Towards Isotropic Deep Learning
A New Default Inductive Bias

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

This position paper explores the often overlooked geometric implications of current functional forms used in deep learning and proposes a new paradigm to be termed *Isotropic Deep Learning*. Principally the existing, and often underappreciated, discrete inductive bias is elevated to a continuous one within this framework. It has been demonstrated that the functional forms of current deep learning influence the activation distributions. Through training, broken symmetries in functional forms can induce broken symmetries in embedded representations. Thus producing a geometric artefact in representations which is not task-necessitated, and solely due to human-imposed choices of functional forms. There appears to be no strong a priori justification for why such a representation or functional form is desirable, while this paper proposes several detrimental effects of the current formulation. As a result, a modified framework for functional forms will be explored with the goal of unconstraining representations by elevating a rotational symmetry-inductive bias throughout the network. This framework is encouraged to be adopted as a new default. This direction is proposed to improve network performance. Preliminary functions are proposed, including activation functions. Since this overhauls almost all functional forms characterising modern deep learning, it is suggested that this shift may constitute a new branch of deep learning.

1 Introduction

Current deep learning models typically employ an elementwise functional form . This is particularly evident in activation functions, sometimes called ridged activation functions . These functions are often displayed univariately as shown in *Eqn.* 1, with σ being a specific activation function implemented, e.g. ReLU, Tanh, etc.

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

However, this display choice obfuscates a crucial (standard) basis dependence. This is explicitly displayed in, what should be considered a more implementation-correct multivariate form, of *Eqn.* 2. This reveals the functional form's usually hidden $\hat{e}_i$ basis dependence. The multivariate form is depicted for an n neuron layer, with activation vector $\vec{x} \in \mathbb{R}^n$. This standard basis dependence is arbitrary and appears as a historical precedent, rather than appropriate inductive bias, discussed further in *App* A.

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

Due to this basis dependence, non-linear transformations differ angularly in effect Bird [2025]. Therefore, this will be termed an *anisotropic function*, indicating this rotational asymmetry. Due to the pervasive use of these functional forms, including optimisers, normalisers, regularisers, activation functions etc., current deep learning as a paradigm may consequently be termed *anisotropic deep learning*. Despite its implications, this *choice* of basis-dependent anisotropy appears unappreciated and incidental in the development of most contemporary models, with anisotropic forms are almost treated as axiomatic to deep learning rather than a considered choice. Hence, re-evaluating and systematically reformulating this foundational aspect of modern deep learning, with such widereaching consequences, is felt to constitute a new branch: *Isotropic Deep Learning*.

This asymmetry in the non-linear transform is particularly about the standard (Kronecker) basis vectors and their negative $\{+\hat{e}_i, -\hat{e}_i\}_{\forall i}$, due to the elementwise nature. Therefore, it can be said to distinguish the standard basis - a '*distinguished basis*'[1]. For example, the standard basis is visible in *Fig.* 1 showing the mapping of elementwise-$\tanh$ on a variety of test shapes. Most generally, this

Figure 1: [Insert caption here] [Desmos Link]

choice of functional forms can be considered to break *continuous* rotational symmetry, and reduce it to a *discrete* rotational symmetry — a permutation symmetry of the standard basis. In effect, if the function is treated in its multivariate form, it is equivariant to a permutation of the components of its vector decomposed in the standard basis. For an element of the permutation group $\mathbf{P} \in \mathcal{S}_n$, the following equivariance relation holds $f(\mathbf{P}\vec{x}) = \mathbf{P}f(\vec{x})$.

Non-linearities are usually pivotal to the network's ability to achieve a desired computation, as seen through the universal approximation theorem's Cybenko [1989] explicit dependence on the form of the activation function Hornik [1991]. The non-linearities produce differing local transformations, such as stretching, compressing, and generally reshaping a manifold. Consequently, the network may be expected to adapt by moving representations to geometries about these distinguished directions, to use specific local transforms to achieve the desired computation. Hence, an anisotropy about a distinguished basis is induced into the activation distribution shown by.

This has been empirically demonstrated by Bird [2025]: training results in the broken symmetry of the functional forms, inducing a broken symmetry in the activations. Since these non-linear zones are centred around the distinguished bases, the embedded representations are expected to move to beneficial angular arrangements about the arbitrarily imposed privileged basis's geometry. For example, they appear to move towards the non-linearities' extremums, aligned, anti-aligned or other geometries, through training. This may correspond to a local, dense or sparse codings and superposition, respectively.

Therefore, the network has adapted its representations through training, for probably various optimisation reasons, due to these functional form choices. The causal hypothesis aids in explaining the observed tendency of privileged-basis alignment. This is the causal hypothesis underlying the author's position: **functional forms should be a deliberate and considered decision, with a suitably optimal, and minimally harmful, default**. Currently, anisotropy results in a human-caused representational collapse onto the privileged basis, and it is suggested that this is frequently not

[1]This is suggested generalisation from a *'privileged basis'* discussed in Elhage et al. [2022]. The change to *'distinguished basis'* reflects that the basis may be more-or-less aligned to the representation; whereas *'privileged basis'* is felt to suggest a basis which is more aligned to the embedded activations. The term 'basis' will be retained even though the set of '*distinguished vectors*' may also be under-/over complete for spanning the whole space, as demonstrated in Bird [2025]. There may be multiple distinguished bases, such as aligned and anti-aligned with the standard basis for the model.

a task-necessitated collapse. There appears to be little justification as to why this is universally desirable, with several key negative implications discussed in *Sec.* 2. Without a priori justification, this inductive bias may be detrimental to computation, so unconstraining the activation appears generally preferable.

Overall, historic and frequently overlooked functional forms for modern deep learning directly influence the models' activations and therefore behaviour. There exists a functional form basis dependence which appears entirely arbitrary, obsfuscated and hence neglected. Yet, a causal link between this arbitrary basis and activations has been empirically demonstrated, and hence, a resultant effect on the final performance of the model is hypothesised. These are often underappreciated choices which have consequences and should be well-justified and studied. This is the position of the authors.

Throughout the rest of this position paper, **it is argued that a departure from this anisotropic functional form paradigm towards the isotropic paradigm is generally preferable as an inductive bias**, unless otherwise justified. It encourages the reader to be conscious of these choices when designing a model, as well as the usual architectural tool kit. Particularly, isotropic choices, equivalent to basis independence, may be thought to unconstrain the representations into more optimal arrangements for a task. The tenets of this paradigm are suggested for all architectures on general tasks.

2 Hypothesised Problems of Anisotropy

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

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

Furthermore, there are some specific instances when anisotropy may be detrimental to performance, one of those is in the self-attention step of transformers, speculated in *App.* F.1.

2.1 The Neural Refractive Problem

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

It appears to be a phenomenon in all anisotropic activation functions to date. The 'refraction' typically occurs more significantly at larger magnitudes — potentially a failure mode under network extrapolation. Mathematically, this has several representations, a magnitude-varying 'dynamic refraction' shown in *Eqn.* 3 or differentially in *Eqn.* 4. Also defined is a 'static refraction' definition shown in *Eqn.* 5. These are described for a multivariate activation function $\mathbf{f}$ and vector $\vec{x} = \alpha\hat{x}$ where $\hat{x}$ is a unit vector. This relation may be satisfied for a single direction, a subset of the space or all directions $\hat{x} \in \mathcal{X} \subseteq S^n$. The relations generally show how the activation function alters the direction of its input vector in an anisotropic manner.

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

$$\exists \hat{x} \in \mathcal{S}^{n-1}, \exists \alpha_0 : \frac{\partial}{\partial \alpha} \left.\frac{\mathbf{f}\left(\alpha_1\hat{x}\right)}{\left\|\mathbf{f}\left(\alpha_1\hat{x}\right)\right\|}\right|_{\alpha_0} \neq \vec{0} \tag{4}$$

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

It can be seen that along a straight-line trajectory in direction $\hat{x}$, the result of the activation function is a curved line if dynamically refracted. Therefore, if the linear feature hypothesis is followed, *every* linear feature, in these directions, becomes curved following the activation function. The network may exploit some of this curvature to construct new linear features in the subsequent layers; however, there may be many instances where this curvature is detrimental to established semantics. The network may lose semantic separability, produce magnitude-based semantic inconsistency or compensatory maladaptations in later layers, due to this refraction. This may hinder network performance and is resolved by isotropic choices.

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

This may be particularly consequential for functions such as Sigmoid and Tanh, since large magnitude inputs end up at particular limit points (discussed as trivial representational alignments in Bird [2025]). For example, Tanh produces the limit points shown in *Eqn.* 6 when $\hat{x}\cdot\hat{e}_i \neq 0$ for all i. If there exists an i, s.t. $\hat{x}\cdot\hat{e}_i = 0$, then the corresponding index also has a 0 as a limit point. A fully-connected layer can only effectively separate two such converging directions at a time, which are then further curved by a subsequent activation function.

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

Consequently, semantic separability is lost except for 3^n discrete limit points for Tanh and Sigmoid. Therefore, embedded activations may be expected to align with these limit points, an empirically observed tendency Bird [2025]. Similarly, ReLU has one distinct limit point, $\vec{0}$, but otherwise an orthant unaffected by neural refraction. The authors speculate whether this is an additional reason for the success of ReLU, due to only a subset of directions experiencing the neural refraction phenomena. Furthermore, this would suggest an advantage of Leaky-ReLU: despite featuring static-refraction, directions do not become overlapped, so semantic separability is retained. Otherwise, the network may expend training time on producing robust semantic separability, a needless compensatory adaptation, which may lower representational capacity or extend training as a result.

More generally, dynamic deflection of trajectories may cause semantic ambiguity for the network, where only samples interpolable from training samples are reliably semantically identifiable. Particularly, *the larger the deflection, the greater the semantic ambiguity expected*. More considerable deflections typically occur at larger magnitudes in many current functions. Therefore, a magnitude-dependent semantic inconsistency may arise due to such deflections. A deflection function can be a trivial diagnostic measure defined by *Eqn.* 7 for a particular activation function.

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

This may 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 very large magnitude of the linear feature. 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 $\left\|\mathbf{f}\left(\alpha\hat{x}\right)\right\| = \sigma\left(\alpha\right)$ in *Eqn.* 9.

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

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

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

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

In standard notation, *Eqn.* 10 can be rewritten into the *final functional form for isotropic activation functions* shown in *Eqn.* 11.

$$\mathbf{f}\left(\vec{x}\right)=\sigma\left(\|\vec{x}\|\right)\hat{x} \tag{11}$$

This can be generalised as a result of rotational equivariance of the function. This can be expressed as a condition in *Eqn.* 12, which uses a commutator bracket for convenience, with $\forall\mathbf{R}\in\mathrm{SO}\left(n\right)$. This bracket can be used to similarly define the current permutation-anistropic 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 discussed in *App* **??**.

$$\left[\mathbf{R},\mathbf{f}\right]=\left(\mathbf{Rf}-\mathbf{Rf}\right)=\vec{0} \tag{12}$$

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

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 $\mathcal{O}\left(n\right)$ 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.

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

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

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

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 then fails excessively. Finally, it may be expected that the network produces compensatory adaptations for the phenomena. Since neural refraction is a non-linear and anisotropic phenomenon, it cannot be inverted by a single subsequent layer, potentially wasting training time on corrections to the activation distributions due to unintended refraction. These refraction corrections may be limited in scope and produce maladaptive behaviour outside of the original training data.

2.2 Weight Locking, Optimisation Barriers and Disconnected Basins

'Weight locking' is a term to describe how particulary the weight parameter[2] may suffer from being stuck in local-minima found further into loss valleys encountered after a sufficient amount of training. This arises only due to the anisotropic functional form's *discrete* permutation symmetry.

Qualitatively, this is because the semantically meaningful linear features tend to become aligned with geometric positions about the distinguished bases. Any small perturbation to a parameter may misalign activations to the network's existing 'understood' semantics. In effect, further small perturbations to the parameters may move activations from a semantically-aligned to a semantically-dislocated state, thus the activation's meaning becomes ambiguous, forfeiting performance and resulting in a 'false' local-minima created only by the *discrete* nature of the permutation symmetry. It may also suggest that the optimisation barrier may be some function of the angular separation of the semantic directions. Consequently, creating a plethora of architectural local minima in the space. Only sufficiently large perturbations to a parameter may move activations between two differing semantically aligned directions.

[2]Though similarly applies to a 'locking' of the bias to $\vec{0}$.

This semantic dislocation is an emergent consequence of breaking the continuous symmetric forms, as without continuous rotational symmetry, it results in the dual phenomena of connectivity of many minima basins being lost. In effect, enforcing the isotropy constraints results in sets of continuously connected local minima which can be smoothly transformed into one another, by corresponding parameter rotations shown in *Eqn.* 15, a consequence of *Eqn.* 11. If this is downgraded to discrete rotational symmetry (i.e. permutation symmetry), then artificial optimisation barriers may reemerge in these basins. In this case, only a sufficiently large perturbation to the parameters may dislodge the network into a more optimal minima, while more minor perturbations are insufficient. Effectively, the discrete permutation symmetry may result in a discretised lattice solution for the parameters, much like how it breaks the symmetry of activations through training too Bird [2025].

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

This exists as a qualitative intuition since, until robust methods to determine semantically meaningful directions are produced, this hypothesis remains difficult to verify. Nevertheless, steps can be taken immediately to counteract the problem, and this is to introduce isotropy to connect these minima.

2.3 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. It may be conjectured that semantically meaningful linear directions may also be encouraged to discretise, aligning with these anisotropic embeddings. This generally appears to be the case, with notable counterexamples. Moreover, these counterexamples are for networks which *do not feature anisotropic functional forms*.

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 important nuance. It appears a poor inductive bias to have functional forms encourage discretised representations. Isotropic functions do not prevent discrete semantics, but they don't encourage them either, enabling continuous ones since they generally unconstrain the representations. 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. 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 may also produce a better organised semantic map at each layer of the network since intermediate representations may now relate otherwise discrete features, and therefore, this connectivity can bring them into continuous proximity (which 'weight locking' may typically prevent). This may additionally aid researchers in comparing representational alignment between models and biology[3].

Therefore, in general settings, the inductive bias of isotropy appears more appropriate as a default unless justification for anisotropy is present. It is an inductive bias that meaningful semantics are often continuous and interpolatable, 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. It suggests that a continuous rotational symmetry allows a continuous embedding, since functional forms do not induce direction-based symmetry breaking. It is hoped this will enable networks to acquire a more optimal and natural representation embedding for the given task.

[3]Though there is no guarantee that *all* basins are connected, so therefore would not necessarily be alignable through continuous rotation transforms. Furthermore, the success of ensemble methods with diverse constituent models would suggest that diverse disconnected basins remain, complicating representational alignment even under isotropy conditions, though isotropy may help somewhat.

3 Alternative View: Embedding Folding

4 Isotropic Implementations

This position paper argues for the implementation of isotropic functional forms into neural networks as a default inductive bias. Near-term adoption may be rate-limited due to the development of suitable functions, particularly since *anisotropic deep learning* has a substantial head start and analogues to existing functions are not so trivial to produce. Despite this, several preliminary implementations are outlined in this section as a starting point; however, these functions are far from optimal and substantial research and development are required to bring isotropic deep learning into practicality.

Nevertheless, below is a non-exhaustive list of activation functions and some of their consequences. A brief summary of other functions, including optimisers, regularisers, and normalisers, is also discussed. It is hoped that a directed search for new functions may occur.

4.1 Activation Functions

As stated, the isotropic functional form for activation functions is given in *Eqn.* 13. In *Tab.* 4.1, it is 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, but will be outlined 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**
$\mathbf{f}(\vec{x}) = \sum_{i=1}^{N} \sigma\left(\lVert\vec{x} - \vec{c}_i\rVert\right) \hat{e}_i$	$\mathbf{f}(\vec{x}) = \sum_{i=1}^{n} \sigma\left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i$	$\mathbf{f}(\vec{x}) = \sigma\left(\lVert\vec{x}\rVert\right) \hat{x}$

Isotropic-Tanh is described in *Eqn.* 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.

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

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

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

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

In effect, 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.

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}; R_0, R_\infty\right) = \min\left(\max\left(\lVert\vec{x}\rVert - R_0, 0\right), R_\infty\right) \hat{x} \tag{18}$$

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \quad \vec{x} \mapsto \mathbf{f}\left(\vec{x}; R_0\right) = \begin{cases} \vec{0} & : \quad \lVert\vec{x}\rVert < R_0 \\ \vec{x} & : \quad \lVert\vec{x}\rVert \geq R_0 \end{cases} \tag{19}$$

These activation functions continue to use $\mathbf{f}\left(\vec{0}\right) = \vec{0}$ property.

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 $0 < \alpha \ll 1$.

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

Isotropic-Soft-ReLU , $\alpha = 0$, 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.

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

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

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

5 Argument Against Isotropy (and Quasi-Isotropic Activation Functions)

It is argued that anisotropies result in an activation distribution shift, which may be detrimental to the network's performance. This is because this inductive bias is typically introduced universally, and if no justification exists for this particular distribution, then it may be a suboptimal imposition by the network designer. However, it could also be argued that some symmetry-breaking anisotropy may be beneficial, by clustering parts of the activation distribution, leading to classifications that develop more quickly. In classification, one of the most common applications of deep learning, this clustering may be a suitable a priori justification for anisotropy. Therefore, introducing isotropy may limit the network's performance in this case.

Despite this, current activation functions produce anisotropies along a Cartesian grid, due to their standard basis dependence. This particular arrangement does not seem justified through classification. Anisotropies can be introduced in a more general arrangement, enabling a more uniform distribution of directions from which semantics could develop.

A middleground may be to relax the hard isotropy condition and introduce slight symmetry breaking in many directions. Then the network has many distinguished vectors, a subset of which it may align its representations too in a task-dependent manner. Therefore, it does not favour a particular basis, but still introducing some desirable consequences of anisotropy. If one further restricts the functions to not feature dynamic refraction, then it limits detrimental anisotropic effects.

This approximately basis-free anisotropy appears preferable since it does not constrain representations to the arbitrary standard basis. It can be achieved by introducing many small perturbations to the direction unit-vector only, producing a softer symmetry breaking.

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

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

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

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

5.1 Real-Time Dynamical Network Topology

An appealing feature of isotropic deep learning is the relation displayed in *Eqn.* 15, showing that due 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 which 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. If it is deemed to have negligible value, it can be pruned with little adverse effect on the network.

Moreover, ζ latent neurons can be included, with zero-initialised singular values fully connected to existing neurons. 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. This is only enabled through an isotropic functional form, due to the continuous rotational symmetry available. 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 prior to training. Due to the computational cost, this does not need to be computed at every step, only periodically, and can be performed layerwise.

6 Conclusion

In this position paper, the isotropic functional form is proposed as a better default inductive bias for deep learning. Current forms have been demonstrated in the literature to produce task-unmotivated representational artifacts which may limit semantic expressibility of the networks. It is further argued that the current anisotropic functional forms may have detrimental effects in performance and learning, through the 'neural refractive problem', 'weight locking' and 'discrete semantics'. Hence, removing such constrains from the model is argued to also unconstrain the representations from any particular basis. It is then expected that the network can produce a more natural activation representation based upon task neccessities, rather than human imposed by functional forms.

The adjustment to functional forms is through promoting the existing discrete rotational symmetry (permutation symmetry) of modern deep learning to a continuous special-orthogonal symmetry. This has substantial consequences for the form of almost every function in the modern day deep learning: activation functions, regularisers, normalisers, optimisers and more. It is proposed that the breadth of this reformulation, is best represented as a novel branch of deep learning: *Isotropic deep learning* to distinguish it from existing paradigms.

In this position paper, several example isotropic functions are showcased as a starting point, yet these are analogues and not expected to be inherently optimal or better since they display superficial their similarity to existing functions. Instead, it is the authors position that this change to isotropic deep learning is generally advantageous, but may need substantial time to development as a paradigm such that better optimised implementations are discovered which suitably leverage the new properties available from isotropy. Therefore, empirical work will be presented in following papers as to not distract from the primary arguments motivating this shift to Isotropic deep learning.

It is hoped that the proposed ideas may stimulate the communities' interest into beginning a search for such Isotropic functions, which will hopefully bring the paradigm into widespread applicability and adoption.

References

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.

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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.

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

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

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

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

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization, Aug 2019. URL https://distill.pub/2017/feature-visualization/.

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

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

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.

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.

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

A Historical Precedent

Neural networks, of all forms, represent their information through a neural code which remains under debate Quiroga et al. [2005], Cammack et al. [2021], Kreiman et al. [2000]. Several codings have been proposed and it appears that initially anisotropic functions may have indirectly arisen from one of these, local coding, which was relatively more common around the advent of artificial neural networks. Local coding is a neuropsychological hypothesis of one neuron's activation representing the presence of one real-world stimulus. This is also commonly known as the grandmother neuron interpretation Gross [2002], Connor [2005], and less so as gnostic neurons and gnostic fields Konorski [1968]. This hypothesis was developed Barlow [1953], Konorski [1968] and debated from the early 1950s through to the 1970s Gross [2002], with continuing discussion to the present day Quiroga et al. [2005], Connor [2005], Graham and Field [2006].

Crucially, McCulloch and Pitts co-authored a paper supporting the presence of several differing pattern feature detectors in frog's retinas Lettvin et al. [1959] expanding on Hartline's earlier work of similar findings Hartline [1938]. Thus, it was shown that retinal ganglions can respond to distinct real-world patterns, so-called 'bug detectors' for the frog, acting much like the description of grandmother neurons. These are the same McCulloch and Pitts who are earlier credited as the inventors of the binary threshold network McCulloch and Pitts [1943] which led to the first perceptron neural network being developed. In the very first sentence of McCulloch and Pitts [1943] work, it states:

> *Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic.*
>
> *–A Logical Calculus of the Ideas Immanent in Nervous Activity McCulloch and Pitts [1943], p. 115.*

Hence this premises artificial neural networks on a binary logic, inducing this early basis-dependent nature. This appears to have implicitely encouraged the the adoption of the heaviside step function in early deep learning, which has a continuous lineage to the vast array of elementwise functions utilised today. This lineage is connected through a series incremental modifications from discrete heaviside step functions, such as differentiable sigmoid-based functions, to non-saturating ReLU through to its variants. Which in-turn may have influenced the wider array of functions, like elementwise dropout or approximating hessians as a diagonal in adaptive optimisers.

Therefore, I argue that the pervasivness of anisotropic deep learning can be traced back to a trajectory set in the early developments in the field — an underappreciated *choice* of functional form influenced, by the discoveries of their time.

A further indication is in one of McCulloch and Pitts [1943]'s concluding statements:

> *[...] pushed to ultimate psychic units or "psychons," for a psychon can be no less than the activity of a single neuron. Since that activity is inherently propositional, all psychic events have an intentional, or "semiotic," character. The "all-or-none" law of these activities, and the conformity of their relations to those of the logic of propositions, insure that the relations of psychons are those of the two-valued logic of propositions. Thus in psychology, introspective, behavioristic or physiological, the fundamental relations are those of two-valued logic.*
>
> *–A Logical Calculus of the Ideas Immanent in Nervous Activity McCulloch and Pitts [1943], p. 131.*

This "psychon", with semiotic qualities and two-valued logic, is highly suggestive of a local coding approach — alligning with the paradigm of vector decomposition in modern neural networks through generalising the two-value logic. This local coding perspective is argued to have led to a preference for elementwise logic and functional forms. Hence, the activation vectors within neural networks are treated as just an array of numbers, as opposed to a direction and magnitude, which is the defining feature of the elementwise anisotropic forms.

As a result, the roots of anisotropies can be seen in neural networks since their first conception. At this time, the non-linear activation function utilised was the heaviside step function and this persisted into the multilayer perceptron Rosenblatt [1958], but later generalised whilst keeping an elementwise form Maas et al. [2013]. These continuous-generalised perceptron layers continue to be prevelent becoming a pervasive backbone of many current models Hochreiter and Schmidhuber

[1997], Krizhevsky et al. [2012], Vaswani et al. [2023], as well as being adapted into other structures such as convolutional neural networks.

[unfinished]

B 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, motivated and 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 Cohen and Welling [2016a] and discussion of Steerable-CNNs Cohen and Welling [2016b], Harmonic networks Worrall et al. [2017] and Spherical-CNNs Cohen et al. [2018]. Following this is a brief summary of similarities to isotropic deep learning between these methods. Finally, crucial differences will be made clear to show isotropic deep learning's distinction to these methods and to geometric deep learning paradigm as a whole.

The approach of Cohen and Welling [2016a]'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 efficiency in weight sharing, 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: $f : \mathcal{G} \to \mathbb{R}^n$. Every element within the group is assigned an n-dimensional vectors, by f, such that the vectors are interelated to one another through the group action. This may be intuited as an enlarged representation space for the activations, due to these interelated 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 accomodate 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.* 25 — for more details of precise implementation see Cohen and Welling [2016a]. 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 $f : \mathcal{G} \to \mathbb{R}^n$. This can also be viewed as augmenting the number of filters within the convolution with each action, but in practice this is achieved more efficiently through indexing exploiting the structure of the group. 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, as the elementwise nature commutes with the group action so preserves equivariance.

$$[f \star \psi](g) = \sum_{h \in \mathcal{G}} \sum_{k} f_k(h)\, \psi_k\left(g^{-1} h\right) \tag{25}$$

In the subsequent works of, Cohen and Welling [2016b], Worrall et al. [2017] and Cohen et al. [2018], they make considerable progress in developing this paradigm, by extending the architectures and tools. In Cohen and Welling [2016b], 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 Worrall et al. [2017], the authors then use the steerable filters, to construct equivariance to continuous patch rotation using finite filters, using spherical harmonic functions which exhibit the desirable rotational equivariance. With Cohen et al. [2018], they generalise these concepts for images over a spherical shell, $\mathcal{S}^2$, and lift it to an $\mathrm{SO}(3)$ continuous symmetry equivariance, using a fourier transform-like method. Through these, and others, networks as a whole are made equivariant to both discrete group transforms and extended upto 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 similar seeming construction of isotropic networks and the key differences can be discussed with these existing examples.

Overall, the similarities between approaches is 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 also both may 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 . Rather, it is constructed around the geometry of embedded representations, as internal symmetries rather than a network-wide externally applied symmetry instilled by a is predominantly task-dependent inductive bias.

The core of the differences between Equivariant networks and Isotropic deep learning is the wholly difference 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 explicitely enforce the external symmetry of the data domain into its solutions. Whereas, for isotropy, the symmetry is applied at the level of activation vector spaces, through a symmetry on the class of network functions. The latter originates from an argument of representational geometry not being arbitrarily deformed due to disinguished 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-neccessitated versus a general symmetry of interal 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 neccesarily 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 actually constructed from a family of symmetry classes, e.g. special orthogonal, where particular layers then acquire a specific instance of this family to be equivarient 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 $\mathbf{f} : \mathbb{R}^l \to \mathbb{R}^l$. As a consequence the $\mathrm{SO}\,(l)$ is used in its equivariance to define $\mathbf{f}$'s functional form. Therefore per-layer a specific symmetry from a symmetry family is used which functional forms are equivariant. Whereas, an equivariant network, is made equivariant to a specific instance of a symmetry class which is task-neccessitated. Hence, generally isotropy is in regard to a symmetry family rather than a specific instance of a symmetry group. In isotropic deep learning the network as a whole does not need to respect a particular symmetry[4].

Moreover, this has very differing consequences for the vector spaces themselves. The equivariant network modify 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 accomodate the group structure in Cohen and Welling [2016a], though this differs for later irreducible representations like those in Cohen and Welling [2016b]. Both leaving a stark difference in the structure of the activation space. Isotropy needn't affect the 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

[4]but may only if desirable

continuous one. It leaves the architecture topology, and consequently vector space construction, unchanged - giving it broad applicability.

This is a consequence of isotropy's foundational derivation, which will only be briefly outlined as is being subtantially 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 arbitrarily number of hidden layers and 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 which can be grouped and systematically divided up vector spaces, chained through continuous maps, then one can apply group actions to this continuous-valued node topology leaving the endowed directed graph unchanged. It is here where the continuous isotropic symmetry arises and can be broken into discrete permutation symmetries characterising modern deep learning. Consequently, one can show the isotropic symmetry originates from the arbitrary network architecture itself, rather than a task-dependent symmetry informing the development of a specific architecture — as is the case in equivariant networks. From this, isotropy is applied to functional forms across the board: activation functions, initialisations, normalisations, regularisers, optimisers, etc., but not neccessarily affecting architectures. This sets the approaches within two very different directions, in terms of the origin of its symmetry and its relation to architectures. In this regard, isotropy can be viewed as a more fundamental and natural behaviour to the architectures than its broken symmetry counterpart of current anisotropic deep learning. Overall, some overlap may occur, but isotropy is predominantly a symmetry from an architecture influencing functional forms, whereas equivariant networks are a symmetry of the underlying data-structure influencing 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. $\mathrm{SO}(3)$, 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 actually a very different symmetry to the $\mathrm{SO}(3)$ in Cohen et al. [2018]'s work. This is an $\mathrm{SO}(3)$ in the activation space vectors, not a symmetry in the coordinates of the base space, like Cohen et al. [2018] achieved, and must not be conflated. Therefore, since it remains a symmetry of representations, not data structure, one would have to 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 particularly that they are parrallel techniques, they are constructed for entirely differing purposes. One for respecting a specific symmetry in solutions present in a specific task, such as in many physics-related problems; whilst one removing a basis dependence which may arbitrarily and anisotropically affect the distribution of representations in all problems, giving it a motivation 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. In general, this shows that isotropy is a framework with wider and more flexible applicability, justifying the assertion of a universal default inductive bias, but 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, actually make the frameworks incompatible, seen through the restriction of pointwise nonlinearities in equivariant networks. This mutually-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 similaritiy in their fundamental construction from an equivariance relation and philsophical focus on symmetry principles. Equivariant networks are made to enforce an end-to-end symmetry deduced from its data structure, enforced by architecture modifications like group-convolution Cohen and Welling [2016a]. This greatly increases parameter efficiency and ensures physical solutions for specific problems. Isotropic deep learning is promoting the existing discrete rotational symmetry to a continuous one in localised functional forms, not neccessarily 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 major deviation from the foundational blueprint within Geometric deep learning , despite the use of equivariance and symmetries. They have substantial differences in how and which symmetries arise and how they affect the models, how activations are represented in networks and how parameters are constructed. Isotropic Deep Learning redefinines the form for parameter initialisation, rather than restructuring parameters, so not effect the layer structure which can remain dense.

Currently, the approaches are largely mutually-exclusive, though specific instances of isotropic deep learning can be made to express some characterising properties equivariant network — so they retain do have some very minimal overlap. Isotropic deep learning, more suitably falls within the class of geometries of representations, more closely related to work of Elhage et al. [2022], Olah et al. [2019], Carter et al. [2020]. This interdisciplinary approach, alongside work such as Elhage et al. [2022], may be more appropriately classified under representational geometry in deep learning.

C Initialisers, Normalisers, Regularisers and Optimisers

Do not wish to discourage anisotropic distributions if they are beneficial, afterall, the goal is to unconstrain the network and regularisers should reflect this. The scale factor α controls how many anisotropic-centres there are

gradient clipping Can imagine a dense thick layer hybridising the a

Include layer norm [unfinished]

D Stochastic Isotropy — Producing Immediate Anisotropic Analogs

One method to approximate isotropy with current functions is by stochastically choosing a basis for the anisotropic function to operate on. 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 privilidge 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 onto 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: $\mathbf{B} \sim \mathrm{SO}\,(n)$. 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 and many others . The existing matrix-multiplication procedure is computationally cumbersome, so 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 anologous functions converted from existing anisotropic ones.

For the example of standard dropout, shown in *Eqn.* 26, it can be made stochastically-isotropic by including the basis-transform shown in *Eqn.* 27. Where $\vec{x}$ is the activation vector, with normalisation factor S_a and S_{si}, dropout-mask $M_i = \vec{M} \cdot \hat{e}_i$ and standard-basis vectors $\hat{e}_i$.

$$\vec{x}' = S_a \sum_{i=1}^{N} M_i \left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i \tag{26}$$

$$\vec{x}' = S_{si} \sum_{i=1}^{N} \left(\mathbf{B}\vec{M} \cdot \hat{e}_i\right) \left(\vec{x} \cdot \hat{e}_i\right) \hat{e}_i \tag{27}$$

Similar formulations have been demonstrated, such as RotationOut by Hu and Poczos [2020], which 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 Givens [1958], which is not uniform over the space of special-orthogonal matrices — a neccessity for full stochastic-isotropy. Nevertheless, the implementation by Hu and Poczos [2020] is somewhat encouraging.

Overall, 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.

D.1 Considering Correllating the Stochastic-Isotropy

A curious extention, particularly to stochastically-isotropic dropout, would be to correllate 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: $\mathrm{SO}\,(n) \ni \mathbf{R}^{(\mathbf{t}+\mathbf{dt})} = \mathbf{R}^{(\mathbf{t})}\delta\mathbf{R}$, with $\delta\mathbf{R} = e^{\mathfrak{r}\cdot\vec{n}}$, with $\mathfrak{r}$ being the corresponding (normalised) anti-symmetric generators for rotations and $\vec{n} \sim \mathcal{N}(\vec{0}, \sigma \mathrm{I}_n)$ with $0 < \sigma \ll 1$. This procedure results in a random walk of the rotation matrix at each time step.

Following this, a time-correlated Bernoulli distribution can be defined. Beginning with $\vec{D}^{(0)}$, divide up the layer of neurons into two sets: inactive $I_n = \left\{ i | \vec{D}_i^{(n-1)} = 0 \right\}$ and active $A_n = \left\{ i | \vec{D}_i^{(n-1)} = 1 \right\}$. Then we have two hyper-parameters: the standard dropout probability λ and an overlap probability Γ, such that $|A_n| \, q + |I_n| \, \Gamma = (|A_n| + |I_n|) \, \lambda$ - where q is not a free parameter. If $|A_n| = 0$ or $|I_n| = 0$, then temporarily define $q = \Gamma = \lambda$. If not, then one needs to prevent unnormalised probabilities as shown in *Eqn.* 28.

$$\Gamma = \max\left(0, \max\left(\lambda + \frac{|A_n|}{|I_n|}\left(\lambda - 1\right), \min\left(1, \min\left(\lambda + \frac{|A_n|}{|I_n|}\lambda, \Gamma\right)\right)\right)\right) \tag{28}$$

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

This creates a link between the stimulus' presentation time to the network and the neurons its alters, such that stimuli presented in a smaller time window purturb similar subset of the network's neurons. This may produce an encoding similar to that found in human cognition, where neurons are thought to go through excitability cycles of slightly differing frequencies and phases. When the excitability is higher, information (engrams) preferentially encodes upon those neurons Yiu et al. [2014], Chen et al. [2020]. As groups of neurons begin to decohere, there remains some overlap, such that memories are interlaced if they occur within a temporal window of coherence. *This potentially gives neural networks using isotropic dropout an advantage when it comes to time-series data.*

E Taxonomy of Functional Forms

[unfinished]

F Potential Applications

Besides the proposed general applicability of the isotropic modifications

F.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 Vaswani et al. [2023], where isotropic-tanh may be of particular benefit, in replacing the softmax operation .

Softmax is defined through elements being bounded between zero and one, $\mathbf{f}\left(\vec{x}\right) \cdot \hat{e}_i \in [0, 1]$ 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 Elhage et al. [2022], particularly when stimuli do not coexist, so interference is minimised. Such a scenario may be a quantity which is continuous, but negative and positive values 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, but zero may be better represented as a neutral middle-point instead. In this case, the negative of a semantic direction may be equally meaningful. It may be expected that this is the case in the self-attention step.

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 an semantic correction to an activation in transformers even when it is innappropriate.**

The self-attention step comparese 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 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, $0 \leq \|\mathbf{f}\left(\vec{x}\right)\| \leq 1$, whilst elementwise its values are $-1 \leq \mathbf{f}\left(\vec{x}\right) \cdot \hat{e}_i \leq 1$. 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 a appealing drop-in-replacement for softmax in the attention step, at least conceptually. Its continuous rotational symmetry may also offer advantage, since the underlying outer-product of self attention $QK^T = \vec{x}^T W_Q^T W_K \vec{x} \hat{=} \vec{x}^T W'_{kq} \vec{x}$, is also isotropic in $\vec{x}$, then enabling more even interpolation between, and purturbation to, the values. Hence, an isotropic adaption to a self-attention may appear as shown in *Eqn.* 29, which will be explored in future work.

$$\text{Attention}\left(Q, K, V\right) = \text{Isotropic-Tanh}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{29}$$

However, this does not make transformers ‘isotropic’ as a whole, since further anisotropic steps exist.

F.2 Semantic Alignment through Embeddings

In much the same way, differing languages may produce comparable semantic maps when artificial basis anisotropies are removed from embeddings, disrupting alignment otherwise. Therefore previously unasigned terms may from their meanings and comparable semantic maps be ilucidated, particularly from an isotropic framework.