A. Notations

We provide a lookup table for notations used in our work.

Table 6. Description of notations used in our work.

Notation	Description
$\mathcal{G}$	undirected, flow-driven spatial network (or graph)
$\mathcal{V}$	set of nodes in $\mathcal{G}$
n_i	node i in $\mathcal{G}$
$\mathcal{E}$	set of edges in $\mathcal{G}$
e_{ij}	edge (or link) in $\mathcal{G}$ between n_i and n_j
$\overline{e_{ij}}$	average edge length estimated over edges in $\mathcal{G}$
σ	standard deviation of edge length estimated over edges in $\mathcal{G}$
t	link prediction target
e_{ij}^t	target link (or edge) between n_i^t and n_j^t
n_i^t	target node i affiliated to e_{ij}^t
h	number of hops
$\mathcal{G}_h^t$	h-hop subgraph extracted around e_{ij}^t
$\mathcal{L}(\mathcal{G}_h^t)$	line graph representation of $\mathcal{G}_h^t$
$\mathcal{V}'$	set of nodes in $\mathcal{L}(\mathcal{G}_h^t)$
n_i'	node (or vector embedding) i in $\mathcal{L}(\mathcal{G}_h^t)$
$\mathcal{E}'$	set of edges in $\mathcal{L}(\mathcal{G}_h^t)$
e_{ij}'	edge (or link) in $\mathcal{L}(\mathcal{G}_h^t)$ between n_i' and n_j'
k	number of message-passing iterations
s_i	scalar value generated in GAV layer
$\lvert s_i \rvert$	absolute value of s_i
$\tilde{n}_i'$	intermediate node representation (or vector embedding) in GAV layer
$\hat{n}_i'$	updated, refined node representation (or vector embedding)
Q	query sequence in multi-head attention operation
K	key sequence in multi-head attention operation
V	value sequence in multi-head attention operation
$\hat{y}_{ij}^t$	GAV's predicted probability of existence of e_{ij}^t
y_{ij}^t	ground truth label of existence of e_{ij}^t
$\mathcal{E}_{\mathcal{N}(n_i^t)}$	set of refined vector embeddings originally created from edges adjacent to n_i^t
N_i	matrix consisting of n_i' and its direct neighbors
$\mathcal{N}(n_i)$	set of nodes in the direct neighborhood of n_i
$\mathcal{N}(n_i) \cup n_i$	set of nodes in the direct neighborhood of n_i including n_i itself
$\phi_\theta^{(1)}$, $\phi_\theta^{(2)}$	learnable functions in GAV layer
$\phi_\theta^{(3)}$	learnable function in readout module
d_{spatial}	spatial dimension (2 or 3)
d_{message}	dimension of $\tilde{n}_i'$ in GAV layer
δ	maximum distance threshold utilized in spatial sampling during preprocessing
$\|$	concatenation operation
$\mathcal{L}_{\text{BCE}}$	binary cross-entropy loss function

B. More on Interpretability and the Modification of Vector Embeddings

We provide the interested reader with more visualizations regarding the GAV layer's modification of vector embeddings (see Section 3.2) on the validation set of the ogbl-vessel benchmark, similar to Fig. 5. These visualizations can also be interpreted as qualitative results. Fig. 6 depicts subgraph representations $\mathcal{G}_h^t$ (h set to one) of 12 positive (real, plausible) target links, while Fig. 7 depicts subgraph representations of 12 negative (sampled, implausible) target links. Please note that the respective last rows depict challenging cases, as indicated by GAV's predicted probabilities $\hat{y}_{ij}^t$. Additionally, we would like to highlight our hypothesis from Section 4.5 that GAV may attempt to assign the two target nodes (red and green) to sink and source nodes for negative, implausible vessel formations (see Fig. 7), which results in superior representations for link prediction in flow-driven spatial networks that can be effortlessly classified in our physically plausible readout module. Please note that we conduct an additional experiment modifying a toy example in Sec. D to further facilitate interpretability.

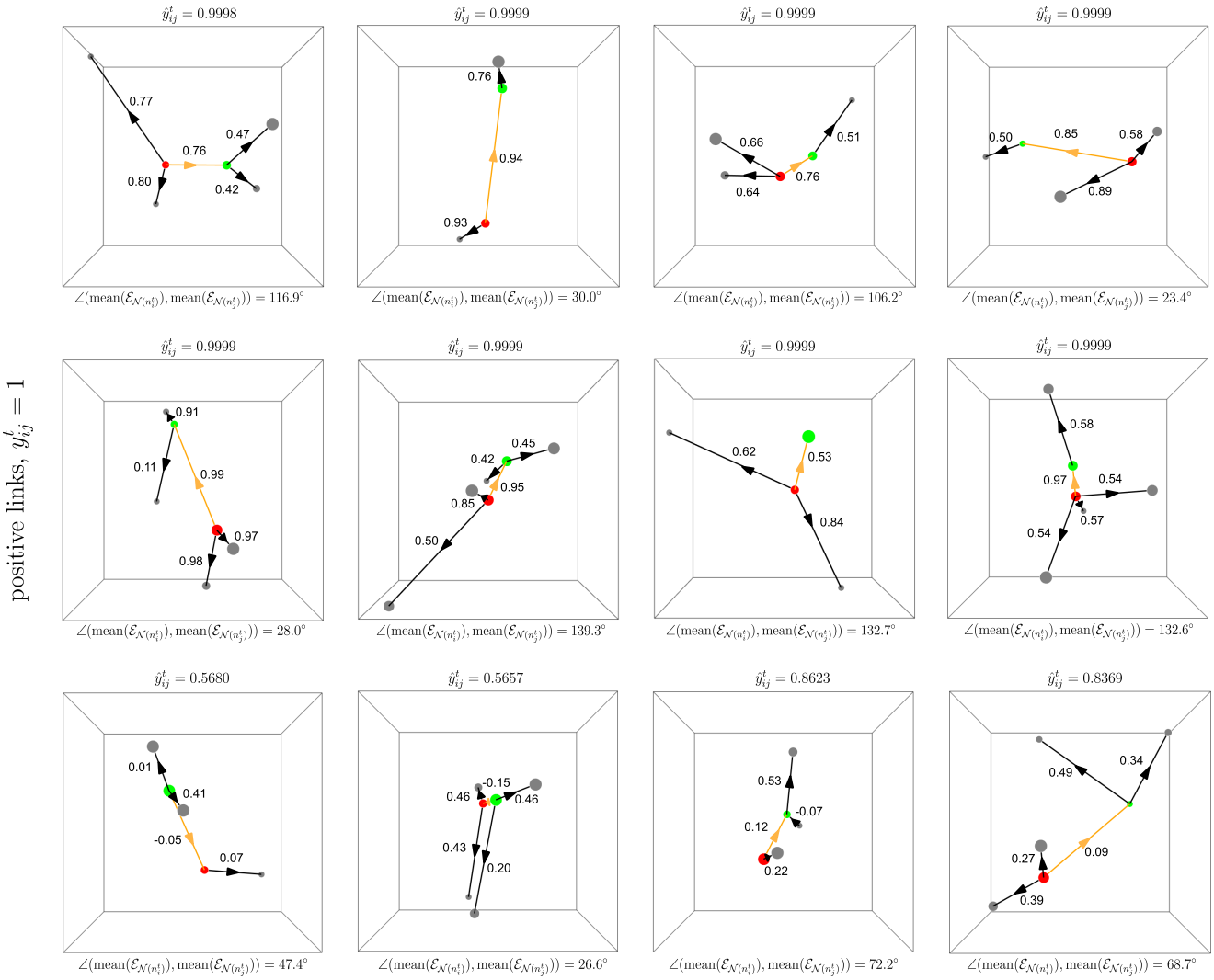

Figure 6. Visualization of the effect of our GAV layer on vector embeddings. We visualize subgraph representations $\mathcal{G}_h^t$ of 12 positive target links ($y_{ij}^t = 1$) together with the GAV layer's predicted scalar values $s_i \in (-1, 1)$. The scalar values s_i used to update vector embeddings in $\mathcal{L}(\mathcal{G}_h^t)$ have been projected to their corresponding edges in $\mathcal{G}_h^t$ (see Fig. 2) to provide an interpretable visualization. The directionality of edges (indicated by arrows) already incorporates potential shifts in the direction of vector embeddings enforced by our GAV layer. We additionally report the angle $\angle$ between the vector embeddings aggregated around the two target nodes (see Section 3.4) and the predicted probability of link existence $\hat{y}_{ij}^t$.

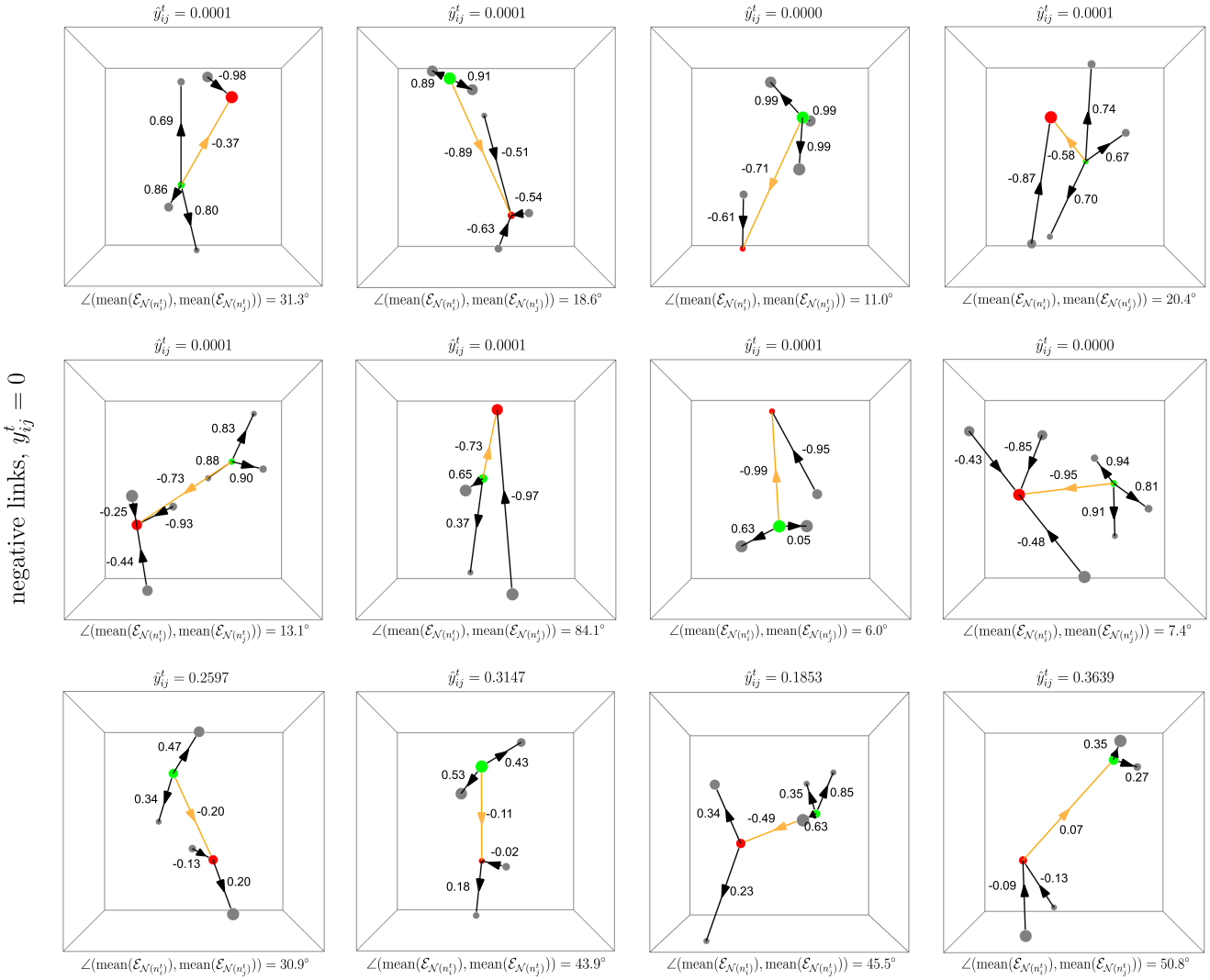

Figure 7. Visualization of the effect of our GAV layer on vector embeddings. We visualize subgraph representations $\mathcal{G}_h^t$ of 12 negative target links ($y_{ij}^t = 0$) together with the GAV layer's predicted scalar values $s_i \in (-1, 1)$. The scalar values s_i used to update vector embeddings in $\mathcal{L}(\mathcal{G}_h^t)$ have been projected to their corresponding edges in $\mathcal{G}_h^t$ (see Fig. 2) to provide an interpretable visualization. The directionality of edges (indicated by arrows) already incorporates potential shifts in the direction of vector embeddings enforced by our GAV layer. We additionally report the angle $\angle$ between the vector embeddings aggregated around the two target nodes (see Section 3.4) and the predicted probability of link existence $\hat{y}_{ij}^t$.

C. Initialization of Vector Embeddings

The initialization of the direction of vector embeddings represents an important implementation detail and is based on a straightforward intuition. To be precise, we initialize vector embeddings to point away from the target link e_{ij}^t, *i.e.*, towards nodes with a node degree of one (leaf nodes). The vector embedding representative of the target link e_{ij}^t is set to point from n_i^t to n_j^t. An exemplary initialization of vector embeddings for a 1-hop subgraph can be found in Fig. 2.

D. GAV and Structural Properties

This section elaborates on how GAV's predictions rely heavily on structural properties, such as bifurcation angles, which reflect functional properties of the underlying physical system [10]. To this end, we prepare and modify a synthetic mock example in Fig. 8. Specifically, we vary the bifurcation angle ψ_b spanned between two edges connected to n_i^t (red) to generate morphological implausible and plausible blood vessel formations (see Fig. 8).

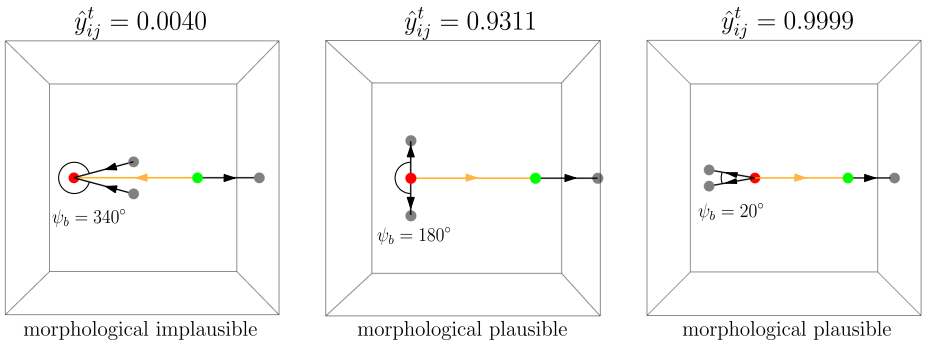

Figure 8. Morphological implausible (left) and plausible (middle and right) blood vessel formations formed around the target link (orange) with varying bifurcation angles ψ_b. GAV correctly identifies morphological plausible blood vessel formations that fulfill relevant hemodynamic functional properties [10].

As expected, GAV differentiates between plausible and implausible blood vessel formations formed around the target link. GAV assigns a high probability of target link existence $\hat{y}_{ij}^t$ to plausible and a low probability of target link existence to implausible formations. Please note that Fig. 8 additionally maps the potentially modified directionality of vector embeddings onto their corresponding edges, similar to Fig. 6 and Fig. 7. One can observe the shift in the directionality of vector embeddings created from edges adjacent to nodes with high bifurcation angles ψ_b, transforming the target nodes (red and green) to sink and source nodes for morphological implausible blood vessel formations (see Fig. 8, left).

E. Visualization of Datasets

In Fig. 9, we graphically visualize two flow-driven spatial networks representative of murine whole-brain vessel graphs and road networks.

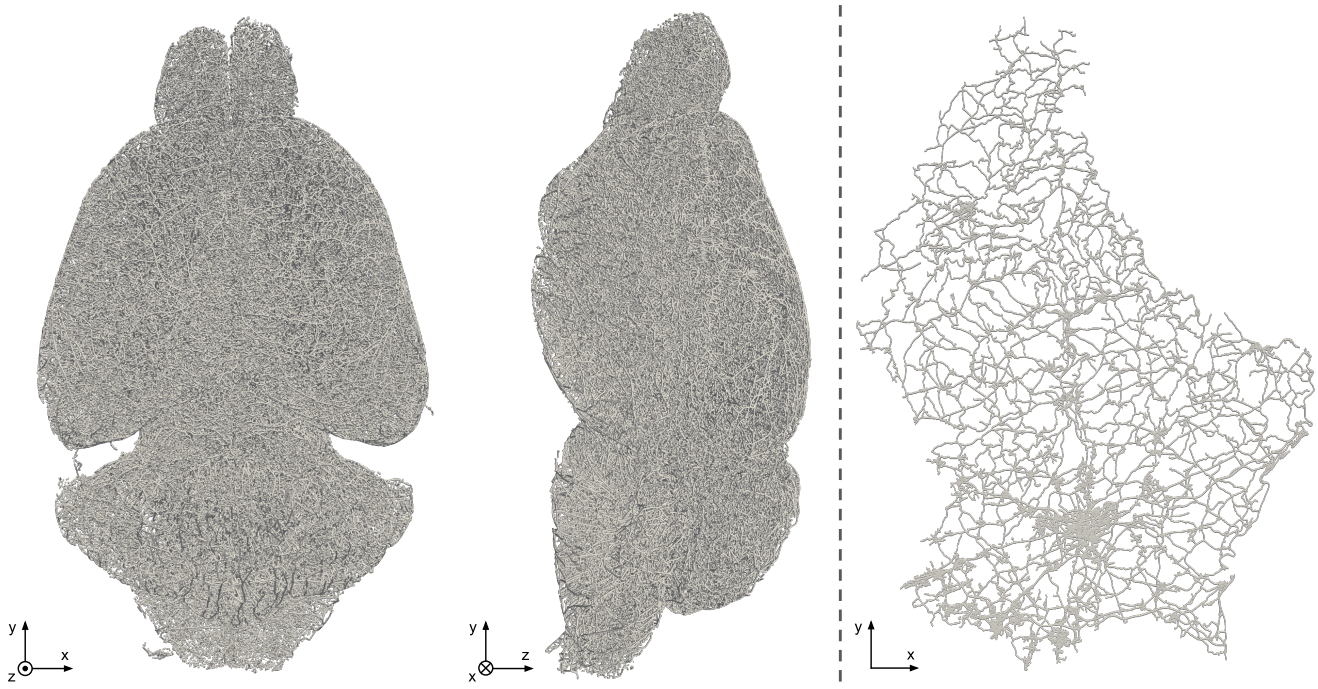

Figure 9. Visualization of a whole-brain vessel graph and a road network. The depicted flow-driven spatial networks correspond to the raw cd1-tc-vessel and luxembourg-road datasets.

F. More Ablations on the GAV Layer

Since the GAV layer relies on a set of specific design choices, we conduct additional ablation studies determining their influence on the link prediction performance on the validation set of the ogbl-vessel benchmark. To this end, we experiment with different versions of the GAV layer. First, we deactivate the multi-head attention operation; second, we exclude the residual connection; and third, we exchange the leaky ReLU non-linearity in the GAV layer with the ReLU non-linearity. We report our findings in Table 7.

Table 7. Ablations on the GAV layer's main design choices.

Attention	Residual Connection	Leaky ReLU	AUC ↑	Hits@100 ↑	Hits@50 ↑	Hits@20 ↑
✓	✓	✓	98.39	34.46	26.30	19.81
✗	✓	✓	97.48	15.56	9.18	5.68
✓	✗	✓	98.34	34.90	27.61	19.28
✓	✓	✗	98.34	34.16	26.47	17.31

Deactivating the multi-head attention operation (second row) results in a drastic AUC decrease of 0.91, indicating the importance of neighborhood awareness when modifying vector embeddings via our proposed GAV layer. Excluding the GAV layer's residual connection (third row) and using ReLU instead of Leaky ReLU non-linearities (fourth row) leads to a slight reduction in AUC of 0.05, respectively. Based on our reported standard deviation value of $\pm$ 0.02 (see Table 2), we argue that this performance decrease is indeed significant.

G. Evaluation Metrics

To compare GAV to existing baseline algorithms, we report quantitative results based on the area under the receiver operating characteristic curve (AUC), following the obgl-vessel benchmark. The AUC metric indicates the performance of a classifier by plotting the true positive rate against the false positive rate at all possible classification thresholds. Therefore, AUC provides an aggregate performance measure indicating the classifier's ability to distinguish between positive and negative links.

We introduce the evaluation metric Hits@k as an additional, stricter performance measure. Hits@k compares the classifier's prediction of every single positive link to a randomly sampled set of 100,000 negative links, resulting in a ranking among 100,001 links with respect to the probability of link existence. Based on this ranking, Hits@k indicates the ratio of positive links ranked at the k-th place and above. In the context of this work, we evaluate Hits@k at $k = 100$, $k = 50$, and $k = 20$, inspired by other Open Graph Benchmark [4] link prediction benchmarks.

H. Configuration of Our Secondary Baseline

We incorporate the EdgeConv message-passing layer [16] into the SEAL framework [17, 19], which has been shown to deliver results on par with or superior to the state-of-the-art on multiple link prediction benchmarks, to introduce a strong, *secondary baseline* (SEAL+EdgeConv) for link prediction in spatial networks. To be precise, we incorporate EdgeConv in SEAL's DGCNN [18]. EdgeConv's update function can be observed in Table 10. Here, ϕ_θ represents a two-layer MLP with an input dimension of 64, a hidden dimension of 32, and an output dimension of 32. Our modified DGCNN employs in total three EdgeConv layers, with the only difference being that the input dimension of the first EdgeConv layer's MLP corresponds to 70 and the output dimension of the third EdgeConv layer's MLP to 1. Our EdgeConv version utilizes a mean feature aggregation scheme. We set the number of in- and output channels of the DGCNN readout operation's two 1D convolutions to 1 & 16 and 16 & 32, respectively. The kernel sizes and strides of the two 1D convolutions correspond to 65 & 65 and 5 & 1. We set the input, hidden, and output dimensions of the DGCNN readout operation's MLP to 38, 128, and 1. The global sort pooling layer's parameter k is set to 10. All hyperparameters were tuned on the validation set of the ogbl-vessel benchmark.

I. GAV's Performance on Non-Flow-Based Link Prediction Benchmarks

To additionally confirm that GAV is specialized for link prediction in flow-driven spatial networks, we conduct an experiment on the ogbl-collab benchmark [4], which represents a collaboration network given by an undirected graph where nodes are associated with authors while edges indicate collaborations between them. Node features are comprised of 128-dimensional vectors representative of an author's scientific work (averaged word embeddings reflecting the content of scientific papers). Based on the collaboration network, the task is to predict future collaborations between authors. To adjust

GAV to the task of the ogbl-collab benchmark, we model vector embeddings representative of edges in the collaboration network as the difference between 128-dimensional feature vectors of two nodes incident to an edge. We report our findings in Table 8.

Table 8. Comparison between GAV and SEAL on ogbl-collab and ogbl-vessel. Please note that the increase in GAV's trainable parameters in the experiment on ogbl-collab is mostly due to the increased number of node features (128 vs. 3).

Dataset	Model	# Params ↓	Eval. Metric ↑ (%)
ogbl-collab	SEAL [19]	501,570	64.72 Hits@50
	GAV (ours)	44,194	16.72 Hits@50
ogbl-vessel	SEAL [19]	172,610	80.50 AUC
	GAV (ours)	8,194	98.38 AUC

As expected, GAV, relying on the idea of modeling simplified physical flow in flow-driven spatial networks, does not deliver competitive results on the ogbl-collab benchmark. This is because GAV's strong inductive biases are tailored to link prediction in flow-driven spatial networks and are, therefore, too restrictive for non-flow-based networks, such as ogbl-collab. This repeatedly demonstrates GAV's ability to intuitively model the underlying physical process in flow-driven spatial networks.

To adapt GAV to non-flow-based networks more appropriately, we encourage future work to explore the use of pseudo-spatial positions embedded in nodes of non-flow-based networks rather than its actual node features for the sake of creating vector embeddings. Pseudo-spatial position could, *e.g.*, be determined based on the Fruchterman-Reingold force-directed algorithm [6].

J. On Translation and Rotation Invariance

In this section, we briefly investigate GAV's behavior under rotations and translations of the h-hop enclosing subgraph $\mathcal{G}_h^t$. Specifically, we aim to investigate whether rotation and translation of $\mathcal{G}_h^t$ result in similar predictions. Since GAV encodes edges as vector embeddings spanned between two nodes (see Section 3.1), translation invariance is *explicitly ensured*. However, even though rotation preserves the length and relative angles of edges, rotation invariance is *not explicitly ensured*. This is, *e.g.*, because queries and keys forwarded to the GAV layer's attention operation are not explicitly rotation equi- or invariant, which is one of the key requirements for rotation invariant attention weights and hence potential rotation invariant predictions [2]. An empirical experiment rotating exemplary input graphs around all three axes, however, demonstrates that GAV's predictions are relatively robust to rotation, indicating to some degree implicit, learned rotation invariance (see Table 9). We encourage future work to further explore the necessity of explicitly encoded translation and rotation invariance in the context of link prediction for flow-driven spatial networks.

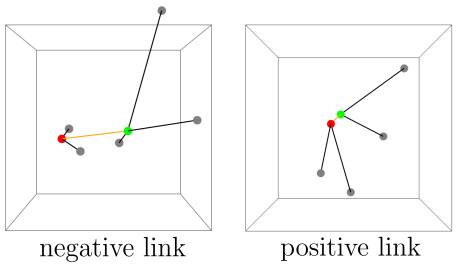

Figure 10. Exemplary $\mathcal{G}_h^t$ extracted from the ogbl-vessel benchmark around a negative and positive link used for experiments in Table 9.

Table 9. Experiment on rotation invariance of predictions. We rotate three exemplary subgraphs $\mathcal{G}_h^t$ around all three axes with a step size of 1° to investigate GAV's behavior under rotation. We report the standard deviation of predicted target link probability $\hat{y}_{ij}^t$ over all 360 predictions.

Subgraph $\mathcal{G}_h^t$	$\sigma_{\hat{y}_{ij}^t}$ (x-axis)	$\sigma_{\hat{y}_{ij}^t}$ (y-axis)	$\sigma_{\hat{y}_{ij}^t}$ (z-axis)
Fig. 8, right	$5.15{\cdot}e^{-5}$	$1.86{\cdot}e^{-6}$	$1.34{\cdot}e^{-5}$
Fig. 10, left	$9.63{\cdot}e^{-2}$	$1.12{\cdot}e^{-3}$	$7.64{\cdot}e^{-3}$
Fig. 10, right	$1.85{\cdot}e^{-3}$	$1.08{\cdot}e^{-3}$	$3.37{\cdot}e^{-4}$

K. Commonalities and Differences between SEAL and GAV

Since the influential SEAL link prediction framework [17, 19] represents one of the most prominent works on learned, GNN-based link prediction algorithms, we would like to clearly state the commonalities and differences between SEAL and our proposed link prediction algorithm tailored to flow-driven spatial networks, GAV. Although GAV utilizes some concepts introduced by SEAL (subgraph extraction/classification & labeling trick), which are provably used in most competitive

approaches and can, therefore, be seen as common practices, we, for the first time, introduce the principle of physical flow to link prediction. To this end, we propose not only a novel flow-inspired, parameter-efficient message-passing layer updating vector embeddings but also a physically plausible readout module facilitating interpretability. Our contributions result in an increase of more than 22% in AUC compared to SEAL on ogbl-vessel.

L. Message-Passing Update Functions

We provide a concise overview of message-passing layers featured in our work and their respective high-level, final node update functions in Table 10. Here, d_i stands for the node degree of n_i, α_{ij} for the learned attention coefficient between n_i and n_j, and ϕ_θ for an arbitrary learnable function. We would like to highlight the simplicity of the GAV layer's final update function.

Table 10. Message-passing update functions.

Message-Passing Layer	Update Function
GAV layer	$\hat{n}_i = s_i \cdot n_i$
EdgeConv [16]	$\hat{n}_i = \frac{1}{\lvert\mathcal{N}(n_i)\rvert} \sum_{n_j \in \mathcal{N}(n_i)} \phi_\theta(n_i \,\|\, n_j - n_i)$
GAT layer [14]	$\hat{n}_i = \alpha_{ii} \cdot \phi_\theta(n_i) + \sum_{n_j \in \mathcal{N}(n_i)} \alpha_{ij} \cdot \phi_\theta(n_j)$
SAGE layer [3]	$\hat{n}_i = \phi_\theta^{(1)}(n_i) + \phi_\theta^{(2)}(\frac{1}{\lvert\mathcal{N}(n_i)\rvert} \sum_{n_j \in \mathcal{N}(n_i)} n_j)$
GCN layer [5]	$\hat{n}_i = \phi_\theta(\sum_{n_j \in \mathcal{N}(n_i) \cup n_i} \frac{1}{\sqrt{d_i \cdot d_j}} n_j)$

M. Medical Relevance of the Link Prediction Task for Whole-Brain Vessel Graphs

Since GAV has been developed around the ogbl-vessel benchmark, we would like to provide more details on the medical relevance and the application of link prediction algorithms for whole-brain vessel graphs. As already mentioned in Section 1, vascular network representations of the brain originate from a multi-stage, imperfect process, typically consisting of a segmentation stage followed by a graph extraction stage (skeletonization and pruning). Detailed pipelines for whole-brain vessel graph generation can be found in the literature [1, 7, 8, 15]. Each stage of the graph generation pipeline introduces noise and artifacts to the extracted whole-brain vessel graphs. The initial segmentation stage [13], *e.g.*, often results in under- or over-connected vessel segmentation masks, which in turn result in equally under- or over-connected whole-brain vessel graphs. This is mostly due to the shortage of annotated training data (especially in the 3D domain), which is required for accurate vessel segmentation via supervised state-of-the-art deep-learning-based segmentation techniques. Under-/over-connectivity, however, limits the application of whole-brain vessel graphs for subsequent medically relevant downstream tasks, such as the diagnosis, treatment, and analysis of neurovascular brain disorders (*e.g.*, aneurysms or strokes). This is because these downstream tasks require flawlessly connected whole-brain vessel graphs free of artifacts to obtain a deeper understanding of neurovascular brain disorders by, *e.g.*, accurately recognizing anomalies in blood flow patterns via blood flow modeling [9]. To overcome the obstacle of under-/over-connectivity in whole-brain vessel graphs and, therefore, to enable researchers to obtain a more accurate and advanced understanding of neurovascular brain disorder, one can either optimize whole-brain vessel graph generation pipelines [11, 12] or utilize the task of link prediction, which we extensively investigate in this work.

References

[1] Dominik Drees, Aaron Scherzinger, René Hägerling, Friedemann Kiefer, and Xiaoyi Jiang. Scalable robust graph and feature extraction for arbitrary vessel networks in large volumetric datasets. *BMC bioinformatics*, 22(1):1–28, 2021. 7

[2] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks. *Advances in neural information processing systems*, 33:1970–1981, 2020. 6

[3] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 7

[4] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv preprint arXiv:2005.00687*, 2020. 5

[5] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 7

[6] Stephen G Kobourov. Spring embedders and force directed graph drawing algorithms. *arXiv preprint arXiv:1201.3011*, 2012. 6

[7] Jennis Meyer-Spradow, Timo Ropinski, Jörg Mensmann, and Klaus Hinrichs. Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications*, 29(6):6–13, 2009. 7

[8] Johannes C Paetzold, Julian McGinnis, Suprosanna Shit, Ivan Ezhov, Paul Büschl, Chinmay Prabhakar, Anjany Sekuboyina, Mihail Todorov, Georgios Kaissis, Ali Ertürk, et al. Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 7

[9] Franca Schmid, Giulia Conti, Patrick Jenny, and Bruno Weber. The severity of microstrokes depends on local vascular topology and baseline perfusion. *Elife*, 10:e60208, 2021. 7

[10] Matthias Schneider, Johannes Reichold, Bruno Weber, Gábor Székely, and Sven Hirsch. Tissue metabolism driven arterial tree generation. *Medical image analysis*, 16(7):1397–1414, 2012. 3, 4

[11] Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. In *European Conference on Computer Vision*, pages 422–439. Springer, 2022. 7

[12] Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16560–16569, 2021. 7

[13] Mihail Ivilinov Todorov, Johannes Christian Paetzold, Oliver Schoppe, Giles Tetteh, Suprosanna Shit, Velizar Efremov, Katalin Todorov-Völgyi, Marco Düring, Martin Dichgans, Marie Piraud, et al. Machine learning analysis of whole mouse brain vasculature. *Nature methods*, 17(4):442–449, 2020. 7

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 7

[15] Thomas Wälchli, Jeroen Bisschop, Arttu Miettinen, Alexandra Ulmann-Schuler, Christoph Hintermüller, Eric P Meyer, Thomas Krucker, Regula Wälchli, Philippe P Monnier, Peter Carmeliet, et al. Hierarchical imaging and computational analysis of three-dimensional vascular network architecture in the entire postnatal and adult mouse brain. *Nature Protocols*, 16(10):4564–4610, 2021. 7

[16] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 5, 7

[17] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018. 5, 6

[18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 5

[19] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34:9061–9073, 2021. 5, 6