independence and a resulting gauge invariance, which has separate motivations including those in *App.* D.2. Hence, these two differing proposals: an algebraic task-driven external-geometric symmetry framework and a general internal-analytical symmetry framework, both of which utilise group-theoretic notation as a core feature but differ otherwise. It is argued that primitive reformulations provide a framework with potentially broader and more flexible applicability, even when the data doesn't exhibit known symmetries, justifying the assertion that it may have the potential to offer a better default inductive bias one day or may be leveraged on a task-by-task basis. Yet, it does not supersede the case-by-case application of a strong inductive bias in equivariant networks, which yields state-of-the-art performance in specific tasks. As it stands, the substantial differences in approach to symmetry render several equivariant networks incompatible with primitive reformulations, as evident in several models' restriction to pointwise nonlinearities. However, this is not a fundamental incompatibility between these two symmetry approaches, as seen in other models [25, 67]. As the understanding of the emergent implications of primitives grows, more implementations may accommodate and utilise this paper's approach as an additional design axis for bias control if desirable.

Concluding: the primitive-first and the various generalised models under geometric deep learning share a similarity in their fundamental constructions from group theoretic approaches and a philosophical focus on symmetry principles. Equivariant networks are designed to enforce an end-to-end symmetry derived from their data structure, which is enforced by architectural implementations such as group convolution [23]. This dramatically increases parameter efficiency and ensures physical solutions for specific problems. Isotropic deep learning promotes the existing discrete permutation symmetry to a continuous one in localised functional forms, whilst not necessarily making the network equivariant as a whole. The premise is to unconstrain embedded representations for general problems by primarily removing arbitrary basis dependence in functional forms. This is hypothesised to enable networks to form better-structured latent spaces. Similar considerations and implications exist for the other symmetry-defined primitive branches, and it is argued to be beneficial in investigating these consequences, particularly through the taxonomic proposal. In summary, there exist considerable differences between equivariant networks, which are instilled with an a priori respect for a task-dependent symmetry. In contrast, isotropy is being developed as a less

restrictive inductive bias as a universal default for functional forms.

Hence, Isotropic Deep Learning is both a framework of geometry and deep learning; however, it currently represents a significant deviation from the foundational blueprint within Geometric Deep Learning [27], despite the use of a group-theoretic approach. They have substantial differences in how and which symmetries arise, their implications on models and the broader field, how activations are represented in networks and interpreted, and additionally how parameters are constructed and training dynamics. Isotropic Deep Learning more suitably falls within the class of geometries of representations, more closely related to the work of Elhage et al. [16], Olah et al. [31], Carter et al. [68]. This interdisciplinary approach, alongside work such as Elhage et al. [16], may be more appropriately classified under a name such as '*representational geometry*' in the context of deep learning. This is then broadened to the proposed '*taxonomic deep learning*' system, which may enable clearer comparisons between group-theoretic approaches, facilitating shared tooling while also encouraging consideration of all scales and their compositions, thereby unifying these disparate approaches.

E.2 A Further Comparison on Activation Function

The approach to Geometric Deep Learning has considered smaller compositions down to non-linear considerations in the furtherance of model-scale equivariances. In this section, one such approach will be explained and then discussed for comparison. The study of this will delineate the tangent considerations underlying the two distinct approaches to symmetry in deep learning. This will begin with an abridged overview of Finzi et al. [69]'s generalised approach to equivariant multilayer perceptrons, followed by an analysis between the differing symmetry considerations.

The subtleties and depth of the following method are far more extensive and better articulated in the author's outstanding original paper; the discussion which follows is intended to be a high-level, approachable overview for a general audience. I highly recommend consulting the original exposition for a fuller understanding and appreciation of their advancements.

As a consequence, several *reductions* in generality will be made for simplicity, as these are sufficient for comparison in tangent usage of symmetry being considered. For example, we will consider only rank-1 tensors (vector inputs), which this method generalises far past — these generalisations make the method especially effective, and necessary for its applications, but are not needed for this comparison in approaches. Furthermore, the orthogonal group will be referred to draw parallels with isotropy; yet, both methods generalise over other groups in their respective considerations.

At a high-;evel, the motivation for Finzi et al. [69] is constructing multilayer perceptron blocks, f_{EMLP}, which exhibit an algebraic equivariance to *general* groups: $\forall g \in \mathcal{G}$ then $g \circ f_{\text{EMLP}} = f_{\text{EMLP}} \circ g$. The intention would be to compose these upto a model-level construction, potentially with other blocks, but in such a way to ensure the resultant model produces an end-to-end equivariance following from the underlying data-distribution. At a high-level, this motivation is depicted in *Eqn.* 73, for an example model $f_{\text{model}} = f_{\text{EMLP n}} \circ \cdots \circ f_{\text{EMLP 2}} \circ f_{\text{EMLP 1}}$. One then desires end-to-end equivariance such as $\forall g \in \mathcal{G}$: $g \circ f_{\text{model}} = f_{\text{model}} \circ g$.

$$
\begin{aligned}
f_{\text{model}} \circ g &= f_{\text{EMLP }n} \circ \cdots \circ f_{\text{EMLP 2}} \circ f_{\text{EMLP 1}} \circ g \\
&= f_{\text{EMLP }n} \circ \cdots \circ f_{\text{EMLP 2}} \circ g \circ f_{\text{EMLP 1}} \\
&\vdots \\
&= g \circ f_{\text{EMLP }n} \circ \cdots \circ f_{\text{EMLP 2}} \circ f_{\text{EMLP 1}} \qquad (72) \\
&= g \circ f_{\text{model}} \qquad (73)
\end{aligned}
$$

This aligns with the general methodology of geometric deep learning for producing networks which are invariant or equivariant to a symmetry derived from the underlying data structure.

For this consideration, we will consider the case of a d-dimensional vector (rank-1) tensor input — again, the author generalises this to general rank tensor inputs. Yet one can consider having a data distribution in which m features rank-1 vectors are desired to be mapped to n features, rank-1 vectors, where the rank-1 tensors transform as expected.

One can then set up a linear map: $f_{\text{linear}} : \mathbb{R}^{d\times m} \to \mathbb{R}^{d\times n}$, with $Z_{di} = f_{\text{linear}}\left(X_{cj}\right)_{di}$. To ensure equivariance, this requires the following constraint resulting from *Eqn.* 73: $f_{\text{linear}}\left(\mathbf{R}_{cb}X_{bj}\right)_{di} = \mathbf{R}_{de}f_{\text{linear}}\left(X_{cj}\right)_{ei}$, denoted in summation convention. This limits the possibilities for the linear map which f_{linear} can take, constraints which Finzi et al. [69] provide an implementation to compute efficiently.

In the simplest case, one could consider $f_{\text{linear}} : \mathbb{R}^{d\times 1} \to \mathbb{R}^{d\times 1}$. The layer's parameters are then constrained by a system of linear equations, which can be solved using the methods therein. This results from the above constraint for $g \circ f_{\text{EMLP}} = f_{\text{EMLP}} \circ g$, where if one considers the multilayer perceptron as $\mathbf{W}\vec{x} + \vec{b}$, and $\mathbf{G}$ the matrix representation of g, then $\mathbf{WG}\vec{x} + \vec{b} = \mathbf{GW}\vec{x} + \mathbf{G}\vec{b}$ for all $\vec{x}$. This would yield a constraint $\mathbf{WG} = \mathbf{GW}$ for all matrix representations $\mathbf{G}$ of actions of $\mathcal{G}$ — similarly for bias $\vec{b} = \mathbf{G}\vec{b}$, which will not be considered moving forth. This then expresses the possible weight matrices as a parameterised sum over effectively decomposed basis directions, $\hat{\mathbf{W}}_i$, for the weights: $\mathbf{W} = \sum_i \beta_i \hat{\mathbf{W}}_i$. For example, for an end-to-end $\mathrm{O}(n)$ equivariance symmetry, the weights are restricted to $\mathbf{W} = \beta \mathrm{I}_n$, whilst an end-to-end S_n equivariance would yield $\mathbf{W} = \beta_1 \mathrm{I}_n + \beta_2 \mathbf{1}$, where $\mathbf{1} = \vec{1}\vec{1}^T$ is the matrix of all elements equal to one. Re-generalising this recovers numerous successful architectures, after the following step.

So far, this remains a linear map, and the objective is an equivariant generalisation of a multilayer perceptron, requiring the consideration of an activation function. Practically, this necessitates a differing representation where aspects of the rank-1 vector which transform trivially (invariant to the action) are separated from those which do vary under its action. These produce a set of rank-0 scalars, s_j, which are by definition invariant to the group actions. The pointwise activation function is then

applied to these: $\sigma(s_j)$ and merged with the rank-1 tensor 'part' under a gated non-linearity: Gated $(\vec{v}_a) = \vec{v}_a \sigma(s_a)$. This collectively, then commutes with the group-action and provides the equivariant multilayer perceptron layer, which generalises over many groups. Though in the form discussed, this does not retain the universal approximation over general groups. Hence, the authors consider its generalisation by inserting a bilinear layer contracting various objects before the linear map mixing, which addresses the prior approximation problem.

Nevertheless, even without such generalisations, the parallels and differences between considerations of symmetry can be made clear. To begin, one can draw parallels. Considering solely activation functions from the primitive reformulations of this work, defined under equivariance to a group family, then consider a map such as $f_{\text{linear}} : \mathbb{R}^{d\times 1} \to \mathbb{R}^{d\times 1}$, then in some sense, the latter can be reinterpreted as some form of parameterised activation function. Particularly, if you enable it to be composed generally with any other primitive/map and allow its equivariance to a general symmetry family. Each's mutual definition ensures this; these must align, admittedly in this slightly contrived context, due to both's shared algebraic equivariance for all the actions of the chosen group. However, this is a rather far extrapolation from its original intent, and this rather restricted context is only intended to demonstrate a connection between such tooling.

Despite this, it is acknowledged that the original framing diverges far from this. This construction is intended for use in end-to-end network for a data-driven, and it was not derived to be applied as an instance of a parameterised activation function for composition with other general non-equivariant layers. This is a far reframing of the original premise considered in the paper, and is only intended to establish some parallels with the present paper. Additionally, in practice, this construction may not be as trivial as the drop-in replacement provided for orthogonal isotropic functions proposed in *Sec.* 3.1.

More interestingly, both approaches would apply differently and remain relevant in practice. This better demarcates the differing approach philosophies to symmetry in deep learning. As stated, the Geometric Deep Learning blueprint is to consider the underlying data structure present in the chosen domain, lifting this to a symmetry to which a model can be designed about to feature and end-to-end equivariance or an invariance to this group action. In such cases, the model is structured such that a symmetry passes through the network or the network is invariant to it. In some cases, the symmetry does not have the opportunity to 'act' on the network itself and induce biases. In some regards, this paper is a systematic investigation, moreover leveraging the inductive biases which result from symmetry *breaking* taxonomised by its various forms and compositions, and differing primitives. This, as described in *Sec.* 5, and is, in its many forms, hypothesised to be critical to the expressibility of deep learning. This symmetry breaking *still* occurs in these geometric deep learning models, just not with respect to the end-to-end construction enabled by architectural and parameter sharing adaptations. This best demonstrates how the two approaches to symmetry, regarding end-to-end networks and the primitive-first approach, are intended to break and change the primitive-form's symmetry to control inductive biases for beneficial computation in a computationally cheap alteration with no guarantee of end-to-end symmetry. Hence, they are tangent in both purpose and their implications. These act as differing design axes, and even both considerations may pertain to geometric deep learning models, due to the broad and general reach of the primitive-first approach being proposed.

The multilayer perceptron of Finzi et al. [69] continues to feature such consideration under the linear map composed with pointwise activation functions, which mix the scalars that transform trivially under the action. The influence of such a choice can be considered generally, which demarcates the considerations of this paper in the consequences resulting from primitive forms. For example, the activation function applied in this context, over the trivially transformed scalars, continues to act as an S_n-derived application of a primitive. This is hypothesised to also exert an inductive bias on the system, its optimisation trajectories, representations, phenomena and resultant learning, which may require analysis. Whether such reformulations can act in unison and interplay with geometric deep learning models remains to be considered; such proofs tend to rely on pointwise non-linearities as in the paper and may require rederivation.

Overall, this clearly demonstrates the differing considerations of the usage of symmetries within deep learning. The effects of the primitive-first approach begin with defining sets of primitives connected by their construction under various groups, and extending from this base upwards under compositions reaching the model level. Meanwhile, the geometric deep learning approach operates in the reverse direction, desiring model-scale in/equivariance and considering the various composing blocks required to achieve this — a complementary top-down approach. These tangent approaches result in group-theoretic considerations independently, but overlap can be established as shown due to this common mathematical root. Such a diverse perspective on group-theoretic principles in deep learning may enable broader insights, analyses of interactions, shared tooling and deeper understanding. Such complementary approaches are considered under the formalism of *Sec.* 5.2, which is intended to further such connections through the proposed taxonomy.

F Anisotropy as a Historical Precedent?

This section examines the historical precedent for anisotropic functions, as well as how external influences may have shaped the field in this manner. In particular, a discussion is provided on how such functional form choices became nearly axiomatic to the definition of deep learning. It requires a look into how coding schemes in neuroscience may have imparted this inductive bias from deep learning's[13] conception as a distinct discipline. However, there have been *many* influences throughout the field's development, and this section is only intended as a retrospective and speculative analysis of how neuroscientific coding schemes may have produced and reinforced the anisotropic deep learning to become the dominant paradigm by an author who is not a historian. Therefore, this is only speculation on how elementwise functional forms became ubiquitous.

Neural networks, in all their forms, represent their information through a neural code, which remains under debate [71, 72, 73]. Several codings have been proposed, and it appears that initially anisotropic functions may have indirectly arisen from one of these: local coding.

Local coding predates deep learning [74] and was relatively more common around the advent of artificial neural networks. It has now been largely superseded by distributed codes, often sparse [21], as a model of biological neuronal activity [75] and, to some extent, in deep learning [31, 76, 16, 17]. Local coding is a neuropsychological hypothesis that one neuron's activation represents the presence of one real-world stimulus. This is also commonly known as the grandmother neuron interpretation [77, 78], and less so as gnostic neurons and gnostic fields [79]. This neuroscientific hypothesis was developed [74, 80, 79] and debated [81] from the early 1940s through to the early millennium [77, 71, 78, 75], with a growing consensus that the brain frequently uses sparse coding in the present day. However, it will be argued that this somewhat outdated local coding scheme has left a lasting impact on the field of deep learning.

Crucially, McCulloch and Pitts co-authored a paper supporting the presence of several differing pattern feature detectors in frogs' retinas [82], expanding on Hartline's earlier work of similar findings [83]. Thus, it was shown that retinal ganglions can respond to distinct real-world patterns, so-called 'bug detectors' for the frog. Despite the study of a ganglion, this work aligns closely with the description of grandmother neurons. These are the same McCulloch and Pitts who were earlier credited as the inventors of the binary threshold network [84], which laid the foundation for the first perceptron neural network to be developed [70]. In the very first sentence of McCulloch and Pitts [84] work, it states:

> *Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic.*
>
> *–A Logical Calculus of the Ideas Immanent in Nervous Activity [84], p. 115.*

This quote could be considered to premise artificial neural networks on a framework of treating neuronal activity and interacting representations as a binary logic. This is arguably an earlier perspective of the same vein of work that led to their later paper [82], indicating a growing interest in local coding at the time of deep learning's initial development and particularly forming a link between McCulloch and Pitt's foundational work in deep learning and their neuroscientific work. This is in addition to the computational convenience of the binary approach for early computers. A further indication of a pervasive local coding paradigm in one of McCulloch and Pitts [84]'s concluding statements:

> *[...] pushed to ultimate psychic units or "psychons," for a psychon can be no less than the activity of a single neuron. Since that activity is inherently propositional, all psychic events have an intentional, or "semiotic," character. The "all-or-none" law of these activities, and the conformity of their relations to those of the logic of propositions, insure that the relations of psychons are those of the two-valued logic of propositions. Thus in psychology, introspective, behavioristic or physiological, the fundamental relations are those of two-valued logic.*
>
> *–A Logical Calculus of the Ideas Immanent in Nervous Activity [84], p. 131.*

This "psychon", with semiotic qualities and two-valued logic, is suggestive of a local coding approach. This aligns strongly with the paradigm of binary operations on components of decomposed vectors in modern neural networks, by generalising the two-value logic. McCulloch and Pitts [84] may have normalised the brain-inspired, basis-dependent nature of computation very early in deep learning's foundations based upon these neuroscientific paradigms of the time. This likely influenced Rosenblatt, who later expanded upon McCulloch and Pitts [84] directly, including the implementation of the Heaviside step function into his perceptron models [70].

These notions may have implicitly encouraged the adoption of the Heaviside step function in early deep learning through the foundational work of Rosenblatt [70]. This approach to activation functions has a continuous lineage to the vast array of elementwise functions utilised today. Connected through a series of incremental modifications from discrete Heaviside step functions, to the adoption of differentiable sigmoid-based functions [85], to non-saturating ReLU [15] through to its variants. This, in turn, may have had some influence on the broader array of functions, such as elementwise Dropout [2] or approximating Hessians in a basis-dependent manner in adaptive optimisers [47, 48, 7] and many other instances. This is argued to have become more entrenched in the field over time as future developments continued to replicate the elementwise pattern and build upon it.

Deep learning may have diverged from neuroscience's later developments because of a fundamental difference between the fields. The brain has been incrementally shaped by natural selection; it is a natural structure, and modern interpretations of any specific coding do not produce a corresponding shift in its structure. Whereas for deep learning, these are artificial

[13] While the term 'Deep learning' did not have usage in the early stages of the field's development, it is adopted in this overview for clarity and continuity. This extends to early models such as [70].

networks, shaped by choices. Thus, if coding interpretations have shaped the field, they may permanently induce such biases into network characteristics. Local coding may have influenced early researchers to use an anisotropic form, the discretising effects of which then bias representations to retain local coding [17]. Observations of this can then be used to implicitly justify the elementwise form, thereby preventing further scepticism in choosing this approach. This may have contributed to cementing this choice as foundational to the field, resulting in the underappreciation of the existence of such a choice. Moreover, the anisotropy may have circularly self-reinforced through performance metrics: anisotropic forms can produce symmetry broken representations [17], which is speculated to have further benefit when using anisotropic functional forms which operate along such predisposed discretised representations.

Hence, the effect may be two-fold: when analysing deep learning models, they frequently display local coding, implicitly suggesting that the corresponding elementwise functional forms are most appropriate and thus not drawing attention to it. Furthermore, the discretised representations may, in turn, benefit the further continuation of these forms, since they may now outperform alternatives due to the preexisting effect on representations. Thus, the approach is speculated to have become reinforced through feedback loops. This may have also had some influence on the direction of processors developed for such tasks. All of this may contribute to why the choice has become a staple and generally unappreciated as a *choice*.

These decisions then produce a tendency towards local coding [17], summarised in a sentiment expressed by Olah et al. [31].

> Szegedy et al. found that random directions seem just as meaningful as the directions of the basis vectors. More recently Bau, Zhou et al. found the directions of the basis vectors to be interpretable more often than random directions. *Our experience is broadly consistent with both results; we find that random directions often seem interpretable, but at a lower rate than basis directions.*
>
> *–Feature Visualization, Distill blog publication. [31]*

This summarises the long series of conflicting observations within neural networks, some suggesting monosemanticity [86], others distributed [87], and some a mixture [88]. It is not suggested that deep learning fails to produce such distributed representations, which are often observed throughout its development [70, 16]. However, its functional forms are argued to have an inductive bias that predisposes representations towards basis-aligned local coding. Hence, this wide variety of results can be reframed as a strong bias by functional forms, but partially overridden by more complex arrangements under strong task-dependent conditions. This frequently observed break from local coding may also indicate that alternative representation distributions are optimal, but limited in observations to certain situations due to these functional form biases.

The author contends that this bias has been unintentionally but systematically obfuscated through many years of mathematical notation suppressing explicit basis-dependence, leading to sentiments such as the following:

> *This internal structure has appeared in situations where the networks are not constrained to decompose problems in any interpretable way. The emergence of interpretable structure suggests that deep networks may be learning disentangled representations spontaneously*
>
> *–Network Dissection: Quantifying Interpretability of Deep Visual Representations [32], p. 1.*

By virtue of their elementwise functional forms, artificial neural networks are loosely constrained to decompose problems in an interpretable manner [17]. These inductive biases are injecting structure, despite the impression that no such constraints are applied. It is felt that this may be an explicit reference to a sentiment generally felt as a field, rather than one held solely by Bau et al.

This paper as a whole is intended to clarify this underappreciated and implicit inductive bias. Hence, to prevent any self-reinforcement from such an implicit bias, one would require Isotropic deep learning to be a one-step overhaul or a meticulous analysis of all functional forms, since any hybridised approaches may systematically reintroduce biases in representations and affect results.

Overall, this section has provided a brief overview of how anisotropic deep learning can be traced back to a trajectory set in the early developments in the field, shaped by the biological discoveries of their time, and has become pervasive through modern developments driven by research momentum and hypothesised self-reinforcement. The foundations of the field were defined by the choice of the elementwise Heaviside step function, which was implemented due to both computational necessity and likely biological inspiration. It has evolved into a generalised, elementwise form used nearly universally today through a continuity of modifications. The cross-disciplinary approach between neuropsychology and artificial neural network development may have initially inspired this functional form; however, it is argued that they have likely diverged due to the different systems studied. Yet, this has imparted an implicit inductive bias, entrenched into the mathematical structures of deep learning, which now lacks a suitable a priori justification — as neuroscience has made local coding outdated as a *universal* code. The following subsection discusses how isotropic networks may better interact with distributed codes.

F.1 Isotropy as Semantic Modulation for Distributed Codes

Non-linear transformations play foundational roles in deep learning, critically through the universal approximation theorems which demonstrate the *dense* network's ability to perform complex manipulations to model data effectively. Beyond their role in universal approximation, activation functions may also be conceptually interpreted as modulators of semantic content, selectively remapping and suppressing features that correspond to various notions. This discussion aims to provide a qualitative and speculative perspective on this role of activation functions.

Local coding, as discussed in *App.* F, is characterised by a one-to-one correspondence between a neuron and a semantic. Therefore, each neuron's activation is associated with the prevalence of a distinct semantic concept. In contrast, distributed codes have semantic meaning that is dispersed across a population of neurons. Consequently, semantics no longer tend to align with individual neurons. There are conveniences to local coding; its simplicity makes networks very interpretable, a benefit for both AI safety and diagnosing network pathologies. It is therefore often given as an intuitive first-order approximation to the action of deep learning models. However, this first-order heuristic may inadvertently suggest that such inductive biases are benign — a position this paper challenges.

Modulation of distinct semantics is a desirable action. Operations such as bounding the strength, rectifying the signal, and distorting aspects of its stimulus to neuron-response curves may all be beneficial semantic actions for a network, altering its internal expressions and enabling better interactions between concepts. These actions can all be achieved using the application of an activation function: Tanh and Sigmoid, ReLU [15], or Leaky-ReLU [15], Swish [13] and SiLU [89], respectively. Alongside comparative and logical actions between semantics, such as those achieved by Softmax or gate mechanisms [56], respectively. If each neuron encodes a single semantic feature, then under this assumption, it is appropriate to apply such operations elementwise, independently acting on each neuron to scale their semantic representation. Under a local coding assumption, elementwise operations suffice in modulating and enabling interactions between entire semantic features, since each neuron corresponds to an independent semantic.

An additional influence may have been the expert system approach, where fixed if-then logic produces a condition on each meaningful quantity to produce the desired output. This methodology may also have influenced the on-off activation function approach, such as the Heaviside step function. This approach aligns conceptually with local coding.

A consensus is emerging that whilst networks often tend to this local coding [31], the representations are often more nuanced in practice [31, 88, 16]. It is argued that the network may balance local coding with interference and representational capacity needs [16]. This paper and Bird [17]'s work have further explored the issue by implicating functional forms as responsible for the formation of this coding tendency.

Under these more generalised distributed codes, single activation actions and comparisons are insufficient for interacting with whole semantics. Such operations may only distort the fraction of the semantic, represented through a single neuron. This underpins the neural refractive problem. Instead, a distributed logic is required. Instead of assigning operations upon numbers as components of arrays, it is suggested to consider a full vector space treatment (inner-product space), with magnitude and directions as foundational quantities rather than components. It is argued that operations should act on these, more fundamental and basis-free quantities.

This is reinforced by the changing affine transforms — due to parameters adapting during training or random symmetry-breaking initialisations. Therefore, vector directions may be unpredictably distributed and change rapidly during training, causing activations to move around the representation space. As activations evolve around the representation space through training, so too do the semantics they individually and collectively represent. Therefore, activation functions as semantic moderators should not be chosen to only act upon single neurons, as semantics are not wholly expressed by individual neurons. Especially, since the tendency towards symmetry-broken local coding only emerges after considerable training [17]. Moreover, this is compounded by lasting polysemanticity in specific neurons [31, 16] even after training.

Since semantics are often found to be distributed in the representation space, and adapt through learning, without a predictive theory of their trajectories, a sensible inductive bias is to apply the modulation isotropically. Hence, ensuring that any linear feature, including those off-axis, are modulated consistently by isotropic forms. This respects both distributed coding schemes and polysemantic neurons. Furthermore, the application of functions isotropically still correctly modulates locally coded semantics since their functionality can be identical along the standard basis. Thus, applying non-linearities exclusively along the standard basis is inconsistent with the goal of manipulating internal semantic meaning in distributed codes. Removing such an inductive bias is expected to remove this local coding bias, enabling more distributed representations with increased representation capacity whilst balancing concept interferences.

In conclusion, the intuitive local coding approach may have had some encouragement on the elementwise functional form used in contemporary deep learning; however, relaxing this inductive prior to a distributed neural code suggests that isotropic activation functions may be considerably more appropriate. Had distributed coding been prevalent during the early developments of deep learning, then isotropic functional forms may have emerged as the default paradigm: modulating vector magnitudes instead of standard components decomposed on an arbitrary basis. This could have been seen as a more biologically inspired computational model. While exact Isotropic forms are likely biologically implausible, due to constraints of individual neurons with localised responses, such limitations do not constrain artificial systems. Hence, Isotropy may be considered a better inductive bias to approximate coding intuitions.