Learning from Protein Structure with Geometric Vector Perceptrons

Bowen Jing*
Department of Computer Science
Stanford University
bjing@stanford.edu

Stephan Eismann*
Department of Applied Physics
Stanford University
seismann@stanford.edu

Patricia Suriana
Department of Computer Science
Stanford University
psuriana@stanford.edu

Raphael J.L. Townshend
Department of Computer Science
Stanford University
raphael@cs.stanford.edu

Ron O. Dror
Department of Computer Science
Stanford University
rondror@cs.stanford.edu

Abstract

Learning on 3D structures of large biomolecules is emerging as a distinct area in machine learning, but there has yet to emerge a unifying network architecture that simultaneously leverages the geometric and relational aspects of the problem domain. To address this gap, we introduce *geometric vector perceptrons*, which replace standard dense layers to operate on collections of Euclidean vectors. Graph neural networks equipped with such layers are able to perform both geometric and relational reasoning on efficient representations of macromolecular structure. We demonstrate our approach on computational protein design and improve over state-of-the-art methods, achieving a perplexity of 5.29 on CATH 4.2.

1 Introduction

Many efforts in structural biology aim to predict, or derive insights from, the structure of a macromolecule (such as a protein, RNA, or DNA), represented as a set of positions associated with atoms or groups of atoms in 3D Euclidean space. These problems can often be framed as functions mapping the input domain of structures to some property of interest—for example, predicting the quality of a structural model or determining whether two molecules will bind. Thanks to their importance and difficulty, such problems, which we broadly refer to as *learning from structure*, have recently developed into an exciting and promising application area for deep learning [8, 10, 14, 15, 18].

Successful applications of deep learning are often driven by techniques that leverage the problem structure of the domain—for example, convolutions in computer vision [5] and attention in natural language processing [16]. What are the relevant considerations in the domain of learning from structure? Using proteins as the most common example, we have on the one hand the arrangement and orientation of the amino acids in space, which govern the dynamics and function of the molecule [4]. On the other hand, proteins also possess relational structure in terms of their amino-acid sequence

*Equal contribution

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

and the residue-residue interactions that mediate the aforementioned protein properties [9]. We refer to these as the *geometric* and *relational* aspects of the problem domain, respectively.

Recent state-of-the-art methods for learning from structure are successful by leveraging one of these two aspects [7, 13, 1, 19, 2, 10]. Commonly, such methods either employ graph neural networks (GNNs), which are expressive in terms of relational reasoning [3], or convolutional neural networks (CNNs), which operate directly on the geometry of the structure.

Here, we present a unifying architecture that bridges these two families of methods to leverage *both* aspects of the problem domain. We do so by introducing *geometric vector perceptrons* (GVPs), a drop-in replacement for standard multi-layer perceptrons (MLPs) in aggregation and feed-forward layers of GNNs. GVPs operate directly on both scalar and *geometric* features—features that transform as a vector under a rotation of spatial coordinates. GVPs therefore allow for the embedding of geometric information at nodes and edges without reducing such information to scalars that may not fully capture complex geometry. We postulate that our approach makes it easier for a GNN to learn functions whose significant features are both geometric and relational.

Our method (GVP-GNN) can be applied to any problem where the input domain is a structure of a single macromolecule or of molecules bound to one another. In this work, we specifically demonstrate our appproach on computational protein design (CPD): given a protein backbone structure, we attempt to infer an amino acid sequence which will fold into that structure.

2 Methods

Previous GNN architectures for learning from protein structure incorporate the 3D geometry of the protein by encoding *vector* features (such as node orientations and edge directions) in terms of rotation-invariant *scalars*, often by defining a local coordinate system at each node [10]. We instead propose that these features be *directly represented as geometric vectors*—features in a global $\mathbb{R}^3$ coordinate system—at all steps of graph propagation. This allows geometric features to be directly propagated without transforming between local coordinates, which we postulate enables the GNN to more easily access global geometric properties of the structure. The key challenge with such a representation, however, is to perform graph propagation in a way that simultaneously preserves the full expressive power of the original GNN while maintaining the rotation invariance provided by the scalar representations. We do so by introducing a new module, the *geometric vector perceptron*, to replace dense layers in a GNN.

Geometric Vector Perceptrons The geometric vector perceptron is a simple module for learning vector-valued and scalar-valued functions over geometric vectors and scalars. That is, given a tuple $(\mathbf{s}, \mathbf{V})$ of scalar features $\mathbf{s} \in \mathbb{R}^n$ and vector features $\mathbf{V} \in \mathbb{R}^{\nu \times 3}$, it computes new features $(\mathbf{s}', \mathbf{V}') \in \mathbb{R}^m \times \mathbb{R}^{\mu \times 3}$. The computation is formally described in Algorithm 1 and illustrated in Figure 1.

At its core, the GVP consists of two separate linear transformations $\mathbf{W}_m, \mathbf{W}_h$ for the scalar and vector features, followed by nonlinearities σ, σ^+.[2] However, before the scalar features are transformed, we concatenate the L_2 norm of the transformed vector features $\mathbf{V}_h$; this allows us to extract rotation-invariant information from the input vectors $\mathbf{V}$. An additional linear transformation $\mathbf{W}_\mu$ is inserted just before the vector nonlinearity to control the output dimensionality independently of the number of norms extracted.

The GVP is conceptually simple, yet provably possesses the desired properties of invariance/equivariance and expressiveness. First, the vector and scalar outputs of the GVP are equivariant and invariant, respectively, with respect to an arbitrary composition of rotations and reflections in 3D Euclidean space described by R *i.e.*,

$$\text{GVP}((\mathbf{s}, R(\mathbf{V}))) = (\mathbf{s}', R(\mathbf{V}')) \tag{1}$$

This is due to the fact that the only operations on vector-valued inputs are scalar multiplication, linear combination, and the L_2 norm.[3] We include a formal proof in the Appendix.

[2] We specifically choose ReLU and sigmoid, respectively.

[3] The nonlinearity σ^+ is a *scaling* by σ^+ applied to the L_2 norm.

In addition, a GVP can approximate *any* continuous rotation- and reflection-invariant scalar-valued function of $\mathbf{V}$. More precisely, let G_s be a GVP defined with $n, \mu = 0$—that is, the part of a GVP that transforms vector features to scalar features. Then G_s is able to ϵ-approximate a function $F : \mathbb{R}^{\nu \times 3} \to \mathbb{R}$ that is invariant with respect to rotations and reflections in 3D under mild assumptions.

Theorem. *Let R describe an arbitrary rotation or reflection in $\mathbb{R}^3$. For $\nu \geq 3$ let $\Omega^\nu \subset \mathbb{R}^{\nu \times 3}$ be the set of all $\mathbf{V} = [\mathbf{v}_1, \quad \ldots, \quad \mathbf{v}_\nu]^T \in \mathbb{R}^{\nu \times 3}$ such that $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are linearly independent and $0 < ||\mathbf{v}_i||_2 \leq b$ for all i and some finite $b > 0$. Then for any continuous $F : \Omega^\nu \to \mathbb{R}$ such that $F(R(\mathbf{V})) = F(\mathbf{V})$ and for any $\epsilon > 0$, there exists a form $f(\mathbf{V}) = \mathbf{w}^T G_s(\mathbf{V})$ such that $|F(\mathbf{V}) - f(\mathbf{V})| < \epsilon$ for all $\mathbf{V} \in \Omega^\nu$.*

We include a formal proof in the Appendix. As a corollary, a GVP with nonzero n, μ is also able to approximate similarly-defined functions over the full input domain $\mathbb{R}^n \times \mathbb{R}^{\nu \times 3}$.

Algorithm 1 Geometric vector perceptron

Input: Features $(\mathbf{s}, \mathbf{V}) \in \mathbb{R}^n \times \mathbb{R}^{\nu \times 3}$.
Output: Features $(\mathbf{s}', \mathbf{V}') \in \mathbb{R}^m \times \mathbb{R}^{\mu \times 3}$.
$h \leftarrow \max(\nu, \mu)$
GVP:
  $\mathbf{V}_h \leftarrow \mathbf{W}_h \mathbf{V} \quad \in \mathbb{R}^{h \times 3}$
  $\mathbf{V}_\mu \leftarrow \mathbf{W}_\mu \mathbf{V}_h \quad \in \mathbb{R}^{\mu \times 3}$
  $\mathbf{s}_h \leftarrow \|\mathbf{V}_h\|_2$ (row-wise) $\quad \in \mathbb{R}^h$
  $\mathbf{v}_\mu \leftarrow \|\mathbf{V}_\mu\|_2$ (row-wise) $\quad \in \mathbb{R}^\mu$
  $\mathbf{s}_{h+n} \leftarrow \text{concat}(\mathbf{s}_h, \mathbf{s}) \quad \in \mathbb{R}^{h+n}$
  $\mathbf{s}_m \leftarrow \mathbf{W}_m \mathbf{s}_{h+n} + \mathbf{b} \quad \in \mathbb{R}^m$
  $\mathbf{s}' \leftarrow \sigma(\mathbf{s}_m) \quad \in \mathbb{R}^m$
  $\mathbf{V}' \leftarrow \sigma^+(\mathbf{v}_\mu) \odot \mathbf{V}_\mu$, (row-wise multiply) $\quad \in \mathbb{R}^{\mu \times 3}$
return $(\mathbf{s}', \mathbf{V}')$

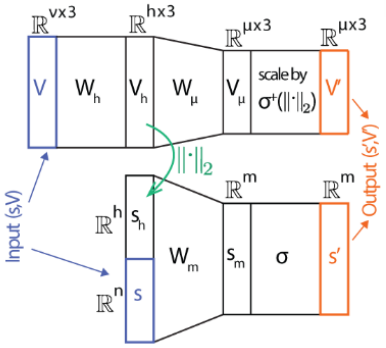

Figure 1: Schematic of the geometric vector perceptron.

Representations of Proteins A GVP-augmented GNN (GVP-GNN) can operate on and update graphs whose embeddings contain both vector and scalar features. Here we show how a 3D protein backbone structure can be efficiently represented in such a graph.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph where each node $\mathfrak{v}_i \in \mathcal{V}$ corresponds to an amino acid. Each node embedding $\mathbf{h}_\mathfrak{v}^{(i)}$ has vector features corresponding to the *forward* and *reverse* unit vectors in the directions of $\text{C}\alpha_{i+1} - \text{C}\alpha_i$ and $\text{C}\alpha_{i-1} - \text{C}\alpha_i$, respectively, and a unit vector in the imputed direction of $\text{C}\beta_i - \text{C}\alpha_i$.[4] These three vectors unambiguously define the absolute orientation of each amino acid residue. Each node embedding also has scalar features encoding the dihedral angles ϕ, ψ, ω.

Let the set of edges be $\mathcal{E} = \{\mathbf{e}_{i \to j}\}_{i \neq j}$ for all i, j where $\mathfrak{v}_i$ is among the $k = 30$ nearest neighbors of $\mathfrak{v}_j$ as measured by the distance between their Cα atoms. Each edge has an embedding $\mathbf{h}_e^{(i \to j)}$ with the unit vector in the direction of $\text{C}\alpha_i - \text{C}\alpha_j$, a radial basis encoding of the distance $\text{RBF}(||\text{C}\alpha_i - \text{C}\alpha_j||_2)$, and a sinusoidal encoding of $i - j$ (the distance along the sequence).

Collectively, these features are sufficient for a complete description of the protein backbone.

3 Experiments

Synthetic Tasks We first perform controlled experiments on a synthetic dataset to investigate if the GVP indeed improves the geometric and joint geometric-relational reasoning abilities of GNNs. This dataset mimics the essential qualities of the domain of protein structures: each synthetic "structure" consists of $n = 100$ random points in $\mathbb{R}^3$ in the ball of radius $r = 10$, each associated with a random unit vector in order to endow it with an orientation. Three random points in each structure are labelled as "special" to define the learning tasks.

[4] Cβ is the second carbon from the carboxyl carbon C.

We choose ground-truth labels in order to explicitly separate geometric and relational aspects in different tasks. In the "Off-center" task, the network predicts the distance from the centroid of the three special points to the centroid of the entire structure. In the "Perimeter" task, the network predicts the perimeter of the triangle defined by the three special points. We characterize the former as primarily geometric, as it requires reasoning about global properties of the 3D *shape*, and the latter as primarily relational, as it involves distances between three *pairs* of nodes. In the "Combined" task, the networks predicts the difference of the (normalized) off-center and perimeter objectives.

We compare a 3-layer CNN, a 3-layer standard GNN, and a GVP-GNN that is otherwise identical to the standard GNN on these three tasks. All models have the same intermediate dimensionality and training procedure; no hyperparameter tuning or architecture search is performed. For each of five randomly shuffled splits, we train three models of each type with different initialization seeds and take the ones with the best validation loss. As expected, the CNN significantly outperforms the standard GNN on the geometric task, and vice versa for the relational task. The GVP-GNN, however, simultaneously matches the CNN on the geometric task and the standard GNN on the relational task. Further, when we combine the two tasks in one objective, the GVP-GNN does significantly better than either the standard GNN or the CNN (Table 1). On the basis of these results, the GVP appears successful in combining the strengths of the CNN and GNN into a single architecture.

Table 1: Comparison of the CNN, standard GNN, and GVP-GNN on the three objectives on the synthetic test set. The MSE losses are standardized such that predicting a constant value (i.e. the mean) would result in unit loss. Results are reported as the mean $\pm$ S.D. over five randomly shuffled splits.

Model	Parameters	Off-center (geometric)	Perimeter (relational)	Combined
CNN	59k	0.319 ± 0.014	0.532 ± 0.028	0.522 ± 0.016
Standard GNN	40k	0.871 ± 0.045	0.128 ± 0.009	0.421 ± 0.025
GVP-GNN	22k	$\mathbf{0.206 \pm 0.024}$	$\mathbf{0.106 \pm 0.006}$	$\mathbf{0.155 \pm 0.024}$

Computational Protein Design Next, we evaluate GVP-GNN on the real-world task of computational protein design (CPD). This corresponds to learning amino acid sequences from protein backbone structures and can be formalized as learning a generative model for $p(\text{sequence} \mid \text{structure})$. Following Ingraham et al. [10], we frame this as an *autoregressive* task and use a masked encoder-decoder architecture to capture the joint distribution over all positions: for each position i, the network models the amino acid distribution at i based on the complete structure graph, as well as the sequence information at positions $j < i$.

We use the CATH 4.2 dataset curated by Ingraham et al. [10], in which all available structures with 40% nonredudancy are partitioned by their CATH (class, architecture, topology/fold, homologous superfamily) classification. Because CPD is difficult to ambiguously benchmark (some structures may correspond to many sequences and others to none at all), the proxy metric of *native sequence recovery*—splitting the set of all known native structures in the PDB and attempting to design sequences for held-out structures—is typically used [11, 12, 17]. Drawing an analogy between sequence design and language modelling, Ingraham et al. [10] also evaluate the model *perplexity* on held-out native sequences. We evaluate our model on both metrics.

Table 2: Performance on the CATH 4.2 test set and its short (< 100 amino acids) and single-chain subsets in terms of per-residue perplexity (lower is better) and sequence recovery (higher is better). Recovery is reported as the median over all structures of the mean recovery of 100 sequences per structure. The short, single-chain, and full test sets include 94, 103, and 1120 structures, respectively.

	Perplexity			Recovery %		
Method	Short	Single-chain	All	Short	Single-chain	All
GVP-GNN (ours)	**7.10**	**7.44**	**5.29**	**32.1**	**32.0**	**40.2**
Structured GNN	8.31	8.88	6.55	28.4	28.1	37.3
Structured Transformer	8.54	9.03	6.85	28.3	27.6	36.4

Our method achieves state-of-the-art performance on CATH 4.2. We substantially improve both in terms of perplexity and recovery over Structured Transformer [10], the previous state-of-the-art, as well as an improved variant of it (Structured GNN) in which the attention mechanism is replaced with standard graph propagation (Table 2). Additionally, our model achieves this performance with fewer parameters—1.01M compared to 1.38M / 1.53M in Structured GNN / Transformer.

4 Conclusion

In this work, we developed the first architecture designed specifically for learning on dual relational and geometric representations of 3D macromolecular structure. At its core, our method, GVP-GNN, augments graph neural networks with computationally simple layers that perform expressive geometric reasoning over Euclidean vector features. Our method possesses desirable theoretical properties and empirically outperforms existing architectures on learning sequence designs from protein structure. In further work, we hope to apply our architecture to other important structural biology problem areas, including protein complexes, RNA structure, and protein-ligand interactions.

Code

An implementation of the GVP will be made available at `https://github.com/drorlab/gvp`.

Acknowledgements

We thank Tri Dao and all members of the Dror group for feedback and discussions.

Funding

We acknowledge support from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program, and Intel Corporation. SE is supported by a Stanford Bio-X Bowes fellowship. RJLT is supported by the U.S. Department of Energy, Office of Science Graduate Student Research (SCGSR) program.

References

[1] N. Anand, R. R. Eguchi, A. Derry, R. B. Altman, and P. Huang. Protein sequence design with a learned potential. *bioRxiv*, 2020.

[2] F. Baldassarre, D. Menéndez Hurtado, A. Elofsson, and H. Azizpour. GraphQA: protein model quality assessment using graph convolutional networks. *Bioinformatics*, 2020.

[3] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[4] J. M. Berg, J. L. Tymoczko, and L. Stryer. Biochemistry, W.H. Freeman and Company, 2002.

[5] N. Cohen and A. Shashua. Inductive bias of deep convolutional networks through pooling geometry. In *International Conference on Learning Representations*, 2017.

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

[7] G. Derevyanko, S. Grudinin, Y. Bengio, and G. Lamoureux. Deep convolutional networks for quality assessment of protein folds. *Bioinformatics*, 34(23):4046–4053, 2018.

[8] J. Graves, J. Byerly, E. Priego, N. Makkapati, S. V. Parish, B. Medellin, and M. Berrondo. A review of deep learning methods for antibodies. *Antibodies*, 9(2):12, 2020.

[9] S. Hammes-Schiffer and S. J. Benkovic. Relating protein motion to catalysis. *Annu. Rev. Biochem.*, 75: 519–541, 2006.

[10] J. Ingraham, V. Garg, R. Barzilay, and T. Jaakkola. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, pages 15794–15805, 2019.

[11] Z. Li, Y. Yang, E. Faraggi, J. Zhan, and Y. Zhou. Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins: Structure, Function, and Bioinformatics*, 82(10):2565–2573, 2014.

[12] J. O'Connell, Z. Li, J. Hanson, R. Heffernan, J. Lyons, K. Paliwal, A. Dehzangi, Y. Yang, and Y. Zhou. Spin2: Predicting sequence profiles from protein structures using deep neural networks. *Proteins: Structure, Function, and Bioinformatics*, 86(6):629–633, 2018.

[13] G. Pagès, B. Charmettant, and S. Grudinin. Protein model quality assessment using 3d oriented convolutional neural networks. *Bioinformatics*, 35(18):3313–3319, 2019.

[14] J. C. Pereira, E. R. Caffarena, and C. N. dos Santos. Boosting docking-based virtual screening with deep learning. *Journal of chemical information and modeling*, 56(12):2495–2506, 2016.

[15] R. Townshend, R. Bedi, P. Suriana, and R. Dror. End-to-end learning on 3d protein structure for interface prediction. In *Advances in Neural Information Processing Systems*, pages 15616–15625, 2019.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[17] J. Wang, H. Cao, J. Z. Zhang, and Y. Qi. Computational protein design with deep learning neural networks. *Scientific reports*, 8(1):1–9, 2018.

[18] J. Won, M. Baek, B. Monastyrskyy, A. Kryshtafovych, and C. Seok. Assessment of protein model structure accuracy estimation in casp13: Challenges in the era of deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1351–1360, 2019.

[19] Y. Zhang, Y. Chen, C. Wang, C.-C. Lo, X. Liu, W. Wu, and J. Zhang. Prodconn: Protein design using a convolutional neural network. *Proteins: Structure, Function, and Bioinformatics*, 2019.

Properties of Geometric Vector Perceptrons

Equivariance and invariance

The vector and scalar outputs of the GVP are equivariant and invariant, respectively, with respect to an arbitrary composition of rotations and reflections in 3D Euclidean space described by R *i.e.*,

$$\text{GVP}((\mathbf{s}, R(\mathbf{V}))) = (\mathbf{s}', R(\mathbf{V}'))$$

Proof. We can write the transformation described by R as multiplying $\mathbf{V}$ with a unitary matrix $\mathbf{U} \in \mathbb{R}^{3\times3}$ from the right. The L_2-norm, scalar multiplications, and nonlinearities are defined row-wise as in Algorithm 1. We consider scalar and vector outputs separately. The scalar output, as a function of the inputs, is

$$\mathbf{s}' = \sigma\left(\mathbf{W}_m \begin{bmatrix} \|\mathbf{W}_h\mathbf{V}\|_2 \\ \mathbf{s} \end{bmatrix} + \mathbf{b}\right)$$

Since $\|\mathbf{W}_h\mathbf{V}\mathbf{U}\|_2 = \|\mathbf{W}_h\mathbf{V}\|_2$, we conclude $\mathbf{s}'$ is invariant. Similarly the vector output is

$$\mathbf{V}' = \sigma^+\left(\|\mathbf{W}_\mu\mathbf{W}_h\mathbf{V}\|_2\right) \odot \mathbf{W}_\mu\mathbf{W}_h\mathbf{V}$$

The row-wise scaling can also be viewed as left-multiplication by a diagonal matrix $\mathbf{D}$. Since $\|\mathbf{W}_\mu\mathbf{W}_h\mathbf{V}\|_2 = \|\mathbf{W}_\mu\mathbf{W}_h\mathbf{V}\mathbf{U}\|_2$, $\mathbf{D}$ is invariant. Since

$$\mathbf{D}\mathbf{W}_\mu\mathbf{W}_h(\mathbf{V}\mathbf{U}) = (\mathbf{D}\mathbf{W}_\mu\mathbf{W}_h\mathbf{V})\,\mathbf{U}$$

we conclude that $\mathbf{V}'$ is equivariant. □

The ability to approximate rotation-invariant functions

The GVP inherits an analogue of the Universal Approximation property [6] of standard dense layers. If R describes an arbitrary rotation or reflection in 3D Euclidean space,[5] we show that a GVP can approximate arbitrary scalar-valued functions invariant under R and defined over $\Omega^\nu \subset \mathbb{R}^{\nu\times 3}$, the bounded subset of $\mathbb{R}^{\nu\times 3}$ whose elements can be canonically oriented based on three linearly independent vector entries. Without loss of generality, we assume the first three vector entries can be used.

The machinery corresponding to such approximations corresponds to a GVP G_s with only vector inputs, only scalar outputs, and a sigmoidal nonlinearity σ; followed by a dense layer. This can also be viewed as the sequence of matrix multiplication with $\mathbf{W}_h$, taking the L_2-norm, and a dense network with one hidden layer. Such machinery can be extracted from any two consecutive GVPs (assuming a sigmoidal σ).

Theorem. *Let R describe an arbitrary rotation or reflection in $\mathbb{R}^3$. For $\nu \geq 3$ let $\Omega^\nu \subset \mathbb{R}^{\nu\times 3}$ be the set of all $\mathbf{V} = [\mathbf{v}_1, \quad \ldots, \quad \mathbf{v}_\nu]^T \in \mathbb{R}^{\nu\times 3}$ such that $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are linearly independent and $0 < \|\mathbf{v}_i\|_2 \leq b$ for all i and some finite $b > 0$. Then for any continuous $F : \Omega^\nu \to \mathbb{R}$ such that $F(R(\mathbf{V})) = F(\mathbf{V})$ and for any $\epsilon > 0$, there exists a form $f(\mathbf{V}) = \mathbf{w}^T G_s(\mathbf{V})$ such that $|F(\mathbf{V}) - f(\mathbf{V})| < \epsilon$ for all $\mathbf{V} \in \Omega$.*

Proof. The idea is to write F as a composition $F = \tilde{F} \circ \omega$ and $\omega = h \circ y$. We show that multiplication with $\mathbf{W}_h$ and and taking the L_2-norm can compute y, and that the dense network with one hidden layer can approximate $\tilde{F} \circ h$.

Call an element $\mathbf{V} \in \Omega^\nu$ *oriented* if $\mathbf{v}_1 = x_1\mathbf{e}_x$, $\mathbf{v}_2 = x_2\mathbf{e}_x + y_2\mathbf{e}_y$, and $\mathbf{v}_3 = x_3\mathbf{e}_x + y_3\mathbf{e}_y + z_3\mathbf{e}_z$, with $x_1, y_2, z_3 > 0$. Define $\omega : \Omega^\nu \to \mathbb{R}^{3\nu-3}$ to be the *orientation function* that orients its input and then extracts the vector of $3\nu - 3$ coefficients, $[x_1, x_2, y_2, x_3, y_3, z_3, \ldots, x_i, y_i, z_i, \ldots]^T$. These elements can be written as

$$\begin{aligned}
x_1 &= \|\mathbf{v}_1\|_2 \\
x_i &= \mathbf{v}_i \cdot \mathbf{v}_1 / x_1, \quad i \geq 2 \\
y_2 &= \sqrt{\|\mathbf{v}_2\|_2^2 - x_2^2} \\
y_i &= (\mathbf{v}_i \cdot \mathbf{v}_2 - x_i x_2)/y_2, \quad i \geq 3 \\
z_3 &= \sqrt{\|\mathbf{v}_3\|_2^2 - x_3^2 - y_3^2} \\
z_i &= (\mathbf{v}_i \cdot \mathbf{v}_3 - x_i x_3 - y_i y_3)/z_3, \quad i \geq 4
\end{aligned}$$

and are invariant under rotation and reflection, because they are defined using only the norms and inner products of the $\mathbf{v}_i$. Then $F = \tilde{F} \circ \omega$, where $\tilde{F} : [-b, b]^{3\nu-3} \to \mathbb{R}$.

The key insight is that if we construct $\mathbf{W}_h$ such that the rows of $\mathbf{W}_h\mathbf{V}$ are the original vectors $\mathbf{v}_i, \forall i$ and all differences $\mathbf{v}_i - \mathbf{v}_j, \forall i, j \leq \min(i, 3)$, then we can compute $\omega(\mathbf{V})$ from the row-wise norms of $\mathbf{W}_h\mathbf{V}$. That is, $\omega = h \circ y$ where $\mathbf{y} = y(\mathbf{V}) = \|\cdot\|_2 \odot (\mathbf{W}_h\mathbf{V}) \in \mathbb{R}^{4\nu-6}$ and h is an application of the cosine law. The GVP precisely computes $\mathbf{y}$ as an intermediate step: $G_s(\mathbf{V}) = \sigma \odot (\mathbf{W}_m\mathbf{y} + \mathbf{b})$. It remains to show that there exists a form $\tilde{f}(\mathbf{y}) = \mathbf{w}^T[\sigma \odot (\mathbf{W}_h\mathbf{y} + \mathbf{b})]$ that ϵ-approximates $\tilde{F} \circ h : [-2b, 2b]^{4\nu-6} \to \mathbb{R}$. Up to a translation and uniform scaling of the hypercube, this is the result of the Universal Approximation Theorem [6]. □

[5] More precisely, if R describes a unitary transformation.