
Towards Isotropic Deep Learning

A New Default Inductive Bias

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 This position paper explores the geometric implications of current functional
2 forms in deep learning and proposes a new paradigm to be termed *Isotropic Deep*
3 *Learning*. The existing inductive bias for functional forms is a discrete rotational
4 symmetry — an often underappreciated *choice*. In this paper, this symmetry
5 is promoted to a continuous rotationally symmetric framework which affects
6 *almost all* functional forms. It has been demonstrated that the functional forms
7 of current deep learning influence the activation distributions. Through training,
8 the discrete symmetries in current functional forms can induce similar broken
9 symmetries in embedded representations [1]. Thus producing a geometric artefact
10 in representations which is not task-necessitated, and solely due to human-imposed
11 choices of functional forms. There appears to be no strong a priori justification
12 for why such a representation or functional form is universally desirable, and
13 this paper proposes several detrimental effects of this current formulation. As a
14 result, this modified framework for functional forms is explored with the goal of
15 unconstraining representations and improve network performance. This framework
16 is encouraged to be developed substantially, such that it can be adopted as a new
17 default in time. A variety of preliminary functions are proposed within the paper,
18 including several activation functions. Since this overhauls almost all functional
19 forms characterising modern deep learning, it is suggested that this shift may
20 constitute a novel category of deep learning.

21 1 Introduction

22 Elementwise functional forms singularly dominate current deep learning [2, 3, 4, 5, 6, 7, 8, 9, 10, 11].
23 This is particularly evident in activation functions, sometimes referred to as ‘ridged’ [12] activation
24 functions. Activation functions are often displayed univariately [13, 14, 15], generally characterised
25 in form by *Eqn. 1*, with σ being a placeholder activation function, e.g. $\text{ReLU}(f(x) = \max(0, x))$,
26 $\text{Tanh}(f(x) = \tanh(x))$, etc.

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

27 However, this display choice obfuscates a crucial (standard) basis dependence. This is explic-
28 itly displayed in *Eqn. 2* as a multivariate functional form, which should be considered a more
29 implementation-correct form¹. This reveals the functional form’s usually hidden \hat{e}_i basis dependence.
30 The multivariate form is depicted for an n neuron layer, with activation vector $\vec{x} \in \mathbb{R}^n$. This standard
31 basis dependence is arbitrary and appears as a historical precedent, rather than appropriate inductive

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

32 bias, discussed further in App. G.

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

33 Due to this basis dependence, non-linear transformations differ angularly in effect [16, 1]. Therefore,
 34 this will be termed an *anisotropic function*, indicating this rotational asymmetry. Due to the pervasive
 35 use of these functional forms, including optimisers, normalisers, regularisers, activation functions
 36 etc., current deep learning as a paradigm may consequently be termed ‘*anisotropic deep learning*’.
 37 Despite its implications, this *choice* of basis-dependent anisotropic form appears underappreciated
 38 and incidental in the development of most contemporary models, with anisotropic forms are almost
 39 treated as axiomatic to deep learning rather than a considered choice. Hence, re-evaluating and sys-
 40 tematically reformulating this foundational aspect of modern deep learning, with such wide-reaching
 41 consequences, is felt to constitute a new branch: ‘*Isotropic Deep Learning*’.

42 This asymmetry in the non-linear transform is particularly about the standard (Kronecker) basis
 43 vectors, and their negative, $\{+\hat{e}_i, -\hat{e}_i\}_{\forall i}$, due to the elementwise application. Therefore, it can
 44 be said to distinguish the standard basis - a ‘*distinguished basis*’². This is an often unappreciated
 45 implicit inductive bias in the representational geometry. For example, the standard basis’ activation
 46 space distortions are visible in Fig. 1 showing the mapping of elementwise-tanh on a variety of
 test shapes. Most generally, this *choice* of functional forms can be considered to break *continuous*

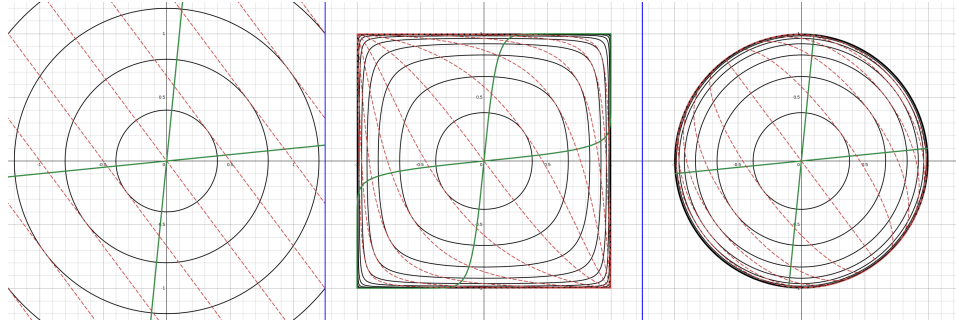


Figure 1: Left shows a 2-dimensional plane, \mathbb{R}^2 , populated with various shapes: black concentric circles, green lines through the origin, red parallel lines and in faint black the standard (cartesian) coordinate axes \hat{e}_1 and \hat{e}_2 (which are kept untransformed). If this space is then 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. Right shows a similar 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. An interactive demonstration of these functions is available [Removed for Anonymity].

47 rotational symmetry, and reduce it to a *discrete* rotational symmetry — a permutation symmetry
 48 of the standard basis. In effect, if the function is treated in its multivariate form, in deep learnings
 49 current formulation, it is equivariant to a permutation of the components of its vector decomposed
 50 in the standard basis. For an element of the permutation group $\mathbf{P} \in \mathcal{S}_n$, the following equivariance
 51 relation holds $f(\mathbf{P}\vec{x}) = \mathbf{P}f(\vec{x})$. However, permutation symmetry is a discrete rotational symmetry
 52 which is a subgroup of the special-orthogonal symmetry proposed: $\mathcal{S}_n \subset \text{SO}(n)$ — the permutation
 53 symmetry can be said to be a broken continuous rotational symmetry.

55 Non-linearities are usually pivotal to the network’s ability to achieve a desired computation, as seen
 56 through the universal approximation theorem’s [17] explicit dependence on the form of the activation
 57 function [18]. The non-linearities produce differing local transformations, such as stretching, com-
 58 pressing, and generally reshaping a manifold - displayed elegantly in Olah [19] (whom also stated

²This is suggested generalisation from a ‘*privileged basis*’ discussed in Elhage et al. [16]. ‘*Distinguished basis*’ is felt to better reflect how representations may be more-or-less aligned to a basis; whereas ‘*privileged basis*’ is felt to suggest a greater alignment. The term ‘basis’ will be retained despite the set of ‘*distinguished vectors*’ potentially being under-/over complete for spanning the activation space, as demonstrated by Bird [1].

59 disillusionment on the elementwise form for different reasons). Consequently, the network may
60 be expected to adapt by moving representations to geometries about these distinguished directions,
61 to use specific local transforms to achieve the desired computation. Hence, an anisotropy about a
62 distinguished basis is induced into the activation distribution shown by.

63 This has been empirically demonstrated by Bird [1]: training results in the discrete symmetry of
64 the functional forms, inducing a broken symmetry in the activations. Since these non-linear zones
65 are centred around the distinguished bases, the embedded representations are expected to move
66 to beneficial angular arrangements about the arbitrarily imposed privileged basis's geometry. For
67 example, they appear to move towards the non-linearities' extremums, aligned, anti-aligned or other
68 geometries, through training [1]. This may correspond to a local, dense or sparse coding [20] or
69 superposition [16], respectively.

70 Therefore, the network has adapted its representations through training due to these functional form
71 choices, probably for various optimisation reasons. The causal hypothesis aids in explaining the
72 observed tendency of privileged-basis alignment. This is the hypothesis underlying the author's posi-
73 tion: **functional forms should be a deliberate and considered decision, with a suitably optimal,**
74 **and minimally harmful, default.** Currently, anisotropy results in a human-caused representational
75 collapse onto the privileged basis, and it is suggested that this is frequently not a task-necessitated
76 collapse. There appears to be little justification for why this is universally desirable, with several key
77 negative implications discussed in *Sec. 2*. Without a priori justification, this inductive bias may be
78 detrimental to computation, so unconstraining the activation appears generally preferable.

79 Overall, historic and frequently overlooked functional forms for modern deep learning directly influ-
80 ence the models' activations and therefore behaviour. A functional form has been entrenched, which
81 is basis-dependent on an arbitrary basis, typically obfuscated, neglected, and seldom questioned [21].
82 Yet, a causal link between this arbitrary basis and activations has been empirically demonstrated [1].
83 Hence, a resultant effect on the final performance of the model is hypothesised and should be explored.
84 These are often underappreciated choices due to decades of largely unquestioned usage, which appear
85 to have significant consequences for representations. Therefore, this should be well-justified as a
86 default and studied. This is the position of the author.

87 Throughout the rest of this position paper, **it is argued that a departure from this anisotropic**
88 **functional form paradigm towards the isotropic paradigm may be generally preferable as an**
89 **inductive bias**, unless otherwise justified. It encourages the reader to be conscious of these choices
90 when designing a model, as well as the usual architectural tool kit. Particularly, isotropic choices,
91 equivalent to basis independence, may be thought to unconstrain the representations into more optimal
92 arrangements for general tasks and architectures. There are some instances where isotropy may be
93 particularly beneficial, such as the amendments discussed for self-attention, speculated in *App. D.1*.
94 At least the tenets of this paradigm of reviewing such functional form choices are encouraged,
95 and comparisons at least should reveal which characteristics of functional forms most significantly
96 contribute to performance.

97 2 Hypothesised Problems of Anisotropy

98 This section argues that current functional forms impose unintended anisotropic performance detri-
99 ments, indicating that *Isotropic Deep Learning*, once substantially developed, are proposed as the
100 default inductive bias unless an alternative is task-necessitated.

101 This section lays out a non-exhaustive set of arguments discussing some implications that anisotropic
102 functional forms may cause. These mainly centre on the role of the activation functions, since this is
103 the area the author has primarily explored in their PhD thus far. To the author's knowledge, some
104 of these failure modes are newly characterised phenomena, such as the so-called '*neural refractive*
105 *problem*'.

106 2.1 The Neural Refractive Problem

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

110 It appears to be a phenomenon in all anisotropic activation functions to date. The ‘refraction effect’
 111 typically occurs more significantly at larger magnitudes — potentially a failure mode under network
 112 extrapolation. Neural refraction is demonstrated by curvature of previously straight origin-intersecting
 113 lines in the centre plot of Fig. 1, but not in the rightmost plot of the same figure.

114 Mathematically, this has several representations, a magnitude-varying ‘dynamic refraction’ shown
 115 in Eqn. 3 or differentially in Eqn. 4. Also defined is a ‘static refraction’ definition shown in Eqn. 5.
 116 These are described for a multivariate activation function \mathbf{f} and vector $\vec{x} = \alpha \hat{x}$ where \hat{x} is a unit
 117 vector. This relation may be satisfied for a single direction, a subset of the space or all directions
 118 $\hat{x} \in \mathcal{X} \subseteq \mathcal{S}^n$. The relations generally show how the activation function alters the direction of its
 119 input vector in an anisotropic manner.

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

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

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

122 It can be seen that along a straight-line trajectory in direction \hat{x} , the result of the activation function is
 123 a curved line if dynamically refracted. Therefore, if the linear feature hypothesis is followed, then
 124 *every* linear feature, in refracted directions, becomes curved following the activation function. The
 125 network may exploit some of this curvature to construct new linear features in the subsequent layers;
 126 however, there may be many instances where this curvature is detrimental to established semantics.
 127 The network may lose semantic separability, produce magnitude-based semantic inconsistency or
 128 compensatory maladaptations in later layers, which fail for out-of-distribution samples. This may
 129 hinder the generalisation performance of the network, which is resolved by isotropic choices.

130 Particularly detrimental, in both refraction cases, can be the loss of semantic separability. If two
 131 distinct trajectories, representing different semantics, are transformed into curves which intersect or
 132 converge, then the separability of these concepts is lost or misrepresented. For example, suppose one
 133 direction is a linear feature for the presence of a dog in an image, whilst the other is for a horse. In
 134 that case, if these activations are of a magnitude where the activation function causes convergence,
 135 the identity of the activation’s meaning can be misconstrued.

136 This may be particularly consequential for functions such as Sigmoid and Tanh, since large magnitude
 137 inputs end up at particular limit points (discussed as trivial representational alignments in Bird [1]).
 138 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
 139 an i , s.t. $\hat{x} \cdot \hat{e}_i = 0$, then the transformed vector has a 0 in the corresponding index. Therefore, all
 140 vectors end up at limit points with sufficient magnitude when using elementwise-Tanh or -Sigmoid.
 141 A fully-connected layer can only effectively separate two such converging directions at a time, which
 142 are then further curved by a subsequent activation function.

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

143 Consequently, semantic separability is lost for large magnitudes except for 3^n discrete limit points for
 144 Tanh and Sigmoid. Therefore, embedded activations may be expected to align with these limit points.
 145 This explains some results empirically observed by Bird [1]. Similarly, ReLU has one distinct limit
 146 point, $\vec{0}$, but otherwise an orthant unaffected by neural refraction. The author speculates whether
 147 this is an additional reason for the success of ReLU, due to only a subset of directions experiencing
 148 the neural refraction phenomena. Furthermore, this would suggest an advantage of Leaky-ReLU:
 149 despite featuring static-refraction, directions do not become overlapped, so semantic separability is
 150 retained. Otherwise, the network may expend training time on producing robust semantic separability,
 151 a needless compensatory adaptation, which may lower representational capacity or extend training as
 152 a result.

153 More generally, dynamic deflection of trajectories may cause semantic ambiguity for the network,
 154 where only samples interpolable from training samples are reliably semantically identifiable. Par-
 155 ticularly, *the more significant the deflection, the greater the semantic ambiguity may be expected.*

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.

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

This may explain why the network may perform excessively poorly 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, as this would be a considerable magnitude of the linear feature, which is typically deflected the most 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 to avoid the damaging consequences of neural refractions.

Angular anisotropies fundamentally cause the refraction phenomenon. If compression and rarefaction of certain angular regions occur, linear features will be deflected in various ways. A fix for this is introducing isotropy — the initial motivation for developing the paradigm. This 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 are only an issue when they affect linear, not affine, features in a potentially unpredictable and thus semantically uninterpretable way.

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

$$\mathbf{f}(\alpha\hat{x}) = \|\mathbf{f}(\alpha\hat{x})\| \hat{x}' \quad (8)$$

$$\mathbf{f}(\alpha\hat{x}) = \sigma(\alpha) \hat{x}' \quad (9)$$

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

$$\mathbf{f}(\alpha\hat{x}) = \sigma(\alpha) \hat{x} \quad (10)$$

In standard notation, Eqn. 10 can be rewritten into the *final functional form for isotropic activation functions* shown in Eqn. 11. This is later compared to other functional forms in Tab. 3.1.

$$\mathbf{f}(\vec{x}) = \sigma(\|\vec{x}\|) \hat{x} \quad (11)$$

This can be generalised as a result of rotational equivariance of the function, which is used to generalise the principle under a symmetry. This can be expressed as a condition in Eqn. 12, which uses a commutator bracket for convenience, with $\forall \mathbf{R} \in \text{SO}(n)$. 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. This 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.

$$[\mathbf{R}, \mathbf{f}] = (\mathbf{R}\mathbf{f} - \mathbf{f}\mathbf{R}) = \vec{0} \quad (12)$$

The relation may be more familiar as $\mathbf{f}(\mathbf{R}\vec{x}) = \mathbf{R}\mathbf{f}(\vec{x})$. This relation only applies to single-argument functions and requires generalising to more circumstances. A preliminary condition may be $\mathbf{f}(\mathbf{R}\vec{a}_1, \dots, \mathbf{R}\vec{a}_N) = \mathbf{R}\mathbf{f}(\vec{a}_1, \dots, \vec{a}_N)$ for $\mathbf{f} : \bigotimes_N \mathbb{R}^n \rightarrow \mathbb{R}^n$.

This introduces the general isotropic functional form for activation functions given in Eqn. 13. This should be a piecewise function, defined using the identity at $\vec{x} = \vec{0}$, but this is suppressed for simplicity. This functional form is $\mathcal{O}(n)$ time for \mathbb{R}^n . Future work is establishing a universal approximation theorem for this functional form, as this is ongoing research for the author's PhD.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}) = \sigma(\|\vec{x}\|) \hat{x} \quad (13)$$

This is not to be confused with the radial-basis functional form displayed in Eqn. 14.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}) = \sum_{i=1}^N \sigma(\|\vec{x} - \vec{c}_i\|) \hat{e}_i \quad (14)$$

The ‘neural refractive problem’ outlines how semantic meanings may become intertwined or ambiguous due to current functional forms consistently skewing linear features in undesirable ways. A

hypothesis is developed that this may 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 excessively. It may be expected that the network produces compensatory adaptations for the phenomena, which might 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 wasting training time on these corrections to the activation distributions due to unintended refraction.

2.2 Emergence of Linear Features and Semantic Interpolatability

As previously mentioned, symmetry-broken functional forms induce symmetry-broken representations. Thus, *approximately* discrete embedding directions are tended towards [16, 1]. Since embedded activations are often discretised, so too may be semantically meaningful directions since these are conjectured to align with these anisotropic embeddings. This generally appears to be the case [22, 23, 1]. Reversing this causality would suggest that a continuous rotational symmetry allows a continuous embedding. Functional forms would not induce direction-based symmetry breaking in their embeddings through training. It is hoped this will enable networks to acquire a more naturally continuous, and hence optimal, representation in their embedding for the given task.

However, many real-life semantics are continuums: colours, positions of objects, broad morphology, even within a single species. Representational collapse onto a single discrete semantic may lose this vital nuance. It appears a poor inductive bias to have functional forms encourage discretised representations. Isotropic functions do not prevent discrete semantics, which can be clustered through bias terms. However, they do not promote discretisation either, enabling continuous representations since they remove arbitrary basis-dependencies and limit points. Therefore, moving towards isotropy is hypothesised to encourage embeddings to be more smoothly distributed, taking on intermediate values between typically discrete linear features and substantially enlarging the expressivity and representation capacity of networks — only limited by concept interferences.

In this case, the discrete concept of ‘representation capacity’ may become irrelevant; each layer may express different continuous arrangements, where differing concepts are angularly suppressed and expressed in analogy to the linear features hypothesis [24]. Instead, the ‘*magnitude-direction hypothesis*’ is proposed as a continuous extension, magnitudes indicating the amount of stimulus present, direction indicating the particular concept. Activations then populate this more continuous manifold.

This continuous-semanticity may also produce a better organised semantic map at each network layer since intermediate representations may now relate otherwise discrete features. This connectivity can bring them continuously into proximity (which ‘weight locking’ discussed in Sec. 2.3 may typically prevent). This may additionally aid researchers in comparing representational alignment between models and biology, discussed further in App. D.4.

Therefore, in general settings, the inductive bias of isotropy appears more appropriate as a default, due to many real-world semantics being continuous. However, anisotropy may also be a good inductive bias if universal discretisation of concepts at all scales and abstraction levels is expected. Isotropy can be thought of as adding an inductive bias that enables continuous and interpolatable semantics while retaining discrete semantics when task-necessitated — as opposed to human imposition. 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 parameter³ may suffer from being stuck in local-minima found further into loss valleys, encountered only after a sufficient amount of training. This arises only due to the anisotropic functional form’s *discrete* permutation symmetry. This discussion relies upon the aforementioned observations of discrete representations due to functional form symmetry breaking.

Qualitatively, this is because the 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 ‘understood’ semantics, once these

³Though similarly applies to a ‘locking’ of the bias to $\vec{0}$.

248 semantics have developed. Consequently, this may largely halt further progress shortly after the
249 formation of discrete semantic directions.

250 In effect, further small perturbations to the parameters may move activations from a semantically
251 aligned to a semantically dislocated state, making the activation’s meaning ambiguous. The ambiguity
252 may negatively affect its corresponding output, forfeiting performance. This would create an emergent
253 ‘false’ local minima due to the discretised semantics. This is hypothesised to be due to the discrete
254 rotational symmetry of functional forms. Consequently, creating a plethora of architectural local
255 minima in the space. Only sufficiently large perturbations to a parameter may move activations
256 between two differing semantically aligned directions.

257 It may also suggest that the optimisation barrier may be some function of the angular separation
258 of the semantically meaningful directions. Angularly closer linear features would be less likely to
259 suffer semantic dislocation if an activation is nudged towards another close semantic direction. By
260 increasing the number of geometric positions representations occupy through varying anisotropy,
261 this effect may be controlled. Extremising this would suggest that isotropy is beneficial since it is
262 expected to produce continuous and interpolable representations.

263 This semantic dislocation is an emergent consequence of breaking the continuous symmetric forms.
264 The dual of this argument is basin connectivity. Without continuous rotational symmetry, the loss
265 landscape loses connectivity of many of its minima. This is a $n!$ local-minima degeneracy in weights
266 due to the permutation symmetry of an \mathbb{R}^n layer. However, enforcing the isotropy constraints results
267 in sets of continuously connected local minima which can be smoothly transformed into one another,
268 by corresponding parameter rotations shown in Eqn. 15, a consequence of Eqn. 11.

$$\forall \mathbf{R} \in \text{SO}(n) : \underbrace{\mathbf{W}^l \mathbf{R}^\top}_{\mathbf{W}'^l} \mathbf{f} \left(\underbrace{\mathbf{R} \mathbf{W}^{l-1}}_{\mathbf{W}'^{l-1}} \vec{x} + \underbrace{\mathbf{R} \vec{b}}_{\vec{b}'} \right) = \mathbf{W}^l \mathbf{f} \left(\mathbf{W}^{l-1} \vec{x} + \vec{b} \right) \quad (15)$$

269 If this is downgraded to discrete rotational symmetry (i.e. permutation symmetry), then artificial
270 optimisation barriers may reemerge in these basins. In this case, only a sufficiently large perturbation
271 to the parameters may dislodge the network into a more optimal minima, while more minor perturba-
272 tions are insufficient. Effectively, the discrete permutation symmetry may result in overlaid sets of
273 discretised lattice solutions for the parameters, much like how it breaks the symmetry of activations
274 through training too [1].

275 This is a qualitative intuition since this hypothesis remains challenging to verify until robust methods
276 to determine semantically meaningful directions are produced. Nevertheless, in either case, steps can
277 be taken immediately to counteract the problem, and this is to introduce isotropy to connect these
278 minima.

279 3 Isotropic Implementations

280 This position paper argues for implementing isotropic functional forms into neural networks as a
281 default inductive bias. Near-term adoption may be rate-limited due to the development of suitably
282 optimal functions, particularly since *anisotropic deep learning* has a substantial head start and
283 analogues to existing functions are not so trivial to produce. Despite this, several preliminary
284 implementations are outlined in this section as a starting point; however, these functions are likely far
285 from optimal and substantial research and development are required to bring isotropic deep learning
286 into practicality. *It is hoped that the arguments of this position paper will encourage the field to begin*
287 *a directed search for more suitable functions with this guiding principle of isotropy.*

288 Nevertheless, below is a non-exhaustive list of activation functions and some of their consequences.
289 A summary of other functions, including optimisers, regularisers, and normalisers, is also discussed
290 in App. B. In App. D, several suggested initial applications for Isotropy are discussed, including a
291 modification to self-attention in App. D.1 and a proposal for training-time dynamic network topologies
292 App. D.2.

293 3.1 Activation Functions

294 As stated, the isotropic functional form for activation functions is given in Eqn. 13. In Tab. 3.1, it is
295 compared with the other common functional forms. It is clear in this comparison that the isotropic

functional form is basis-independent and relatively simple. Further criteria, in addition to isotropy, are also a performance necessity. 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 paradigm 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(\ \vec{x} - \vec{c}_i\) \hat{e}_i$	$\mathbf{f}(\vec{x}) = \sum_{i=1}^n \sigma(\vec{x} \cdot \hat{e}_i) \hat{e}_i$	$\mathbf{f}(\vec{x}) = \sigma(\ \vec{x}\) \hat{x}$

301

Isotropic-Tanh is described in Eqn. 16. 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.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}) = \tanh(\|\vec{x}\|) \hat{x} \quad (16)$$

This function is reasonably cheap, computation of $r = \|\vec{x}\|$, $\tanh(r)$ and $\text{sech}^2(r)$, need only be computed once (including for backward-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: $\lim_{r \rightarrow 0} \mathbf{J}(r\hat{x}) = \mathbf{I}_n$, justifying $\mathbf{f}(\vec{0}) = \vec{0}$, to preserve a smooth gradient. It is also globally 1-Lipschitz.

Isotropic-ReLU is shown in Eqn. 17, an analogue to its traditional implementation.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}; R_0) = \max(\|\vec{x}\| - R_0, 0) \hat{x} \quad (17)$$

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. 18 and 19, 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.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}; R_0, R_\infty) = \min(\max(\|\vec{x}\| - R_0, 0), R_\infty) \hat{x} \quad (18)$$

315

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}; R_0) = \begin{cases} \vec{0} & : \|\vec{x}\| < R_0 \\ \vec{x} & : \|\vec{x}\| \geq R_0 \end{cases} \quad (19)$$

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. 20, with a small value $0 < \alpha \ll 1$.

$$\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}(\vec{x}; \alpha, R_0) = \begin{cases} \alpha \vec{x} & : \|\vec{x}\| < R_0 \\ \vec{x} - (1 - \alpha) R_0 \hat{x} & : \|\vec{x}\| \geq R_0 \end{cases} \quad (20)$$

Isotropic-Soft-ReLU uses $\alpha = 0$, and ‘Isotropic-SoftLeaky-ReLU’, $\alpha \in (0, 1)$, are left in the derivative form of the radial part $\sigma'(r)$ in Eqn. 21, where $\phi(r)$ is a monotonically increasing function. There are very many suitable candidates fulfilling this ϕ and one may be selected which 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.

$$\sigma : \mathbb{R} \rightarrow \mathbb{R}, \quad r \mapsto \sigma'(r; R_0, \delta, \alpha, \phi) = \begin{cases} \alpha & : \|\vec{x}\| \leq R_0 - \delta \\ \frac{\phi\left(\frac{r - R_0 + \delta}{2\delta}\right)}{\phi\left(\frac{r - R_0 + \delta}{2\delta}\right) + \phi\left(\frac{R_0 + \delta - r}{2\delta}\right)} & : R_0 - \delta < \|\vec{x}\| < R_0 + \delta \\ 1 & : \|\vec{x}\| \geq R_0 + \delta \end{cases} \quad (21)$$

325 **Isotropic-Sinusoids** are a possibility which do not appear to have an analogue within the current
326 anisotropic paradigm — demonstrating a broad array of new possibilities. With the aforementioned
327 isotropic-Relu-like functions, the networks may normalise magnitudes such that the distributions
328 ‘escape’ the non-linear regions of the function, due to utilising the non-linearity constructively,
329 which can initially be an unpredictable learning hurdle. Therefore, a function that introduces non-
330 linearity throughout the space may be desirable. Proposed is ‘isotropic-Sinusoids’, which allows for
331 distributions to be compressed, rarefied and folded (for $|\lambda_m| > 1$) in a predictable manner. It is hoped
332 the network can utilise this for effective computation. This activation function is demonstrated in
333 Eqn. 22. It may be helpful because it includes a monotonicity-violating parameter $\lambda_m \in \mathbb{R}$, enabling
334 folding of embeddings.

$$\mathbf{f}(\vec{x}) = \vec{x} + \lambda_m \sin(\|\vec{x}\|) \hat{x} \quad (22)$$

335 4 Alternative View Against Isotropy

336 It is argued that anisotropies result in an activation distribution shift, which may be detrimental to the
337 network’s performance. This is because this inductive bias is typically introduced universally, and if
338 no justification exists for this particular distribution, then it may be a suboptimal imposition by the
339 network designer. However, it could also be argued that some symmetry-breaking anisotropy may
340 be beneficial, particularly discretised semantics. By clustering parts of the activation distribution,
341 redundant information can be usefully lost, leading to classifications that develop more quickly.
342 In classification, one of the most common applications of deep learning, this clustering may be a
343 suitable a priori justification for anisotropy. Therefore, introducing isotropy may limit the network’s
344 performance in this case.

345 Despite this, current activation functions produce anisotropies along a Cartesian grid, due to their
346 standard basis dependence. This particular arrangement does not seem justified through classification,
347 with Pappas et al. [25] showing a phenomenon of ‘Neural Collapse’ onto an equiangular tightframe
348 for classification networks, which does not align with the standard basis. However, the works of
349 Logan and Shepp [12] suggest that decomposition onto the standard basis can aid with the curse
350 of dimensionality, though this is only indirectly associated with deep learning. In addition, Elhage
351 et al. [16], demonstrate the phenomenon of ‘*Superposition*’, which appears about the privileged basis.
352 However, using this as support for anisotropy or studies finding local coding [] would be circular in
353 logic, since it is a phenomenon of anisotropic networks and is affected by rotations to the anisotropy
354 shown by Bird [1].

355 Nevertheless, anisotropies could be reintroduced in a more general arrangement and in such a way
356 as to mitigate the aforementioned problems. This could consist of a more uniform distribution of
357 anisotropic directions from which semantics could then, more densely, develop. This is outlined in
358 App. A, which the author feels is one of the most crucial developments in this framework.

359 5 Conclusion

360 In this position paper, the isotropic functional form is proposed as a hypothesised better default
361 inductive bias for deep learning, when developed. Current forms have been demonstrated in the
362 literature to produce task-unmotivated representational artefacts [1], which this work hypothesised
363 may limit the networks’ semantic expressibility. It is further argued that the current anisotropic
364 functional forms may have detrimental effects on performance and learning, through the ‘neural
365 refractive problem’, ‘discrete semantics’ and ‘weight locking’. Hence, removing such constraints from
366 the model is also argued to unconstrain the representations from any particular basis. The network is
367 then expected to produce a more natural activation representation based upon task necessities, rather
368 than by human-imposed functional forms.

369 The adjustment to functional forms is through promoting the existing discrete rotational symmetry
370 (permutation symmetry) of modern deep learning to a continuous special-orthogonal symmetry, or
371 perhaps the orthogonal symmetry. This has substantial consequences for the form of almost every
372 function in modern-day deep learning: activation functions, regularisers, normalisers, optimisers and
373 more. It is expected to prove the semantic structure and produce high-capacity embeddings. It is
374 proposed that the breadth of this reformulation is best represented as a novel branch of deep learning:
375 *Isotropic deep learning* to distinguish it from existing paradigms.

376 This position paper showcases several examples of isotropic and quasi-isotropic functions as a starting
377 point. Yet, these are analogues of anisotropic functions and not expected to be inherently optimal
378 or better just because they display superficial similarity to existing functions. Instead, it is the
379 author's position that this change to isotropic deep learning is generally advantageous, but may need
380 substantial time for development as a paradigm, such that better optimised implementations are
381 discovered which suitably leverage the new properties available from isotropy. Therefore, empirical
382 work on these placeholder functions will be presented in the following papers to not distract from the
383 primary motivation for this shift to Isotropic deep learning.

384 The proposed ideas are hoped to stimulate the communities' interest in beginning a directed search
385 for such isotropic functions, which will hopefully bring the paradigm into widespread applicability
386 and adoption.

References

- [1] 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>.
- [2] 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.
- [3] 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>.
- [4] 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>.
- [5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL <https://arxiv.org/abs/1409.1556>.
- [6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL <https://arxiv.org/abs/1502.03167>.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL <https://arxiv.org/abs/1512.03385>.
- [8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL <https://arxiv.org/abs/1412.6980>.
- [9] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. URL <https://arxiv.org/abs/1608.06993>.
- [10] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL <https://arxiv.org/abs/1905.11946>.
- [11] 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>.
- [12] Benjamin F Logan and Larry A Shepp. Optimal reconstruction of a function from its projections. 1975.
- [13] 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.
- [14] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL <https://arxiv.org/abs/1710.05941>.
- [15] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. URL <https://arxiv.org/abs/1606.08415>.
- [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 Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

- [18] 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](https://doi.org/10.1016/0893-6080(91)90009-T). URL <https://www.sciencedirect.com/science/article/pii/089360809190009T>.
- [19] 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].
- [20] 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.
- [21] David Lowe and D Broomhead. Multivariable functional interpolation and adaptive networks. *Complex systems*, 2(3):321–355, 1988.
- [22] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization, Nov 2017. URL <https://distill.pub/2017/feature-visualization/>.
- [23] 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>.
- [24] 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.
- [25] Vardan Papayan, 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>.
- [26] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?, 2019. URL <https://arxiv.org/abs/1805.11604>.
- [27] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL <https://arxiv.org/abs/1607.06450>.
- [28] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017. URL <https://arxiv.org/abs/1607.08022>.
- [29] Yuxin Wu and Kaiming He. Group normalization, 2018. URL <https://arxiv.org/abs/1803.08494>.
- [30] Ward Cheney and David Kincaid. Linear algebra: Theory and applications. *The Australian Mathematical Society*, 110:544–550, 2009.
- [31] 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>.
- [32] Francesco Mezzadri. How to generate random matrices from the classical compact groups, 2007. URL <https://arxiv.org/abs/math-ph/0609050>.
- [33] Kai Hu and Barnabas Poczos. Rotationout as a regularization method for neural network, 2020. URL <https://openreview.net/forum?id=r1e7M6VYwH>.
- [34] 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>.
- [35] 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.

- 481 [36] Lingxuan Chen, Kirstie A Cummings, William Mau, Yosif Zaki, Zhe Dong, Sima Rabinowitz,
482 Roger L Clem, Tristan Shuman, and Denise J Cai. The role of intrinsic excitability in the evolu-
483 tion of memory: Significance in memory allocation, consolidation, and updating. *Neurobiol.*
484 *Learn. Mem.*, 173(107266):107266, September 2020.
- 485 [37] John Bridle. Training stochastic model recognition algorithms as networks can lead
486 to maximum mutual information estimation of parameters. In D. Touretzky, edi-
487 tor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann,
488 1989. URL [https://proceedings.neurips.cc/paper_files/paper/1989/file/](https://proceedings.neurips.cc/paper_files/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf)
489 [0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1989/file/0336dcbab05b9d5ad24f4333c7658a0e-Paper.pdf).
- 490 [38] Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim,
491 Bradley C. Love, Christopher J. Cueva, Erin Grant, Iris Groen, Jascha Achterberg, Joshua B.
492 Tenenbaum, Katherine M. Collins, Katherine L. Hermann, Kerem Oktar, Klaus Greff, Martin N.
493 Hebart, Nathan Cloos, Nikolaus Kriegeskorte, Nori Jacoby, Qiuyi Zhang, Raja Marjeh, Robert
494 Geirhos, Sherol Chen, Simon Kornblith, Sunayana Rane, Talia Konkle, Thomas P. O’Connell,
495 Thomas Unterthiner, Andrew K. Lampinen, Klaus-Robert Müller, Mariya Toneva, and Thomas L.
496 Griffiths. Getting aligned on representational alignment, 2024. URL [https://arxiv.org/](https://arxiv.org/abs/2310.13018)
497 [abs/2310.13018](https://arxiv.org/abs/2310.13018).
- 498 [39] Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape
499 metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and
500 J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34,
501 pages 4738–4750. Curran Associates, Inc., 2021. URL [https://proceedings.neurips.cc/](https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf)
502 [paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf).
- 503 [40] Taco S. Cohen and Max Welling. Group equivariant convolutional networks, 2016. URL
504 <https://arxiv.org/abs/1602.07576>.
- 505 [41] Taco S. Cohen and Max Welling. Steerable cnns, 2016. URL [https://arxiv.org/abs/](https://arxiv.org/abs/1612.08498)
506 [1612.08498](https://arxiv.org/abs/1612.08498).
- 507 [42] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow.
508 Harmonic networks: Deep translation and rotation equivariance, 2017. URL [https://arxiv.](https://arxiv.org/abs/1612.04642)
509 [org/abs/1612.04642](https://arxiv.org/abs/1612.04642).
- 510 [43] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical cnns, 2018. URL
511 <https://arxiv.org/abs/1801.10130>.
- 512 [44] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning:
513 Grids, groups, graphs, geodesics, and gauges, 2021. URL [https://arxiv.org/abs/2104.](https://arxiv.org/abs/2104.13478)
514 [13478](https://arxiv.org/abs/2104.13478).
- 515 [45] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas,
516 Jun 2020. URL <https://distill.pub/2019/activation-atlas/>.

517 A Quasi-Isotropic Functional Forms

518 A middleground, balancing the aforementioned problems whilst enabling representation compression
 519 not just through the bias, is to relax the hard isotropy condition and introduce slight symmetry
 520 breaking in many directions. This can be achieved using many small perturbations to the direction
 521 unit vector only, producing a softer symmetry breaking. Then the network has many distinguished
 522 vectors, a subset with which it may align its representations in a task-dependent manner. Therefore,
 523 it does not favour a particular basis, but still introduces some desirable consequences of anisotropy,
 524 for problems such as classification. If one further restricts the functions from featuring dynamic
 525 refraction, it limits detrimental anisotropic effects.

526 One method is to apply a non-linearity based on rounding the vector’s directions. This is shown in
 527 *Eqn. 23*, where $[\cdot]$ indicates the rounding operation and $\phi(\vec{x}) \neq \vec{x}$. In fact, the anisotropic perturbation
 528 may be implemented as simply as: $\phi(\vec{x}) = \beta\vec{x}$ for $\beta \neq 1$. The overall angular term is approximately
 529 unit-normed, but can be trivially modified to be exactly norm-1.

$$\Phi(\hat{x}; \alpha) = \frac{[\alpha\hat{x}]}{\alpha} + \phi\left(\hat{x} - \frac{[\alpha\hat{x}]}{\alpha}\right) \approx \hat{x} \quad (23)$$

530 This produces a quasi-isotropic functional form shown in *Eqn. 24*, with an isotropy-breaking parameter
 531 α . Slight anisotropic refraction is added, independent of magnitude, so it is predictable and thus
 532 extrapolatable to the network. Due to the angular rarefaction and compression by the proposed non-
 533 linearity, representation over- and underdensities may then occur, where semanticity may begin to be
 534 assigned. However, for $\alpha \rightarrow \infty$, isotropy is continuously reintroduced and could be an optimisable
 535 parameter.

$$\mathbf{f}(\vec{x}) = \sigma(\|\vec{x}\|) \Phi(\hat{x}) \quad (24)$$

B Beyond Isotropic Activation Functions

In this position paper, activation functions are predominantly explored, showing how the isotropic functional form opens up a wealth of new functions to explore and design. However, the 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 [1]’s results empirically demonstrating representational anisotropies due to activation functions, it is also hypothesised that other transforms also incur a similar basis-dependent representational alignment. This hypothesis is used to broaden the scope of isotropic deep learning, encouraging an overhaul to almost all functional forms within modern-day deep learning.

A non-exhaustive list of functional forms requiring reformulation are: initialisers, normalisers, regularisers, operations, optimisers and gradient clipping. This section provides a brief summary of some of these, highlighting anisotropies present in current forms. Isotropic reformulations will follow. These directions are so far incomplete, and future work will be required to both develop and add to these functional forms with empirical benchmarking.

B.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, smooth the loss landscape [26], or initially thought to primarily aid in reducing internal covariate shift [6]. In this section, their Isotropic properties are analysed. Throughout this section, it is essential to stress that it is functional form anisotropy defined over an arbitrary distribution as opposed to anisotropy in activation distribution itself.

There have been many proposed normalisation techniques, including but not limited to "Batch Normalisation" by Ioffe and Szegedy [6], "Layer Normalisation" by Ba et al. [27], "Instance Normalisation" by Ulyanov et al. [28] and "Group Normalisation" by Wu and He [29]. 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, this is not a suggestion that the functions are inherently suboptimal, as they were designed for differing purposes than Isotropic deep learning is. This is simply an analysis of whether current normalisations can be classified as Isotropic deep learning methods, and whether they introduce further representational inductive biases besides the intended distributional shifts.

Batch-normalisation [6] 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 by saturated activation functions, whilst acting as a regulariser due to noise within the statistics, and enabling a higher learning rate alongside less constraints on parameter initialisation. Expressed in multivariate form, *Eqn. 25* shows the batch-normalisation operation, with trainable parameters $\vec{\gamma}$ and $\vec{\beta}$, where $\mathbb{E}_B[\dots]$ indicates an average over the batch.

$$\mathbf{f}(\vec{x}) = \sum_{i=1}^N (\vec{\gamma} \cdot \hat{e}_i) \frac{\vec{x} \cdot \hat{e}_i - \mathbb{E}_B[\vec{x} \cdot \hat{e}_i]}{\sqrt{\epsilon + \mathbb{E}_B[(\vec{x} \cdot \hat{e}_i - \mathbb{E}_B[\vec{x} \cdot \hat{e}_i])^2]}} \hat{e}_i + \vec{\beta} \quad (25)$$

Despite the appearance of many basis terms, \hat{e}_i , under strong assumptions, Batch normalisation could be isotropic. For example, if the activation distributions were a perfect normal distribution, with $\vec{\gamma} = \alpha \vec{1}$ and $\vec{\beta} = \vec{0}$ 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, so normalising in such a way would counterintuitively incur anisotropy. This approach also makes Batch-normalisation sensitive to the mini batch size.

Layer-norm [27] consists of similar approach in normalising activations; however, the statistics are calculated in a different manner. Layer-norm acts sample-wise within a batch, so removes the dependence on batch-size, alongside with other properties, has made it a popular choice of normaliser.

583 It is given by the function in Eqn. 26, where $\mathbb{E}_i[\dots]$ is the expectation over index i occurring in the
 584 basis vectors \hat{e}_i .

$$\mathbf{f}(\vec{x}) = \sum_{i=1}^N \left(\gamma \frac{\vec{x} \cdot \hat{e}_i - \mathbb{E}_i[\vec{x} \cdot \hat{e}_i]}{\sqrt{\epsilon + \mathbb{E}_i[(\vec{x} \cdot \hat{e}_i - \mathbb{E}_i[\vec{x} \cdot \hat{e}_i])^2]}} + \beta \right) \hat{e}_i \quad (26)$$

585 Despite this function producing $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, the intrinsic dimensionality of an \mathbb{R}^n dimensional
 586 object is mapped to an $(n - 2)$ -sphere, \mathcal{S}^{n-2} , embedded within an \mathbb{R}^n space. This can be seen
 587 through $\vec{x} \cdot \vec{1} = 0$ (before addition of $\vec{\beta} = \beta \vec{1}$), therefore, the data lies in an orthogonal subspace
 588 to $\vec{1}$. Following the addition of β , the expression becomes $\vec{x} \cdot \vec{1} = n\beta$, a \mathbb{R}^{n-1} hyperplane with $\hat{1}$
 589 normal and offset from the origin by $n\beta$. By definition, this cannot be Isotropic due to a breakdown
 590 of rotational equivariance when rotating into plane's normal direction.

591 Instance normalisation [28], arose in convolutional neural networks for style transfer. It is defined in
 592 Eqn. 27, where the basis-dependence is expressed through indices for brevity, with notation aligning
 593 with Ulyanov et al. [28].

$$\mathbf{f}(\vec{x})_{tijk} = \gamma_i \frac{x_{tijk} - [\mathbb{E}_{hw}[x_{tishw}]]_{ti}}{\sqrt{\epsilon + [\mathbb{E}_{hw}[x_{tishw} - [\mathbb{E}_{hw}[x_{tishw}]]_{ti}]^2}_{ti}} + \beta_i \quad (27)$$

594 Analysing the Isotropic properties of this normalisation is made difficult due to the use of convolution.
 595 Isotropy is argued to arise from architectural symmetries of the model, discussed briefly in Apps. E
 596 and F. This approach would suggest that the isotropic equivariance relation would be restricted
 597 to a subset of rotations within a linear subspace, indexed per spatial pixel, of the entire activation
 598 space. In effect, for a given spatial location, specific height and width index, there is a resultant
 599 vector in \mathbb{R}^C , for C channels, which the rotations equivariance applies to. This would suggest that
 600 Instance normalisation, when $\beta_i = 0$ and $\gamma_i = 0$, is isotropic for convolutional neural networks.
 601 However, there may be some ambiguity on the restrictions of β_i and γ_i for isotropy, since this could
 602 be interpreted as a single linear layer with restricted parameters. However, such restrictions amount
 603 to a linear transform on the aforementioned subspace of $\mathbf{W} = \text{diag}(\vec{\gamma})$, which is a standard-basis
 604 aligned restriction on the parameters, indicating this is truly anisotropic.

605 Finally group normalisation [29], can be intuited as a middleground between Layer normalisation
 606 and Instance normalisation, since it is argued that channels are not statistically independent. Hence,
 607 the statistics are computed along these groups of channels. Since it is a middleground between both
 608 Instance normalisation and Layer normalisation, arguments for its anisotropy are trivially extended.

609 **B.1.1 Potential Isotropic Normalisers:**

610 [Section under development]

611 **B.2 Initialisers:**

612 [Section under development]

613 **B.3 Operations:**

614 [Section under development]

615 **B.4 Optimisers:**

616 [Section under development]

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.

For example, current (anisotropic-)dropout would appear to privilege the basis anti-aligned with the standard basis, to maximally preserve information when a direction of the standard basis is collapsed. So it is expected to incur an arbitrary basis dependence on 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 special-orthogonal symmetry: $\mathbf{B} \sim \text{SO}(n)$. There are many such methods to produce such a uniform random matrix, with varying computational costs, such as via exponentiation of Lie generator scaled by an appropriately drawn random variable, the Gram-Schmidt procedure [30], and many others [31, 32]. The existing matrix-multiplication procedure is computationally cumbersome so that simpler formulations may be desirable. The following proposed forms are only a starting point for converting anisotropic functions directly to stochastically-isotropic forms. In practice, isotropic functions should be constructed from the ground up rather than merely analogous functions converted from existing anisotropic ones.

For the example of standard dropout, shown in Eqn. 28, it can be made stochastically-isotropic by including the basis-transform shown in Eqn. 29. 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 .

$$\vec{x}' = S_a \sum_{i=1}^N M_i (\vec{x} \cdot \hat{e}_i) \hat{e}_i \quad (28)$$

$$\vec{x}' = S_{si} \sum_{i=1}^N \left(\mathbf{B} \vec{M} \cdot \hat{e}_i \right) (\vec{x} \cdot \hat{e}_i) \hat{e}_i \quad (29)$$

Similar formulations, such as RotationOut by Hu and Poczos [33], showed generally improved performance when the basis is effectively stochastically rotated. However, this method remains stochastically anisotropic as the rotations are generated through Given’s rotations [34], which is not uniform over the space of special-orthogonal matrices — a necessity for full stochastic-isotropy. Nevertheless, the implementation by Hu and Poczos [33] is somewhat encouraging.

This procedure can be generalised and applied to any existing anisotropic function. Yet, 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

A curious extension, particularly to stochastically isotropic dropout, would be to correlate the random bases in time. This may produce a time-like structure in a network’s embedded activation distribution.

If one imagines a random walk of the rotation matrices: $\text{SO}(n) \ni \mathbf{R}^{(\mathbf{t}+\mathbf{dt})} = \mathbf{R}^{(\mathbf{t})} \delta \mathbf{R}$, with $\delta \mathbf{R} = e^{\mathbf{r} \cdot \vec{n}}$, with \mathbf{r} being the corresponding (normalised) anti-symmetric generators for rotations and $\vec{n} \sim \mathcal{N}(\vec{0}, \sigma \mathbf{I}_n)$ 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. 30.

$$\Gamma = \max \left(0, \max \left(\lambda + \frac{|A_n|}{|I_n|} (\lambda - 1), \min \left(1, \min \left(\lambda + \frac{|A_n|}{|I_n|} \lambda, \Gamma \right) \right) \right) \right) \quad (30)$$

659 Leading to $q = \lambda + \frac{|I_n|}{|A_n|} (\lambda - \Gamma)$. Then use one Bernoulli function across all active neurons
660 using $\mathbb{R}^{|A_n|} \ni \vec{D}_{(A)}^{(n)} \sim \text{BernoulliDist.}^{|A_n|}(q)$ likewise for inactive neurons $\mathbb{R}^{|I_n|} \ni \vec{D}_{(I)}^{(n)} \sim$
661 $\text{BernoulliDist.}^{|I_n|}(\Gamma)$. Therefore, correlating the inactive neurons across the time steps, whilst still
662 introducing a degree of random dropout. Thus, the ‘basis of dropout’ undergoes a random walk at
663 every time step, and neurons are randomly chosen to be dropped from the network, with a differing
664 likelihood if they were just previously dropped. The coherence time can be adjusted through Γ , for
665 the specific time-dependent task needed.

666 This creates a link between the stimulus’s presentation time to the network and the neurons it alters,
667 such that stimuli presented in a smaller time window perturb a similar subset of the network’s neurons.
668 This may produce an encoding similar to that found in human cognition, where neurons are thought
669 to go through excitability cycles of slightly differing frequencies and phases. When the excitability
670 is higher, information (engrams) preferentially encodes upon those neurons [35, 36]. As groups of
671 neurons begin to decohere, some overlap remains, such that memories are interlaced if they occur
672 within a temporal window of coherence. *This potentially gives neural networks using isotropic*
673 *dropout an advantage in time-series data.*

D Potential Applications

Besides the proposed general applicability of the isotropic modifications, below are some places where they may yield significant benefit in performance or enable desirable behaviours in networks.

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 especially benefit from its inclusion. One of these is the self-attention step of transformers [11], where isotropic-tanh may be of particular benefit, in replacing the softmax operation [37].

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. As a consequence, it is non-negative, and there are regimes where this may be limiting.

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 stimuli may be a continuous quantity, but its two extremes are mutually exclusive. Many of these semantics exist in the real world: daytime-to-nighttime. These could be represented through a zero-to-one scale; however, perhaps a $[-1, 1]$ scale may be a better representations, with zero better assigned as a neutral middle point. This is because in the linear features hypothesis, often magnitude is indicative of the strength of presence of a stimuli. 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 force a semantic correction to an activation in transformers even when it is inappropriate or force the existence of a near-zero value vector.**

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 quite 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 $QK^T = \vec{x}^T W_Q^T W_K \vec{x} \hat{=} \vec{x}^T W_{kq}^T \vec{x}$, is also basis-independent in \vec{x} , which 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. 31, which will be explored in future work.

$$\text{Attention}(Q, K, V) = \text{Isotropic-Tanh}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (31)$$

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. So, one may hybridise these approaches if the task necessitates.

D.2 Real-Time Dynamical Network Topology

An appealing feature of isotropic deep learning is the relation displayed in Eqn. 15, 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 particular gauge can be chosen that expresses the weights in a beneficial basis.

One such basis may be a magnitude ordering of the singular values for the matrices. One could then set a threshold for the singular value to determine if each corresponding direction in such a matrix has a meaningful contribution to the overall functionality. It can be pruned with little adverse effect on the network if it is deemed to have negligible value.

Moreover, ζ latent neurons can be included, with zero-initialised singular values fully connected to existing neurons. 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 with task-necessitated demand, with minimal impact to performance with these actions. Due to the continuous rotational symmetry available, this is enabled through an isotropic functional form. It poses an interesting research direction, where transfer learning and task-swapping may become more straightforward. 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 trivially extended to convolutional networks, allowing a dynamic number of kernels.

This could offer substantial insight into how parameters may be shared between tasks in real-time. For example, we may postulate that if a new dataset is introduced partway through training on a different dataset, there might be a short-term parameter increase until the network parameter-sharing begins, followed by a phase of pruning until a more compact architecture is reached. These network dynamics may be incredibly insightful.

It appears it may side-step the lottery ticket hypothesis in choosing optimal network size before training. Due to the computational cost, this does not need to be computed at every step, only periodically, and can be performed layerwise.

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 each layer. An example of this is shown in Eqn. 32 or a more general construction in Eqn. 33, for weight matrices \mathbf{W}^j and \mathbf{W}'^j , biases \vec{b}^j and \vec{b}'^j and an activation function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

$$\vec{x}^{l+1} = \sum_j \mathbf{f} \left(\mathbf{W}^j \vec{x}^l + \vec{b}^j \right) \quad (32)$$

$$\vec{x}^{l+1} = \sum_j \mathbf{W}'^j \mathbf{f} \left(\mathbf{W}^j \vec{x}^l + \vec{b}^j \right) + \vec{b}'^j \quad (33)$$

This is effectively a sum over several a feed-forward layer, 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.

D.4 Isotropic Representations May Aid Semantic Alignment

An emerging interdisciplinary field of semantic alignment [38] are trying to produce comparable representations between deep learning and the brain. The author feels it is worth investigating how the representations produced by the Isotropic Deep Learning approach may aid this. 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 of isotropic deep learning forming more ‘natural’ semantic structure in representations.

Current anisotropic networks are imparting a significant effect on the representations [1], producing activation alignment with the discrete distinguished directions produced by the anisotropic functional forms. This is producing an artificial, and possibly unnatural, anisotropic distortion onto the distribution. This may representation basis-dependence may jeopardise efforts to produce representational alignment. Particularly removing anisotropy may help with alignment methods which use continuous rotation like in Williams et al. [39]’s work, since the inductive bias of isotropy is equivariance to continuous rotation.

Despite this, the brain is unlikely to be isotropic due to the approach’s delocalised functional forms. In isotropic networks, neurons instead act as a collective and are arbitrarily decomposable into any

769 set of individual neurons, due to the gauge invariance. Consequently, there is likely to be no single
770 identifiable and agreed upon definition of a neuron in an isotropic network. This is fundamentally
771 incompatible with the structures seen in the brain. However, despite the incompatibility of functional
772 forms with biological neuron behaviour, representations may differ. Representational distortion in
773 deep learning does not imply that the brain also produces anisotropic and discrete features when
774 time-averaging the firings of neurons.

775 A similar approach may be extendable to inferring meaning from undeciphered languages. If
776 isotropy produces a continuous representation, free from basis-distortions, then one may expect a
777 more structured and interpolatable semantic structure. One may speculate whether an approximately
778 language-agnostic structure may develop in similar analogy to representational alignment, which
779 is either assuming or encouraging model-agnostic representation structure. Hence, there may be a
780 chance that a representation free from artificial anisotropies may aid in deducing a representational
781 alignment between known and unknown vocabulary for decipherment. It is speculated that isotropic
782 networks may be especially beneficial for this due to their removal of artificial geometric inductive
783 biases on the representations.

784 Despite this, the success of ensemble models may limit this alignment. Ensemble models use a
785 diverse set of individual models to collectively approximate a task solution. The model diversity
786 would suggest that there are more minima than those connected through a permutation or continuous
787 symmetry, which would not yield diverse individual solutions as are functionally identical. Therefore,
788 this may challenge any assumption of a universal and comparable representational structure for
789 semantics. Despite this, they may produce different approximations to a universal representation.
790 Substantial testing will elucidate this, and the basis-undistorted representations produced by isotropic
791 networks may be particularly beneficial in this research.

E Distinction from Equivariant Networks of Geometric Deep Learning

Both equivariant networks and the proposed isotropic deep learning use an equivariant relation in their definition, which may make them appear similar due to a shared symmetry formalism. However, they differ substantially in how this relation is implemented and motivated, as well as in its consequences and the origin of the symmetry. This section will review those differences. Starting with a discussion of the closest key-concepts within geometric deep learning, namely Equivariant Group-Convolutions [40] and discussion of Steerable-CNNs [41], Harmonic networks [42] and Spherical-CNNs [43]. Following this is a summary of similarities to isotropic deep learning between these methods. Finally, crucial differences will be made clear to show isotropic deep learning’s distinction from these methods and the geometric deep learning paradigm as a whole.

The approach of Cohen and Welling [40]’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 discrete group \mathcal{G} . This symmetry group is chosen a priori, *defined by the known symmetries of the task as an inductive bias* and the symmetry-respecting constraint is applied to the *whole* network architecture. This can have a wealth of benefits: increased weight-sharing efficiency, physically accurate modelling and a resultant increased expressive capacity.

If the task initially has its data distributed over a particular linear base space and obeys a symmetry of that space, then the data can be ‘lifted’ to a symmetry group acting on that space in the equivariant networks framework. This symmetry is displayed by data and feature maps being modelled as a map: $f : \mathcal{G} \rightarrow \mathbb{R}^n$. Every element within the group is assigned an n -dimensional vector by f such that the vectors are interrelated through group action. This may be intuited as an enlarged representation space for the activations due to these interrelated copies. The convolution operation then preserves this discrete symmetry in its transform to produce new features for the group. These activation spaces remain enlarged to accommodate the various poses due to the symmetry. So the geometric structure is increased, and possibly semantic information is enriched by better capturing the geometrical group structure.

The group-convolution is then implemented as a modification of the classic discrete convolution operation: by applying the filter over the group, as shown in Eqn. 34 — for more details of precise implementation see Cohen and Welling [40]. Where k indexes the filter ψ . Note that 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 $f : \mathcal{G} \rightarrow \mathbb{R}^n$. This can also be viewed as augmenting the number of filters within the convolution with each action. Crucially, this is achieved more efficiently in practice through indexing, exploiting the group’s structure. Overall, this adapts the convolutional operation and padding to respect the symmetries in the underlying data structure, but preserves the functional form of surrounding operations like the activation functions, which is demonstrated to be unaffected by such symmetry constraints. Particularly, the non-linearities must remain elementwise in functional form to preserve this equivariance of the convolution in G-CNNs. This is because the elementwise nature commutes with the group action, so preserving equivariance.

$$[f \star \psi](g) = \sum_{h \in \mathcal{G}} \sum_k f_k(h) \psi_k(g^{-1}h) \quad (34)$$

In the subsequent works of [41], [42] and [43], they make considerable progress in developing this paradigm by extending the architectures and tools. In [41], 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, producing a more parameter-efficient construction. In [42], the authors then use the steerable filters to construct equivariance to continuous patch rotation using finite filters and spherical harmonic functions exhibiting the desirable rotational equivariance. With [43], they generalise these concepts for images over a spherical shell, S^2 , and lift it to an $SO(3)$ continuous symmetry equivariance, using a fourier transform-like method. Through these and others, networks are made equivariant to discrete group transforms and extended up to specific continuous group transforms. This subfield is rich with many other discoveries along the same vein; however, other further examples shall not be

discussed since they diverge from a similar seeming construction of isotropic networks, and the key differences can be addressed with these existing examples.

Overall, the similarities between approaches are a tangential use of an equivariance relation as a core principle and a group-theoretic framework, particularly at its most similar, a continuous rotational equivariance of an aspect of deep learning. Similarly related are other geometrical concepts such as vector spaces, Lie groups and gauges. They may also both improve representations for physics-related tasks, where vector space construction can be crucial. Although Isotropic deep learning is of geometrical and deep learning construction, it does not sit cleanly into the current field of geometrical deep learning [44]. Instead, it is constructed around the geometry of embedded representations, as internal symmetries rather than a network-wide externally applied symmetry instilled by a predominantly task-dependent inductive bias.

The core of the differences between Equivariant networks and Isotropic deep learning is the wholly different contexts in which the symmetry arises and its consequences for the network.

For equivariant networks, the symmetry originates from the task itself: the data is lifted from a base space onto a symmetry group, which the network is then designed around to explicitly enforce the external symmetry of the data domain into its solutions. Meanwhile, for isotropy, symmetry is applied at the level of activation vector spaces through symmetry in the class of network functions. The latter originates from an argument of representational geometry not being arbitrarily deformed due to distinguished directions, typically incidentally, imposed by humans. It is an equivariance constraint on the network’s functional forms rather than an equivariance of the network as a whole: a more local vs a global approach. A symmetry of data, which is task-necessitated, versus a general symmetry of internal representational geometry.

This highlights the differing motivations: the injection of highly specific task-aligned inductive bias, informed by the task, 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. Whilst isotropy is the removal of a usually unintended inductive bias, the artificial basis-dependent anisotropies. Thus, isotropy is motivated as a minimal inductive bias as a new default for broader applicability, as a less arbitrary and arguably more natural geometry for representations by removing basis-dependence from functional forms - creating a task-agnostic inductive bias *unless* a priori task-specific knowledge is known for a problem where a strong constraint should be added.

A further difference is that the Isotropic symmetry is *only* enforced in the internal representation spaces, not even necessarily preserved through the transformations between the chain of vector spaces within a network. As a consequence, the symmetries themselves and the effect on activations also differ substantially. For isotropic networks, the equivariance is constructed from a family of symmetry classes, e.g. special orthogonal, where particular layers then acquire a specific instance of this family to be equivariant to. So a general principle of SO family symmetry enforcement on representations, which then functional forms have 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 $\mathbf{f} : \mathbb{R}^l \rightarrow \mathbb{R}^l$. As a consequence, the SO (l) is used in its equivariance to define \mathbf{f} ’s functional form. Therefore, per layer, a specific symmetry from a symmetry family is used, whose functional forms are equivariant. An equivariant network is made equivariant to a specific instance of a symmetry class, which is task-necessitated. Hence, generally, isotropy is 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 symmetry⁴.

Moreover, this has very different consequences for the vector spaces themselves. The equivariant network modifies the dimensionality, or construction of vector spaces, to enforce a global symmetry; whereas, isotropy leaves the vector space construction unchanged, affecting only certain transforms between them. The symmetry constraint in Equivariant networks produces an enlarged dimensionality of the activation spaces to accommodate the group structure in [40]. However, this differs for later irreducible representations like those in [41]. Both leave a stark difference in the structure of the activation space. Isotropy does not affect construction since a specific symmetry is not enforced globally, but a family of symmetries is applied only locally. 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 broad applicability.

⁴but may only if desirable

This is a consequence of isotropy’s foundational derivation, which will only be briefly outlined as it is being substantially developed for future publications. Fundamentally, the isotropic symmetry can be argued to emerge from the architecture itself. 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 arise at the level of arbitrary graph structures. When one examines an arbitrary directed graph and endows its nodes with continuous activations that can be grouped and systematically divided up vector spaces, chained through continuous maps, one can apply group actions to this continuous-valued node topology, leaving the endowed directed graph unchanged. Here, the continuous isotropic symmetry arises and can be broken into discrete permutation symmetries characterising modern deep learning. Consequently, one can show that the isotropic symmetry originates from the arbitrary network architecture itself, rather than a task-dependent symmetry informing the development of a specific architecture, as in equivariant networks. From this, isotropy is applied to functional forms across the board: activation functions, initialisations, normalisations, regularisers, optimisers, etc., but not necessarily affecting architectures. This sets the approaches within two very different directions, regarding the origin of its symmetry and its relation to architecture. In this regard, isotropy can be viewed as a more fundamental and natural behaviour of the architectures than its broken symmetry counterpart of current anisotropic deep learning. Some overlap may occur, but isotropy is predominantly a symmetry from an architecture influencing functional forms. In contrast, equivariant networks are a symmetry of the underlying data structure that influences the architecture.

If one does alter the Isotropic deep learning architecture such that the family of isotropic symmetries is reduced to a specific instance of a symmetry, e.g. $SO(3)$, then one could argue that the network is also globally equivariant, so a type of equivariant network. This establishes a tentative bridge between the two approaches. However, this is a very different symmetry to the $SO(3)$ in [43]’s work. This is an $SO(3)$ in the activation space vectors, not a symmetry in the coordinates of the base space, like [43] achieved, and must not be conflated. Therefore, since it remains a symmetry of representations, not data structure, one must broaden the definition of equivariant networks to establish any overlap with isotropic deep learning.

This is not to say one is better or more principled, or even parallel techniques; they are constructed for differing purposes. One for respecting a specific symmetry in solutions present in a particular task, such as in many physics-related problems, whilst one removes a basis dependence which may arbitrarily and anisotropically affect the distribution of representations in all problems, motivating it for universal adoption in deep learning. Isotropy is a proposal of basis independence and gauge invariance. These are differing proposals, an external-geometric-symmetric framework and an internal-algebraic-symmetric framework, both using equivariances as a core feature. This shows that isotropy is a framework with broader and more flexible applicability, justifying the assertion that one day it may be a better default inductive bias. Still, it does not displace the case-by-case application of a strong inductive bias in equivariant networks. As it stands, the substantial differences in approach to symmetry make the frameworks incompatible, seen through the restriction of pointwise nonlinearities in equivariant networks. This mutual exclusivity might be bridged within further work, only if a task-dependent inductive bias makes it desirable to do so.

Concluding: Isotropic deep learning and the various generalisations of Equivariant network share a similarity in their fundamental construction from an equivariance relation and philosophical focus on symmetry principles. Equivariant networks are made to enforce an end-to-end symmetry deduced from their data structure, enforced by architecture modifications like group-convolution [40]. This dramatically increases parameter efficiency and ensures physical solutions for specific problems. Isotropic deep learning promotes the existing discrete rotational 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. Therefore, equivariant networks are instilled with an a priori respect for a task-dependent symmetry, whereas isotropy is being developed as a universal new default for functional forms.

Hence, isotropic deep learning is both a framework of geometry and deep learning, but currently is a significant deviation from the foundational blueprint within Geometric deep learning [44], despite the

955 use of equivariance and symmetries. They have substantial differences in how and which symmetries
956 arise and how they affect the models, how activations are represented in networks and how parameters
957 are constructed. Isotropic Deep Learning redefines the form for parameter initialisation, rather than
958 restructuring parameters, so as not to affect the layer structure, which can remain dense.

959 Currently, the approaches are essentially mutually exclusive. However, as shown, specific instances
960 of isotropic deep learning can be made to express some characterising properties of an equivariant
961 network, so they retain some very minimal overlap. Isotropic deep learning more suitably falls within
962 the class of geometries of representations, more closely related to the work of Elhage et al. [16], Olah
963 et al. [22], Carter et al. [45]. This interdisciplinary approach, alongside work such as [16], may be
964 more appropriately classified under a name such as *‘representational geometry in deep learning’*.

965 F Taxonomy of Functional Forms

966 Isotropic deep learning is centred around the equivariance to special-orthogonal group actions. This
 967 is demonstrated through an equivariance relation, such as in *Eqn. 35* defining a functional form for
 968 $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

$$\forall \mathbf{R} \in \text{SO}(n) : [\mathbf{R}, \mathbf{f}] = 0 \quad (35)$$

969 It has already been shown how current deep learning's functional forms can be connected through
 970 a similar relations, as an equivariance of functional forms to the permutation group, as shown in
 971 *Eqn. 36*.

$$\forall \mathbf{P} \in \mathcal{S}_n : [\mathbf{P}, \mathbf{f}] = 0 \quad (36)$$

972 Similarly, quasi-isotropic functional forms (though not the specific instance in *Eqn. 23*) can be
 973 connected through a discrete rotational symmetry group ψ_n , with set cardinality $|\mathcal{S}_n| \ll |\psi|$. This
 974 can be represented through *Eqn. 37*.

$$\forall \Psi \in \psi_n : [\Psi, \mathbf{f}] = 0 \quad (37)$$

975 This taxonomic approach could be used as both a unifying perspective and generalised approach
 976 to deep learning. Considering the standard permutation, a specific discrete-rotational, continuous-
 977 rotational and general linear symmetries for an \mathbb{R}^n space associated with a single layer, can be
 978 constructed such that they are unified under a group heirarchy: $\mathcal{S}_n \subset \psi_n \subset \text{SO}(n) \subset \text{GL}(n)$.
 979 Applying each of these to functional forms produces, current deep learning, quasi-isotropic deep
 980 learning, isotropic deep learning and linear approximators respectively. Hence, several forms of ma-
 981 chine learning can be said to be unified together under such approach, including linear approximators.

982 Hence, it is natural to generalise this functional form equivariance relationship more broadly. For
 983 a symmetry family \mathcal{G} , one could impose a functional form equivariance as shown in *Eqn. 38*. In
 984 general this would need to be a task-motivated inductive bias. This could produce a functional form
 985 taxonomy for deep learning approaches.

$$\forall \mathbf{G} \in \mathcal{G} : [\mathbf{G}, \mathbf{f}] = 0 \quad (38)$$

986 These symmetries can all be associated to symmetries of directed graphs, endowed with continuous
 987 activations, as briefly discussed in *App. E*. In effect, one can construct an directed graph which
 988 represents the neuron connectivity of an arbitrary architecture. If nodes within this graph are
 989 organised and divided into groups of neuron layers, then the continuous activations assigned to these
 990 nodes can be said to form an \mathbb{R}^n activation space. Prior to parameter initialisation and specific
 991 functional form implementation, these layers could be considered to feature symmetries of their \mathbb{R}^n
 992 space. Choosing such a symmetry group, which leaves these spaces invariant, can then be used with
 993 the general equivariance relation of *Eqn. 38* to produce functional forms and parameter initialisations
 994 obeying the symmetry. It proposed that this could be a powerful unifying direction for deep learning,
 995 which is may be worth considerable exploration and may generate and categories what could be
 996 considered novel forms of deep learning.

997 **G Historical Precedent**

998 [Section under development]