ICML
International Conference
On Machine Learning

VDW-GNNs

Vector diffusion wavelets for geometric graph neural networks

**David R Johnson, Alexander Sietsema, Rishabh Anand, Deanna Needell,
Smita Krishnaswamy, Michael Perlmutter**

Vector-valued signals

- Graph data is typically a collection of **scalar-valued** node signals, $\mathbf{x}_f : V \to \mathbb{R}$, $1 \leq f \leq F_{\text{scalar}}$.
- However, **vector-valued** signals, $\mathbf{w}_f : V \to \mathbb{R}^d$, $1 \leq f \leq F_{\text{vector}}$, can have important geometric/physical meaning.
  - Positions, velocities, forces, flows, etc.

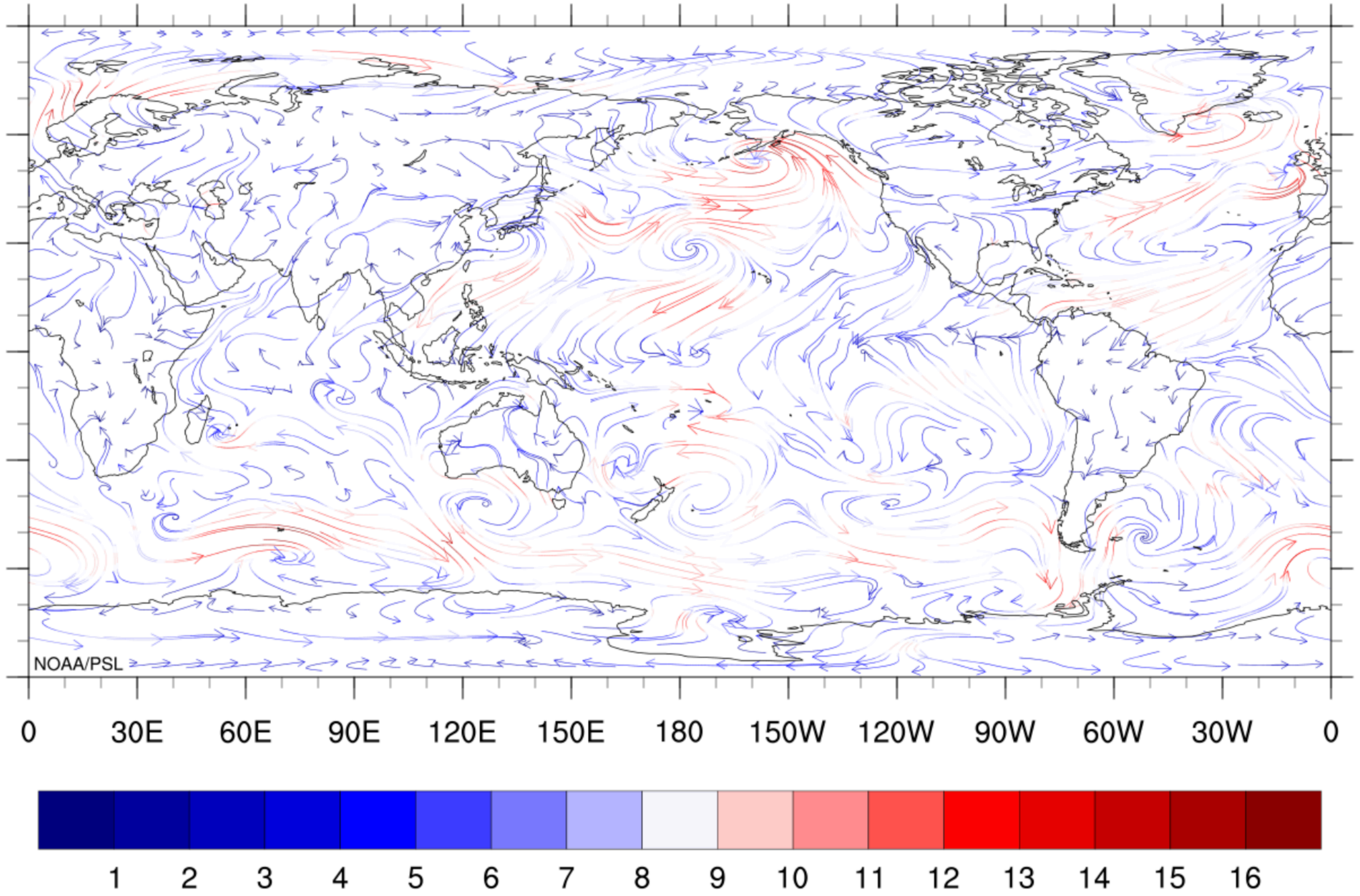

Image source: NOAA Physical Sciences Laboratory.
https://psl.noaa.gov/data/gridded/data.ncep.reanalysis.html

Diffusion wavelets

- Diffusion Wavelets (Coifman and Maggioni 2006) are constructed using powers of the **lazy random-walk diffusion operator** $P = \frac{1}{2}(I + D^{-1}A)$:

$$\Psi_j(P) = P^{2^{j-1}} - P^{2^j}, \; 1 \leq j \leq J.$$

- Increasing powers of P capture smoother / more global signal (via more mixing in the random walk, which approximates the heat kernel)

- Wavelets are bandpass filters; a filter bank of complementary wavelets $\{\Psi_j\}_{j=1}^{J}$ extracts a multi-scale featurization of a signal on a graph

- Applying wavelets built from P is computationally efficient, esp. if A is sparse

Diffusion Wavelets for vectors?

- As an efficient, multi scale feature extractor, we want to be able to use Diffusion Wavelets on vector features, in addition to scalars.
- However, curvature (on a geometric graph or manifold) induces local coordinate misalignment for vector-valued features.
- Therefore, vectors cannot be directly aggregated across locally misaligned neighborhoods.
- **How can we adapt Diffusion Wavelets, to work with vectors?**

Vector Diffusion Wavelets

Building off of Vector Diffusion Maps (Singer and Wu, 2012), we start with 'local PCA' to estimate a basis for the tangent space at each point:

1. After local re-centering and re-scaling neighbor coordinates, we obtain and decompose $\mathbf{B}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^\top \in \mathbb{R}^{d\times n_i}$, where columns of $\mathbf{U}_i$ contain singular vectors for point i (→ basis of local max variation).
2. For neighbors i and j, $\mathcal{O}_{i,j} = \mathbf{U}_i \mathbf{U}_j^\top \in \mathbb{R}^{d\times d}$ rotates a vector from j's tangent frame into i's tangent frame.
3. We construct a diffusion operator $\mathbf{Q} \in \mathbb{R}^{nd\times nd}$ for an entire graph from $\mathcal{O}_{i,j}$ blocks weighted by the entries of $\mathbf{P}$ (the lazy random walk operator):
$$\mathbf{Q}[i,j] = \mathbf{P}[i,j]\mathcal{O}_{i,j} \in \mathbb{R}^{d\times d}\,.$$
4. VDWs that diffuse (stacked) vectors $\mathbf{w} \in \mathbb{R}^{nd}$ between local frames are:
$$\mathbf{\Psi}_j = (\mathbf{Q}^{t_{j-1}} - \mathbf{Q}^{t_j})\mathbf{w} \in \mathbb{R}^{nd}\,.$$

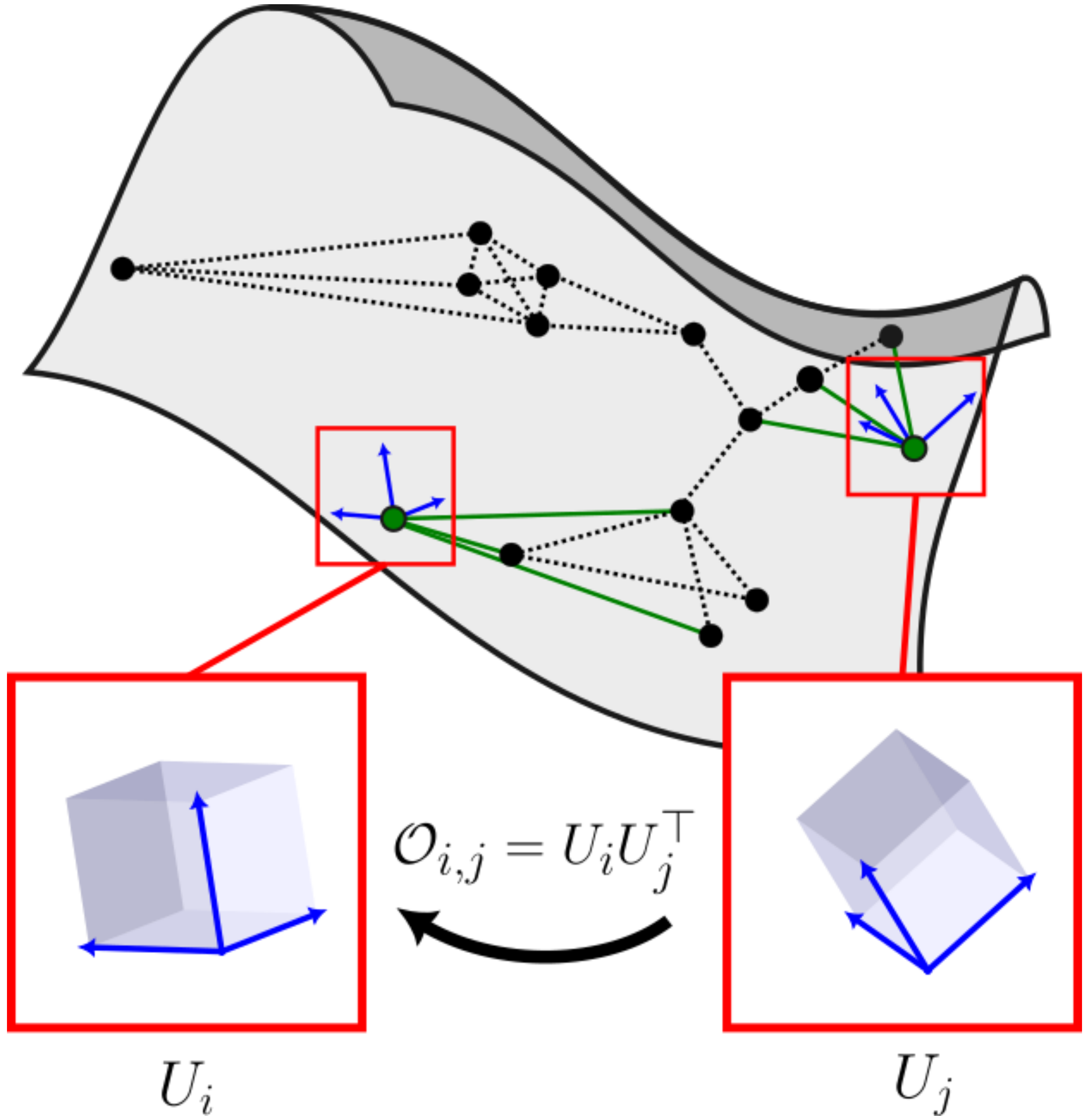

Image source: authors

SO(*d*)-equivariance of VDWs

Theorem: VDWs built from $\mathbf{Q}$ are **equivariant to rotations**, that is:

$$\mathbf{\Psi}_j = \mathbf{Q}^{t_{j-1}} - \mathbf{Q}^{t_j},$$

$$\widetilde{\mathbf{\Psi}}_j(\mathbf{R} \cdot \mathbf{w}) = \mathbf{R} \cdot \mathbf{\Psi}_j \mathbf{w}.$$

Translation invariance is easily automatically handled by the centering step in the local PCA.

→ These properties ensure that vector-valued message passing via VDWs remains consistent under arbitrary/inconsistent rotations to the data, which can help learning efficiency.

Experiment - multi-channel neural recordings

- We adapt a data set from Gosztolai et al. (2025), consisting of 44 sessions of **24-channel neural recordings** where a trained monkey moves its arm in one of seven directions after a cue.

- The task is to create easy-to-classify **3D embeddings** from the original 24D time series data.

- Like Gosztolai et al. (2025), we numerically differentiate the data to obtain ‘**neural velocities**’ (vector-valued features), **assumed to lie in the tangent bundle of a manifold**.

- Models are trained separately on each session, and evaluated on the overall classification performance of an SVM trained on the embeddings.

Results - multi-channel neural recordings

Results on the multi-channel neural recordings data set across five-fold CV on each of 44 sessions.

Model	**Classifier accuracy ↑**	**Sec. per epoch**	**Best epoch**	**Parameter count**
VDW-GNN	**0.66 ± 0.14**	0.23 ± 0.05	73 ± 172	164,868
MARBLE	0.65 ± 0.12	1.41 ± 0.63	20 ± 57	1,443,004
CEBRA	0.57 ± 0.09	0.40 ± 0.04	685 ± 3040	11,171
LFADS	0.35 ± 0.08	0.44 ± 0.11	5 ± 34	1,068,800