Endpoint Symmetry for Edge Updates: Weight-Space Redundancy in GNNs on Undirected Graphs

Charlotte Cambier van Nooten [1] **Stijn van den Beemt** [1] **Yuliya Shapovalova** [1] **Tom Heskes** [1]

Abstract

Graph Neural Networks (GNNs) for undirected graphs typically update edge representations by concatenating endpoint embeddings. Because an undirected edge $\{u, v\}$ has no canonical ordering, concatenation violates endpoint symmetry: the output depends on an arbitrary serialisation choice, not on the graph. The model must spend capacity learning a symmetry it cannot structurally enforce, with no guarantee of finding it. For any weight configuration θ, a partner θ' obtained by permuting the h_u and h_v input blocks of the first weight matrix produces the same model prediction on every undirected edge, inducing a redundant S_2 orbit in weight space so different seeds may land in either half. We resolve this with a Deep Sets aggregation $\rho(g, \phi(h_u) + \phi(h_v))$ that enforces endpoint symmetry by construction, collapses the orbit, and approximates any continuous endpoint-symmetric function arbitrarily well on compact domains. Experiments confirm three predictions: different outputs for (u, v) and (v, u), geometrically distinct seed configurations, and a pronounced interpolation barrier. A real-world power grid application yields higher recall, lower variance, and one-third fewer parameters. Matching architecture symmetry to data symmetry eliminates redundant orbits at no cost to expressivity.

1. Introduction

A core design principle for GNNs on undirected graphs is that predictions should not depend on the arbitrary ordering of nodes. For graph-level tasks this requires invariance to node permutations; for node-level tasks it requires equivariance, so that renaming node u to node v simply moves the prediction for u to position v. Standard message-passing GNNs achieve this through permutation-invariant neighbourhood aggregation (Xu et al., 2019; Bronstein et al., 2021). Edge-level tasks introduce a symmetry requirement that node-level designs do not address. An undirected edge $\{u, v\}$ is the same object as $\{v, u\}$: there is no meaningful distinction between the two endpoints. Any function that produces an embedding for this edge should therefore give the same final prediction regardless of which endpoint is presented first. We call this endpoint symmetry, a form of invariance, the edge embedding must be invariant to swapping u and v.

Standard GNN architectures that maintain evolving edge representations typically update them as

$$g^{(k)}_{\{u,v\}} = \xi\left(g^{(k-1)}_{\{u,v\}},\, h^{(k-1)}_u,\, h^{(k-1)}_v\right), \tag{1}$$

where ξ is an MLP receiving the concatenation of the previous edge embedding and the two endpoint representations. Unless ξ is explicitly constructed to be symmetric in its node arguments, this concatenation imposes a fixed order on h_u and h_v, an order that carries no information for undirected edges and is determined entirely by how the graph happens to be stored in memory. This has a consequence for the weight space. For any θ, the configuration θ', obtained by permuting the input blocks corresponding to h_u and h_v in the first weight matrix, satisfies $\xi_{\theta'}(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$ exactly and produces the same prediction on every undirected edge. The pair $\{\theta, \theta'\}$ is a redundant orbit: two points that are equivalent on undirected graphs but appear as different solutions to the optimizer. Such orbits duplicate local minima on the loss surface and have been shown to hinder optimisation and generalisation (Entezari et al., 2022; Ainsworth et al., 2023).

We propose to replace the asymmetric concatenation with a Deep Sets parameterisation (Zaheer et al., 2017) that enforces endpoint symmetry by construction. We formalise endpoint symmetry as S_2-invariance of edge update functions and show that concatenation-based updates violate it, forcing the model to spend capacity learning a symmetry it cannot structurally enforce and inducing a redundant S_2 orbit of geometrically distinct but functionally equiva-

[1]Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands. Correspondence to: Charlotte Cambier van Nooten <charlotte.cambiervannooten@ru.nl>.

Workshop on Weight-Space Symmetries, held in conjunction with the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

lent weight configurations (§3). The Deep Sets aggregation $\phi(h_u) + \phi(h_v)$ enforces endpoint symmetry by construction, collapses this orbit, and approximates any continuous endpoint-symmetric function arbitrarily well on compact domains (§3). We validate the symmetry violation and its weight-space consequences on a synthetic task with symmetric ground-truth labels, and confirm the principle transfers to n-1 contingency classification in medium-voltage power grids (§4, Appendix F).

2. Background and Related Work

Edge-centric GNNs. Most GNN architectures are node-centric: edges carry information between nodes but do not maintain their own learned representations (Kipf & Welling, 2017; Xu et al., 2019). For edge-level prediction this is insufficient when two edges share the same endpoints but differ in physical properties that determine their label. Edge-centric architectures address this by maintaining explicit edge embeddings $g^{(k)}_{\{u,v\}}$, updated in each layer k via Equation 1 and used alongside node embeddings in message computation (Cambier van Nooten et al., 2025):

$$m^{(k)}_{uv} = \mu\Big(g^{(k)}_{\{u,v\}}, h^{(k-1)}_u, h^{(k-1)}_v\Big), \tag{2}$$

$$h^{(k)}_v = \mathrm{MLP}^{(k)}\Big(h^{(k-1)}_v + \textstyle\sum_{u\in\mathcal{N}(v)} m^{(k)}_{uv}\Big). \tag{3}$$

The aggregation in Equation 3 is permutation-invariant over neighbours, so node-level equivariance is preserved. In edge-centric GNNs, each undirected edge $\{u, v\}$ is stored as two directed pairs, and the node update uses both: the output on (u, v) is used as a message to v, and the output on (v, u) as a message to u (Equation 2). The final prediction associated with an undirected edge $\{u, v\}$ should be a single well-defined quantity and must not depend on whether the model presented the endpoints as (u, v) or (v, u). For node-level tasks, this invariance is achieved by combining both orderings in the node update, so node-level model predictions are endpoint-invariant even when ξ_θ is not. For edge classification, however, the edge embedding $g^{(k)}_{\{u,v\}}$ is used directly as the prediction without such symmetrisation, so the asymmetry of ξ_θ propagates directly to the output. The edge update in Equation 1 is not symmetric in its endpoint arguments: the order in which h_u and h_v are supplied is fixed by the implementation, and a standard MLP has no guarantee of symmetry.

Transforming edges into nodes via the line graph construction (Cai et al., 2022) enforces endpoint symmetry by design, since edge–edge adjacency is inherently unordered. The drawback is that the line graph grows quadratically in node degree and discards the original node features, which limits applicability in domains where node attributes carry physical meaning.

Symmetry levels. The edge update ξ_θ operates on ordered inputs and is in general not endpoint-symmetric: $\xi_\theta(g, h_u, h_v) \neq \xi_\theta(g, h_v, h_u)$. For node-level tasks, the node update symmetrises predictions: $\psi_\theta(\{u, v\}) = \psi_\theta(\{v, u\})$ regardless of whether ξ_θ is symmetric. For edge classification, $g^{(k)}_{\{u,v\}}$ is used directly as the prediction, so endpoint symmetry is a requirement at the level of ξ_θ itself rather than a property of the readout.

Deep Sets. Zaheer et al. (2017) prove that any continuous permutation-invariant function on a set can be decomposed as

$$f(\{x_1, \ldots, x_n\}) = \rho\left(\sum_{i=1}^{n} \phi(x_i)\right), \tag{4}$$

for sufficiently expressive ϕ and ρ. For pairs $\{x_u, x_v\}$ this reduces to $\rho(\phi(x_u) + \phi(x_v))$, the S_2-invariant decomposition. The Deep Sets framework has been applied to point cloud learning (Ravanbakhsh et al., 2017) and set anomaly detection (Ostdiek, 2022), but has not, to our knowledge, been used to enforce endpoint symmetry in GNN edge update functions. The sum $\sum_i \phi(x_i)$ is a canonical aggregation that is both universal and permutation-invariant for continuous functions on sets of arbitrary size; mean and max pooling are special cases that sacrifice universality (Zaheer et al., 2017; Xu et al., 2019).

Weight-space symmetries. A weight-space symmetry exists whenever two distinct parameter configurations represent the same function. In MLPs, permuting hidden neurons together with their weights is the standard example (Entezari et al., 2022; Ainsworth et al., 2023): the function is invariant to this rearrangement, permuting hidden units and their associated weights leaves all predictions unchanged. These duplicate minima complicate gradient-based optimisation and are important for model merging methods such as Git Re-Basin (Ainsworth et al., 2023), which align these permutations to interpolate between independently trained models.

3. Endpoint Symmetry and Weight-Space Redundancy

3.1. Formalisation

Let $G = (V, E)$ be an undirected graph with node feature dimension d and edge embedding dimension d_e. Each edge $\{u, v\} \in E$ is an unordered pair, so the choice of which endpoint to label u and which v is a property of the data structure, not of the graph. Standard frameworks such as PyTorch Geometric represent undirected edges as pairs of directed edges (u, v) and (v, u), so an asymmetric edge update evaluates the same edge twice in opposite orders and may yield different embeddings for the same physical connection. The symmetry of interest is $S_2 = \{e, \tau\}$, the

symmetric group on two elements: e is the identity, τ is the swap, and applying τ twice returns to the identity ($\tau^2 = e$). On the inputs to the edge update, τ acts as $(g, h_u, h_v) \mapsto (g, h_v, h_u)$. Endpoint symmetry requires the edge update to be invariant under τ: it should produce the same output regardless of which endpoint is presented first.

Definition 3.1 (Endpoint symmetry)**.** Let $g \in \mathbb{R}^{d_e}$ denote the edge embedding $g^{(k-1)}_{\{u,v\}}$ from the previous message-passing layer, and let $h_u, h_v \in \mathbb{R}^d$ denote the node embeddings $h^{(k-1)}_u, h^{(k-1)}_v$ of the two endpoints at the same layer. An edge update function $\xi : \mathbb{R}^{d_e} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d_e}$ satisfies endpoint symmetry if it is invariant to the S_2 action that swaps the second and third arguments:

$$\xi(g, h_u, h_v) = \xi(g, h_v, h_u) \qquad \forall\, g \in \mathbb{R}^{d_e},\ h_u, h_v \in \mathbb{R}^d. \tag{5}$$

Equivalently, ξ factors through the unordered pair $\{h_u, h_v\}$: there exists $\bar{\xi}$ such that $\xi(g, h_u, h_v) = \bar{\xi}(g, \{h_u, h_v\})$.

Violating endpoint symmetry means the model's predictions depend on an arbitrary implementation detail, rather than on the graph structure itself. While endpoint symmetry is a natural requirement for functions defined on undirected edges, standard architectures do not enforce it at the level of ξ_θ. For node-level tasks, symmetric predictions are obtained by combining both endpoint orderings in the node update. For edge classification, no such combination occurs: the asymmetry of ξ_θ propagates directly to the output, making endpoint symmetry at the level of ξ_θ a basic correctness requirement.

3.2. Asymmetric updates induce redundant weight-space orbits

Consider a standard MLP ξ_θ taking concatenated inputs $[g, h_u, h_v] \in \mathbb{R}^{d_e+2d}$, with weight matrices $\theta = \{W_1 \in \mathbb{R}^{r\times(d_e+2d)}, W_2, \ldots, W_L\}$. Because endpoints appear in a fixed order, there is no structural constraint forcing $\xi_\theta(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$. Let P be the block permutation swapping the h_u and h_v blocks while fixing g, so $P\,[g, h_u, h_v]^\top = [g, h_v, h_u]^\top$ and $P^2 = I$ (explicit form in Appendix A).

Proposition 3.2 (Redundant S_2 orbit)**.** *The configuration $\theta' = \{W_1 P, W_2, \ldots, W_L\}$ is constructible from any θ and satisfies $\xi_{\theta'}(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$ exactly for every input, so θ and θ' assign the same embedding to every undirected edge. If ξ_θ violates endpoint symmetry, $\theta' \neq \theta$, and the pair $\{\theta, \theta'\}$ is a redundant S_2 orbit: two geometrically distinct weight vectors that produce the same prediction on every undirected edge. (See Appendix A.)*

Each undirected edge is represented as two directed pairs (u, v) and (v, u), and since both orderings are combined symmetrically in the node update (Equation 3), θ and θ' assign the same final prediction to every undirected edge: equivalent solutions that appear as different points to the optimizer. Such orbits duplicate local minima and have been shown to hinder optimisation and generalisation (Entezari et al., 2022; Ainsworth et al., 2023). Unlike neuron-permutation symmetries internal to the MLP, this orbit arises from the interface between model and data.

3.3. A symmetric parameterisation collapses the orbit

Theorem 3.3. *[Symmetric Deep Sets approximation] Any continuous edge update satisfying endpoint symmetry can be approximated arbitrarily well on compact domains by a parameterisation of the form*

$$g^{(k)}_{\{u,v\}} = \rho\Big(g^{(k-1)}_{\{u,v\}},\ \phi\Big(h^{(k-1)}_u\Big) + \phi\Big(h^{(k-1)}_v\Big)\Big), \tag{6}$$

where $\phi\colon \mathbb{R}^d \to \mathbb{R}^m$ and $\rho\colon \mathbb{R}^{d_e} \times \mathbb{R}^m \to \mathbb{R}^{d_e}$ are continuous functions, with m sufficiently large. This parameterisation satisfies endpoint symmetry by construction and collapses the redundant S_2 orbit of Section 3.2: no two distinct configurations (ϕ, ρ) are related by endpoint swapping while computing the same function. (Proof in Appendix B.)

The theorem claims approximation rather than exact representation: for any continuous endpoint-symmetric target function and any desired accuracy ε, a Deep Sets parameterisation with sufficiently large m and sufficiently expressive ϕ, ρ can approximate it arbitrarily well on compact domains. The approximation guarantee does not rely on the countable-universe assumption of Zaheer et al. (2017): sum-injectivity of ϕ follows from Amir et al. (2023) and approximation of ρ from Hornik et al. (1989). Exact universality requires m to scale with the input domain, which is infeasible; uniform approximation on compact sets is sufficient in practice. Node features in trained GNNs lie in bounded regions, and uniform approximation on compact sets is the standard notion of universality for MLPs (Hornik et al., 1989).

Remark 3.4. Equation 6 does not reduce expressive power for functions that depend on $\{h_u, h_v\}$ as an unordered pair. It excludes only ordering-sensitive functions, which are not appropriate on undirected graphs. The trade-off is in the input dimension to ρ: the asymmetric baseline receives $[g, h_u, h_v]$ of dimension d_e+2d, while the symmetric variant receives $[g, \phi(h_u) + \phi(h_v)]$ of dimension $d_e + m$. When $m < 2d$ the first layer of ρ is strictly smaller, saving $d_e \cdot (2d - m)$ parameters. When $m > 2d$, for instance when a wide ϕ is chosen for expressivity, the first layer of ρ grows larger than in the asymmetric case, the orbit-collapse benefit remains but the parameter count increases. Note that $m < 2d$ does not guarantee sufficient expressivity of ϕ: Amir et al. (2023) show that sum-injectivity of ϕ requires $m \geq 4d + 1$, so the parameter saving in ρ when $m < 2d$ may come at the cost of expressivity in ϕ when $m < 4d+1$.

4. Empirical Validation

We validate the endpoint symmetry violation and its weight-space consequences on a synthetic task where the ground-truth edge label is symmetric by construction. The setup isolates the effect of the edge update rule and tests three predictions that follow from the orbit argument of Section 3.2: the asymmetric model produces different predictions for (u, v) and (v, u); two asymmetric seeds converge to geometrically distinct weight configurations consistent with opposite halves of the S_2 orbit; and linear interpolation between them reveals a loss barrier. The symmetric model should exhibit none of these. Data, architectures, and training details are in Appendix E.

Endpoint symmetry violation. We evaluate each trained model on every undirected test edge with both endpoint orderings. Figure 1 plots the two outputs against each other. The asymmetric model's predictions scatter off the diagonal, showing that storage order directly affects the output. The symmetric model lies exactly on the diagonal, as guaranteed by order-independence of the sum. The mean absolute violation $|g^{(k)}_{\{u,v\}} - g^{(k)}_{\{v,u\}}|$ is 0.057 (max 0.962) for the asymmetric model (Table 1).

Weight-space orbit structure. Figure 2 (Appendix E) shows the orbit construction for the trained model. The h_u and h_v blocks of W_1 and $W_1 P$ are visibly swapped while the g block is unchanged. The smaller signed difference for the orbit swap ($\overline{|W_1 - W_1 P|} = 0.066$) compared to a random orthogonal transformation ($\overline{|W_1 - W_1 P_{\text{rand}}|} = 0.112$) suggests the model has absorbed some structure of the endpoint swap into its first-layer weights, though θ and θ' remain geometrically distinct. Pre-activations under θ' on (g, h_u, h_v) and under θ on (g, h_v, h_u) align on the anti-diagonal, confirming $\xi_{\theta'}(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$ exactly. The per-edge loss scatter shows $\mathcal{L}_\theta(u, v) \neq \mathcal{L}_{\theta'}(u, v)$, but since $\mathcal{L}_{\theta'}(u, v) = \mathcal{L}_\theta(v, u)$, the model cannot distinguish which ordering is better. The interpolation barrier of 0.169 (Table 1) confirms the two seeds converge to geometrically distinct basins.

Loss barrier between seeds. We train the asymmetric model from two different random seeds and compute the loss along the linear interpolation $\theta(\alpha) = (1-\alpha)\theta_A + \alpha\theta_B$ for $\alpha \in [0, 1]$. The barrier height $\max_\alpha \mathcal{L}(\theta(\alpha)) - \min_\alpha \mathcal{L}(\theta(\alpha))$ is 0.169 for the asymmetric model (Table 1): two seeds landing in opposite halves of the orbit assign opposite per-edge loss patterns, and linear interpolation passes through configurations consistent with neither half. The zero barrier for the symmetric model follows directly from the absence of a redundant S_2 orbit. The two asymmetric seeds achieve nearly identical test loss ($\Delta = 0.0051$, Table 2), consistent with functionally equivalent solutions in distinct weight-space locations, while the symmetric seeds show

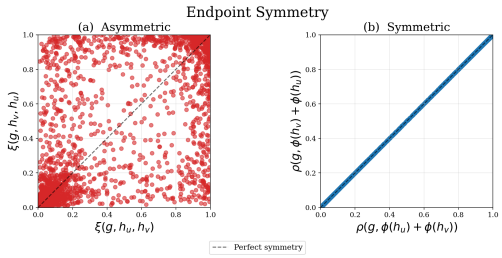

Figure 1. Endpoint symmetry violation. Each point is one undirected test edge. (a) Asymmetric baseline: edge embedding $g^{(k)}_{\{u,v\}}$ vs $g^{(k)}_{\{v,u\}}$, computed from orderings (u, v) and (v, u) respectively. (b) Symmetric architecture: $\rho(g, \phi(h_u) + \phi(h_v))$ vs $\rho(g, \phi(h_v) + \phi(h_u))$.

greater spread ($\Delta = 0.0231$), reflecting internal neuron permutation within ϕ and ρ, a known weight-space symmetry of MLPs, rather than the S_2 orbit structure. Across 15 seeds the asymmetric model achieves 0.834 ± 0.027 vs 0.784 ± 0.014, roughly twice the variance. Figure 3 confirms that the model optimises on the combined loss of both orderings: the symmetrised prediction $(\hat{y}_{uv} + \hat{y}_{vu})/2$ achieves lower loss (0.418) than either asymmetric prediction individually (0.458 and 0.468), consistent with the orbit being redundant at the model prediction level.

Real-world application. We apply the same two variants to n-1 contingency classification in medium-voltage power grids, a safety-critical edge classification task. Full details and results are reported in Appendix F: the symmetric variant achieves higher recall with lower variance across random splits and one-third fewer parameters. The variance reduction matters operationally, since in this domain variance in recall translates directly to variance in missed failures.

5. Discussion and Conclusion

Prior work on weight-space symmetries has focused on symmetries internal to architectures (neuron permutation, sign flips, rescaling) and their effects on loss landscapes and model merging (Entezari et al., 2022; Ainsworth et al., 2023). The S_2 orbit identified here is different; it arises from the interface between model and data, when the parameterisation fails to respect the endpoint invariance that undirected edges require. The broader principle is that redundant orbits can be collapsed by matching the symmetry group of the architecture to that of the data, removing the need to learn invariances that should be free, at no cost to expressive power. An asymmetric model has no structural guarantee of endpoint symmetry and must instead spend capacity trying to learn it from data, with no guarantee of success. The recall improvement on the power grid task is consequently modest; the primary gains are in parameter efficiency and initialisation stability. Looking forward, the argument extends naturally to higher-order hyperedges (Appendix C) and to directed or heterogeneous graphs.

Endpoint symmetry is a basic correctness requirement for edge update functions in GNNs on undirected graphs. Standard concatenation-based updates violate it, inducing a redundant S_2 orbit in weight space and forcing the model to spend capacity learning a symmetry it cannot structurally enforce. The Deep Sets parameterisation enforces endpoint invariance by construction, collapses the orbit, and can approximate any continuous endpoint-symmetric function arbitrarily well on compact domains, with the approximation guarantee following from Amir et al. (2023) and Hornik et al. (1989); in practice m should be chosen with the sum-injectivity threshold $m \geq 4d + 1$ in mind. Empirical results on safety-critical power grid classification confirm improved recall, reduced variance across initialisations, and one-third fewer parameters. Matching architecture symmetry to data symmetry is a broadly applicable strategy for producing more compact, better-conditioned GNN models.

References

Ainsworth, S., Hayase, J., and Srinivasa, S. Git Re-Basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

Amir, T., Gortler, S., Avni, I., Ravina, R., and Dym, N. Neural injective functions for multisets, measures and graphs via a finite witness theorem. In *Advances in Neural Information Processing Systems*, 2023.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Liò, P., and Bronstein, M. Weisfeiler and Lehman go topological: Message passing simplicial networks. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021.

Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Cai, L., Li, J., Wang, J., and Ji, S. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5103–5113, 2022.

Cambier van Nooten, C., van de Poll, T., Füllhase, S., Heres, J., Heskes, T., and Shapovalova, Y. Graph neural networks for assessing the reliability of the medium-voltage grid. *Applied Energy*, 384:125401, 2025.

de Haan, P., Cohen, T. S., and Welling, M. Natural graph networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3636–3646. Curran Associates, Inc., 2020.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*, 2022.

Hajij, M., Zamzmi, G., Papamarkou, T., Natesan Ramamurthy, K., Birdal, T., and Schaub, M. T. *Topological Deep Learning: Going Beyond Graph Data*. arXiv preprint arXiv:2206.00606, 2023.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, January 1989. ISSN 0893-6080.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

Ostdiek, B. Deep set auto encoders for anomaly detection in particle physics. *SciPost Physics*, 12(1):045, 2022.

Ravanbakhsh, S., Schneider, J., and Poczos, B. Deep learning with sets and point clouds. In *International Conference on Learning Representations (Workshop)*, 2017.

van de Sande, P., Danes, M., and Dekker, T. ANDES: Grid capacity planning using a bottom-up, profile-based load forecasting approach. *CIRED*, 2017(1):2097–2100, 2017.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3391–3401. Curran Associates, Inc., 2017.

A. Construction of the S_2 orbit partner

This appendix provides the explicit construction referenced in Section 3.2: given any asymmetric MLP ξ_θ, we construct a distinct weight vector θ' that produces the same prediction on every undirected edge.

Let ξ_θ be an MLP with concatenated input $[g, h_u, h_v] \in \mathbb{R}^{d_e+2d}$ and weight matrices $W_1 \in \mathbb{R}^{r\times(d_e+2d)}, W_2, \ldots, W_L$. Define $P \in \mathbb{R}^{(d_e+2d)\times(d_e+2d)}$ as the block permutation

$$P = \begin{pmatrix} I_{d_e} & 0 & 0 \\ 0 & 0 & I_d \\ 0 & I_d & 0 \end{pmatrix},$$

so that $P[g, h_u, h_v]^\top = [g, h_v, h_u]^\top$. P leaves the edge state unchanged and swaps the h_u and h_v blocks. Note $P^2 = I$.

Functional equivalence and distinctness. Let $\theta' = \{W_1P, W_2, \ldots, W_L\}$, replacing only the first weight matrix. For any input:

$$\begin{aligned} \xi_{\theta'}(g, h_u, h_v) &= \sigma\big(W_1P\,[g, h_u, h_v]^\top\big) + \cdots \\ &= \sigma\big(W_1\,[g, h_v, h_u]^\top\big) + \cdots \\ &= \xi_\theta(g, h_v, h_u), \end{aligned} \tag{7}$$

where σ is the elementwise activation and $W_2, \ldots, W_L$ are unchanged. Since standard frameworks process both (u, v) and (v, u) during message passing and combine their outputs symmetrically in the node update (Equation 3), θ and θ' produce the same result on every undirected edge.

Since ξ_θ violates endpoint symmetry, there exist inputs with $\xi_\theta(g, h_u, h_v) \neq \xi_\theta(g, h_v, h_u)$. If $W_1P = W_1$, then $\xi_{\theta'}(g, h_u, h_v) = \xi_\theta(g, h_u, h_v)$ for all inputs, which combined with (7) gives $\xi_\theta(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$, contradicting endpoint symmetry violation. Hence $W_1P \neq W_1$ and $\theta' \neq \theta$. Applying the swap twice returns to θ since $P^2 = I$, confirming a non trivial S_2 orbit.

Conditionality. The orbit is a redundant pair only when the model has approximately converged on a symmetric task. If the model has not converged, or the task does not require endpoint symmetry, θ and θ' may simply be two functions with different losses.

B. Proof of Theorem 3.3

In this theorem we look at three parts: the Deep Sets parameterisation satisfies endpoint symmetry, it approximates any continuous endpoint-symmetric function arbitrarily well on compact domains, and its S_2 orbit is trivial. For the second claim, let $K \subset \mathbb{R}^d$ be any compact set and $\varepsilon > 0$ be any desired approximation accuracy; we show that sufficiently expressive ϕ and ρ achieve maximum error at most ε over $\mathbb{R}^{d_e} \times K \times K$, as stated in the theorem.

Endpoint symmetry. For any g, h_u, h_v and any choice of ϕ, ρ:

$$\rho(g,\, \phi(h_u) + \phi(h_v)) = \rho(g,\, \phi(h_v) + \phi(h_u))\,,$$

where the equality follows from $a + b = b + a$ for $a, b \in \mathbb{R}^m$. This holds for every input and every choice of (ϕ, ρ), regardless of training: endpoint symmetry is a structural property of the parameterisation, not a learned behaviour.

Example. Let $\phi(h) = Wh$ and $\rho(g, z) = Vz$ for matrices W, V. Then $\rho(g, \phi(h_u) + \phi(h_v)) = V(Wh_u + Wh_v) = VW(h_u + h_v)$. Since $h_u + h_v = h_v + h_u$, the output is the same for both orderings.

Approximation on compact domains. Let $K \subset \mathbb{R}^d$ be compact and let f be a continuous endpoint-symmetric function on $\mathbb{R}^{d_e} \times K \times K$. We want to approximate f uniformly by a parameterisation of the form $\rho(g, \phi(h_u) + \phi(h_v))$ with a single encoder ϕ that does not depend on g. We want to show that for any $\varepsilon > 0$ there exist MLPs ϕ and ρ such that $|\rho(g, \phi(h_u) + \phi(h_v)) - f(g, h_u, h_v)| < \varepsilon$ everywhere on the domain. We construct this in four steps: establishing the sum decomposition structure via Zaheer et al. (2017), constructing an injective sum encoder via Amir et al. (2023), using injectivity to define an exact intermediate target $\bar{\rho}$, and approximating $\bar{\rho}$ by a finite-width MLP via Hornik et al. (1989).

Sum decomposition. Fix any g. The map $(h_u, h_v) \mapsto f(g, h_u, h_v)$ is a continuous S_2-invariant function of the unordered pair $\{h_u, h_v\}$. By Zaheer et al. (2017), any continuous permutation-invariant function on a compact domain can be written in the form $\rho'(\sum_i \phi(x_i))$ for sufficiently expressive ϕ and ρ'. Applied to pairs, this gives the sum structure $\phi(h_u) + \phi(h_v)$. Restoring the dependence on g by extending ρ' to $\rho(g, \cdot)$ gives Equation 6. The argument also works in reverse: any function representable by $\rho(g, \phi(h_u) + \phi(h_v))$ satisfies endpoint symmetry, so the class of representable functions is exactly the class of continuous endpoint-symmetric functions.

Injective sum encoder. For the decomposition to be useful, ρ must be able to recover $\{h_u, h_v\}$ from $\phi(h_u) + \phi(h_v)$. By Amir et al. (2023), for $d > 1$ there exists a shallow MLP $\phi(h) = \sigma(Ah + b)$ with analytic activation and width $m = 4d + 1$ such that the sum encoder $\Phi(\{h_u, h_v\}) = \phi(h_u) + \phi(h_v)$ is injective on multisets of size 2 drawn from K. That is, $\Phi(\{h_u, h_v\}) = \Phi(\{h'_u, h'_v\})$ implies $\{h_u, h_v\} = \{h'_u, h'_v\}$ as multisets.

Continuous inverse. Since $\Phi(\{h_u, h_v\}) = \phi(h_u) + \phi(h_v)$ is a continuous injection on the compact set $K \times K$, it is a homeomorphism onto its image, so the inverse $\Psi = \Phi^{-1}$ exists and is continuous. Since f is endpoint-symmetric it is well-defined on multisets, we can write $\tilde{f}(g, \{h_u, h_v\}) = f(g, h_u, h_v)$ without ambiguity. Define $\bar{\rho}(g, z) = \tilde{f}(g, \Psi(z))$. Then $\bar{\rho}(g, \Phi(\{h_u, h_v\})) = \tilde{f}(g, \{h_u, h_v\}) = f(g, h_u, h_v)$, so $\bar{\rho}$ is a continuous function of (g, z) on a compact domain.

MLP approximation. By Hornik et al. (1989), any continuous function on a compact domain can be approximated uniformly by a finite-width MLP. Applied to $\bar{\rho}$, for any $\varepsilon > 0$ there exists an MLP ρ such that $\sup_{(g,z)} |\rho(g, z) - \bar{\rho}(g, z)| < \varepsilon$.

Combining. For any (g, h_u, h_v) in the domain:

$$\begin{aligned} |f(g, h_u, h_v) - \rho(g, \phi(h_u) + \phi(h_v))| &= |\bar{\rho}(g, \Phi(\{h_u, h_v\})) - \rho(g, \Phi(\{h_u, h_v\}))| \\ &\leq \sup_{(g,z)} |\bar{\rho}(g, z) - \rho(g, z)| \\ &< \varepsilon. \end{aligned}$$

The first equality uses the construction of $\bar{\rho}$; the inequality uses the fact that a pointwise value is bounded by the supremum; the final bound follows from MLP universality. The Zaheer et al. (2017) countable-universe assumption is not required: we invoke the Deep Sets theorem only to establish the sum decomposition structure, and the approximation guarantee follows from Amir et al. (2023) and Hornik et al. (1989).

Why approximation, not exact universality. Exact universality in the Deep Sets sense requires ϕ to map into a space whose dimension grows with the input domain: the original proof uses dimension k, and subsequent analyses suggest $d + d^2 + \cdots + d^{2k-1}$ for k-element multisets (Zaheer et al., 2017); in our case $k = 2$, suggesting $m \geq d + d^2$ for a strict guarantee. For continuous domains this is infeasible in finite dimensions. Universal approximation on compact sets means instead that as m and the capacity of ϕ and ρ grow, the representable function class becomes dense in the class of continuous endpoint-symmetric functions on $K \times K$. In practice we use $m = d$ as a computational compromise. Amir et al. (2023) show that a shallow MLP with analytic activation and width $m = 4d + 1$ achieves sum-injectivity for pairs in $\mathbb{R}^d$.

Dimension requirement for ϕ and ρ. For any fixed error ε and compact set K, the required m depends on how finely ϕ must separate distinct unordered pairs. If m is too small, the map $\{h_u, h_v\} \mapsto \phi(h_u) + \phi(h_v)$ may fail to be injective on unordered pairs: distinct pairs $\{h_u, h_v\} \neq \{h'_u, h'_v\}$ may satisfy $\phi(h_u) + \phi(h_v) = \phi(h'_u) + \phi(h'_v)$, and no ρ can subsequently distinguish them within tolerance ε. Injectivity of ϕ alone is not sufficient; what is required is injectivity of the sum map on unordered pairs, which places constraints on both m and the form of ϕ. The Deep Sets theorem (Zaheer et al., 2017) guarantees that for sufficiently large m there exists a ϕ such that the map $\{h_u, h_v\} \mapsto \phi(h_u) + \phi(h_v)$ is injective on pairs drawn from K. With injectivity of the sum and sufficient expressivity of ρ, the composition $\rho(\phi(h_u) + \phi(h_v))$ can represent any continuous S_2-invariant function, and the error over $\mathbb{R}^{d_e} \times K \times K$ can be made arbitrarily small.Amir et al. (2023) show that a shallow MLP with analytic activation and width $m = 4d + 1$ achieves this for pairs in $\mathbb{R}^d$. Our experiments use $m = d$, it reduces the parameter count relative to concatenation while retaining sufficient expressivity. For applications requiring exact universality, and for smaller ε or larger K, m should be chosen larger.

Why the sum, and not mean or max. Mean pooling is equivalent to sum pooling up to rescaling of ρ. Element-wise max is S_2-invariant but fails approximation universality: it cannot distinguish $\{1, 3\}$ from any other pair with the same coordinate-wise maxima, regardless of m. Sum pooling with a nonlinear ϕ of sufficient output dimension is the standard choice that is both S_2-invariant and universally approximating on continuous domains (Zaheer et al., 2017; Xu et al., 2019).

Trivial S_2 orbit. Suppose (ϕ, ρ) and (ϕ', ρ') are two parameter configurations satisfying

$$\rho(g, \phi(h_u) + \phi(h_v)) = \rho'(g, \phi'(h_u) + \phi'(h_v))$$

for all g, h_u, h_v. Swapping $h_u \leftrightarrow h_v$ leaves both sides unchanged, since $\phi(h_u) + \phi(h_v) = \phi(h_v) + \phi(h_u)$ and likewise for ϕ'. No permutation of (h_u, h_v) maps one configuration to a distinct one while computing a different function: the swap leaves the function unchanged by construction, so there is no distinct (ϕ', ρ') that could be obtained from (ϕ, ρ) by applying the S_2 swap. The orbit of any (ϕ, ρ) under the endpoint-swapping action is therefore a singleton $\{(\phi, \rho)\}$, and no redundant orbit exists. □

Contrast with the asymmetric case. For a concatenation-based MLP (asymmetric), applying the swap to the weights produces the distinct point $\{W_1 P, W_2, \ldots\}$ in weight space (Appendix A). For the Deep Sets parameterisation there is no analogous construction: the sum $\phi(h_u) + \phi(h_v)$ is already invariant, so no rearrangement of the parameters produces a different weight vector computing the same function.

C. Generalisations

The endpoint symmetry argument extends beyond edge pairs. For a hyperedge involving k nodes, the relevant symmetry group is S_k, and the Deep Sets construction generalises to $\rho\big(\sum_{i \in e} \phi(h_i)\big)$. Failing to enforce symmetry introduces a redundant S_k orbit: any permutation of the k endpoint blocks in the first weight matrix yields a functionally equivalent but geometrically distinct parameter configuration, giving $k!$ equivalent configurations in the same orbit. For $k = 2$ this is the S_2 orbit studied in this paper; for $k = 3$ the orbit has size 6; for $k = 4$ it has size 24. The orbit grows factorially with hyperedge order, making symmetry enforcement increasingly important for simplicial complex and higher-order GNN architectures (Bodnar et al., 2021; Hajij et al., 2023). For the symmetric parameterisation, Amir et al. (2023) guarantee sum-injectivity of ϕ with width $m = 2kd + 1$ (recovering $4d + 1$ at $k = 2$), so the required expressivity of ϕ grows linearly with hyperedge order.

D. Remarks on the Symmetric Parameterisation

Remark D.1. Equation 6 does not reduce expressive power on the class of functions that depend on $\{h_u, h_v\}$ as an unordered pair. It excludes only ordering-sensitive functions, which are inadmissible on undirected graphs. The trade-off is in the input dimension to ρ: the asymmetric baseline receives $[g, h_u, h_v]$ of dimension $d_e + 2d$, while the symmetric variant receives $[g, \phi(h_u) + \phi(h_v)]$ of dimension $d_e + m$. When $m < 2d$ the first layer of ρ is strictly smaller, saving $d_e \cdot (2d - m)$ parameters. Amir et al. (2023) show that $m = 4d + 1$ suffices for a strict sum-injectivity guarantee; below this threshold expressivity may be limited. When $m > 2d$, for instance when a wide ϕ is chosen for expressivity, the first layer of ρ grows larger than in the asymmetric case; the orbit-collapse benefit remains but the parameter count increases.

Remark D.2. Other symmetric aggregators (mean, element-wise max) also satisfy endpoint symmetry, but the sum decomposition is the unique universal choice for continuous functions on sets of arbitrary size (Zaheer et al., 2017; Xu et al., 2019). Mean pooling is equivalent to sum pooling up to rescaling of ρ; element-wise max fails universality since it cannot distinguish multisets with the same coordinate-wise maxima. Appending $|\phi(h_u) - \phi(h_v)|$ recovers sensitivity to magnitude differences between endpoints without breaking S_2 invariance, at the cost of reintroducing m input dimensions to ρ.

Remark D.3. Both ϕ and ρ are necessary. Summing raw embeddings $h_u + h_v$ directly is symmetric but insufficiently expressive: distinct endpoint pairs may map to the same sum, and no ρ can distinguish them. ϕ is a learned nonlinear transformation applied per endpoint before aggregation; by Amir et al. (2023), a shallow MLP ϕ with analytic activation and width $m = 4d + 1$ ensures distinct unordered pairs produce distinct aggregated representations. ρ then maps this aggregation, together with the current edge state g, to the updated embedding. Dropping either component reduces to a strictly weaker model.

Relation to equivariant GNNs. A parallel line of work constructs GNNs equivariant to larger groups such as rotation and reflection (Bronstein et al., 2021). Natural Graph Networks (de Haan et al., 2020) generalise equivariance to local automorphism groups of edge neighbourhoods. The present contribution is narrower but targeted: the local S_2 symmetry of undirected edge endpoints is systematically violated by standard architectures, with concrete consequences for weight-space structure.

E. Synthetic Experiment

Data

Each graph has $n = 15$ nodes with $d = 10$-dimensional node features drawn i.i.d. from $\mathcal{N}(0, 0.25I)$, forming a complete undirected graph with $\binom{n}{2} = 105$ edges (stored as 210 directed pairs for message passing). The ground-truth label for edge $\{u, v\}$ is:

$$y_{\{u,v\}} = \sigma\big(0.8\big(\cos(h_u^\top w) + \cos(h_v^\top w)\big)\big)\,, \tag{8}$$

where $w \in \mathbb{R}^d$ is a fixed unit-normalised random weight vector and σ is the sigmoid. The label decomposes as a sum of per-node terms, so $y_{\{u,v\}} = y_{\{v,u\}}$ by construction: the ground truth is perfectly symmetric. Labels are binarised at the 50th percentile to maintain approximate class balance. We generate 250 graphs and use a 70/30 train/test split.

Models

Both models share the same overall architecture: $K = 3$ message-passing steps, hidden dimension 64, edge embedding dimension $d_e = 8$, trained with BCE loss and Adam (lr $= 8 \times 10^{-5}$, up to 500 epochs, early stopping with patience 30 for asymmetric and 50 for symmetric). The two models differ only in how the edge update combines the two endpoint embeddings.

Asymmetric model. The edge update concatenates the current edge embedding g with both endpoint embeddings and applies an MLP:

$$g_{\{u,v\}}^{(k)} = \mathrm{MLP}\big([g_{\{u,v\}}^{(k-1)},\, h_u^{(k-1)},\, h_v^{(k-1)}]\big). \tag{9}$$

The MLP input has dimension $d_e + 2d = 28$. Because the concatenation treats h_u and h_v in a fixed order, the function is not invariant to swapping the endpoints. For every weight configuration θ that solves the task, there exists a distinct configuration θ', obtained by right-multiplying the first weight matrix by the swap permutation P, that computes the same function with the endpoints reversed. On undirected edges these two configurations are functionally equivalent but appear as different solutions to gradient descent (Section 3.2, Appendix A).

Symmetric model. The edge update applies a shared encoder ϕ to each endpoint independently and sums the results before passing to the update network ρ:

$$g_{\{u,v\}}^{(k)} = \rho\big(g_{\{u,v\}}^{(k-1)},\, \phi(h_u^{(k-1)}) + \phi(h_v^{(k-1)})\big). \tag{10}$$

Here $\phi : \mathbb{R}^d \to \mathbb{R}^m$ with $m = 10$, so the input to ρ has dimension $d_e + m = 18$, strictly smaller than the asymmetric case. Commutativity of addition ensures the output is invariant to swapping h_u and h_v for any parameter configuration, collapsing the redundant orbit to a single configuration (Theorem 3.3). The encoder ϕ is a two-layer MLP; without the nonlinear ϕ, summing raw embeddings $h_u + h_v$ would be symmetric but insufficiently expressive, as distinct endpoint pairs can produce identical sums. Amir et al. (2023) show that a strict sum-injectivity guarantee requires width $m = 4d + 1 = 41$; our experiments use $m = 10$ as a practical compromise that reduces parameter count while retaining sufficient expressivity for the task considered. ρ combines the aggregated representation with the current edge state g to produce the updated embedding.

Results

Figure 2 provides weight-space evidence consistent with the orbit construction of Appendix A. The h_u and h_v column blocks are visibly swapped between W_1 and W_1P while the g block is unchanged, exactly as the construction gives θ' from θ: replacing only the first weight matrix with W_1P pre-applies the endpoint swap before the first linear map, so all subsequent layers are unaffected. The signed difference $W_1 - W_1P$ shows the h_u and h_v columns are strongly non-zero and opposite in sign, showing the trained model weights the two endpoints differently at the message update level. This asymmetric weighting is the direct signature of the endpoint symmetry violation (Definition 3.1): a model that respected endpoint symmetry would have approximately equal h_u and h_v column blocks in W_1 and the difference would be near zero throughout.

The signed difference $W_1 - W_1P_{\mathrm{rand}}$ for a random orthogonal transformation of all input columns has a larger mean absolute element-wise difference ($\overline{|W_1 - W_1P_{\mathrm{rand}}|} = 0.112$) than the orbit swap ($\overline{|W_1 - W_1P|} = 0.066$). Despite moving the weights further, P_{rand} does not preserve functional equivalence on undirected edges, confirming that orbit partners are not defined by proximity in weight space but by the specific structure of the endpoint swap that the theory predicts.

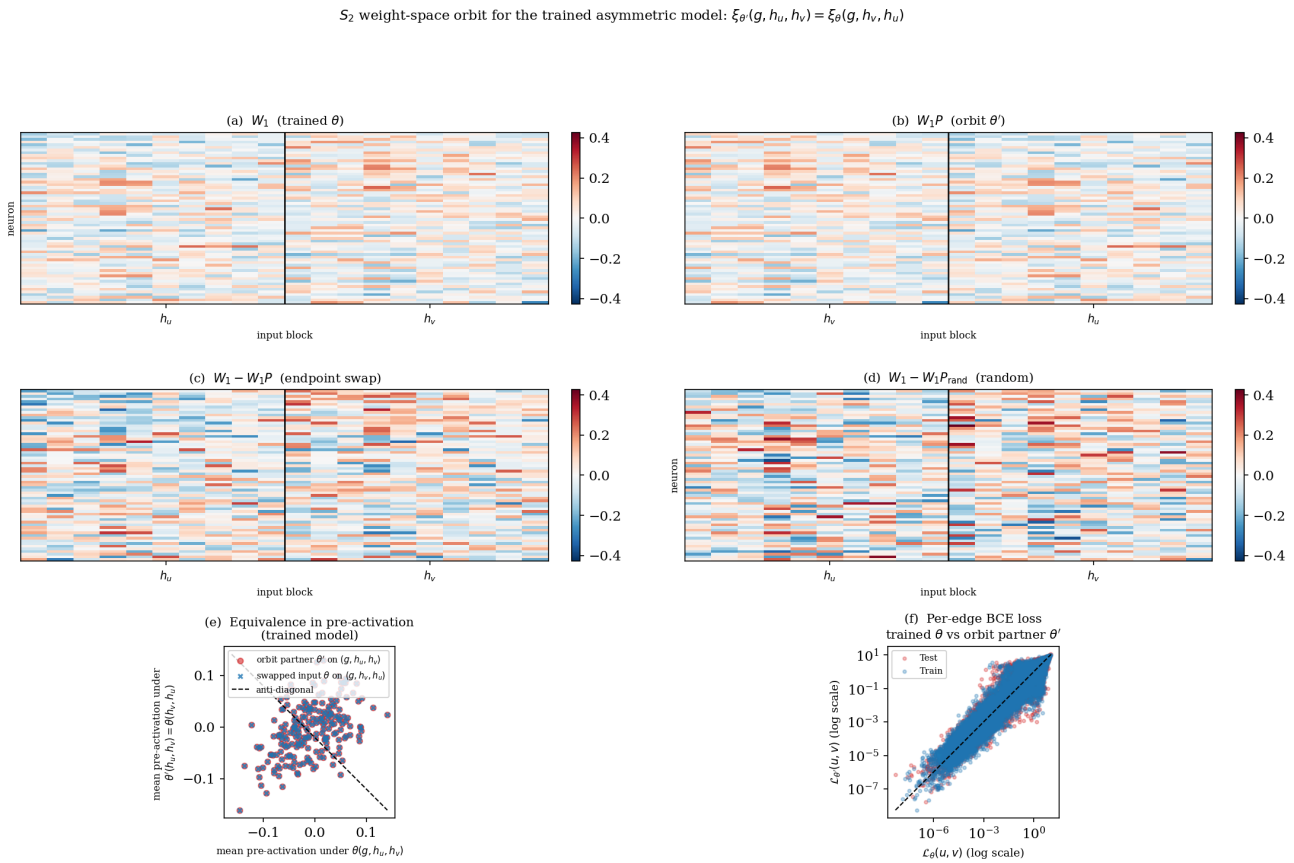

Figure 2. Visualisation of the S_2 weight-space orbit for the trained asymmetric model. (a) The first weight matrix W_1 (configuration θ) of the trained model. (b) The permuted weight matrix W_1P (configuration θ'), where P swaps the h_u and h_v input blocks. (c) Signed difference $W_1 - W_1P$ ($\overline{|W_1 - W_1P|} = 0.066$). (d) Signed difference $W_1 - W_1P_{\text{rand}}$, where P_{rand} is a random orthogonal transformation of all input columns ($\overline{|W_1 - W_1P_{\text{rand}}|} = 0.112$). (e) Mean pre-activations under θ' on (g, h_u, h_v) vs. under θ on (g, h_v, h_u). (f) Per-edge binary cross-entropy loss (log scale) for a trained asymmetric model, plotting the loss computed using weights θ against the loss computed using the permuted weights θ' for both train and test edges.

The mean pre-activations under θ' on (g, h_u, h_v) and under θ on the swapped input (g, h_v, h_u) align on the anti-diagonal for all 200 sampled input pairs, confirming $\xi_{\theta'}(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$ numerically. The mean pre-activations align on the anti-diagonal rather than the diagonal because the learned weights have opposite signs for the h_u and h_v blocks: swapping the endpoints negates the mean pre-activation, while the absolute functional equivalence $\xi_{\theta'}(g, h_u, h_v) = \xi_\theta(g, h_v, h_u)$ holds after the nonlinearity. Since standard frameworks process both (u, v) and (v, u) and combine their outputs symmetrically in the node update, θ and θ' produce the same result on every undirected edge, verifying the functional equivalence claimed in Appendix A.

The per-edge BCE loss scatter under θ vs θ' lies substantially off the diagonal for both train and test edges: the same physical edge receives very different losses depending on which weight configuration is used. This is a direct consequence of the symmetry violation, the model has committed to one half of the S_2 orbit during training, processing (u, v) and (v, u) differently. An endpoint-symmetric model would assign identical losses to θ and θ' by construction, as there is no distinct orbit partner to compare against. Note that panels (e) and (f) compute pre-activations and losses per directed ordering, not the standard symmetrised loss: this is intentional, to make the asymmetry at the message update level directly visible.

Table 1 reports the interpolation barrier height between two independently trained seeds. The asymmetric model's pronounced barrier (0.169) indicates the two seeds converged to solutions separated by a region of elevated loss, consistent with landing in different halves of the S_2 orbit. The symmetric model's barrier is zero by construction, as the Deep Sets parameterisation has no redundant orbit to separate seeds into distinct basins. Across 15 seeds, the asymmetric model achieves higher mean test loss and substantially higher variance, consistent with a loss surface containing duplicate minima at varying positions

Table 1. Synthetic experiment. Test loss: mean ± std over 15 seeds. Barrier: height $\max_\alpha \mathcal{L}(\theta(\alpha)) - \min_\alpha \mathcal{L}(\theta(\alpha))$ along the linear interpolation between two seeds. Viol. mean and Viol. max: mean and maximum of $|g^{(k)}_{\{u,v\}} - g^{(k)}_{\{v,u\}}|$ over test edges at convergence for a single seed, where $g^{(k)}_{\{u,v\}}$ denotes the edge embedding computed from ordering (u, v). For the symmetric model the violation is zero by construction and is omitted.

Model	Test loss	Barrier	Viol. mean	Viol. max
Asymmetric	0.834 ± 0.027	0.169	0.057	0.962
Symmetric	$\mathbf{0.784 \pm 0.014}$	$\mathbf{2.3 \times 10^{-7}}$	0 (by construction)	

Table 2. Train and test loss for seeds $A = 1$ and $B = 99$. The asymmetric seeds achieve nearly identical test loss ($\Delta = 0.0051$), consistent with two functionally equivalent solutions in distinct weight-space locations; the symmetric seeds show greater spread ($\Delta = 0.0231$), reflecting internal neuron permutation orthogonal to the S_2 endpoint symmetry.

	Seed A		**Seed B**		
Model	Train	Test	Train	Test	Δ Test
Asymmetric	0.3437	0.4279	0.3398	0.4228	0.0051
Symmetric	0.3082	0.3282	0.3068	0.3513	0.0231

for different initialisations. The higher variance could in principle reflect neuron-permutation symmetries rather than the S_2 endpoint orbit, but neuron permutations affect both models equally since both use MLPs of the same architecture. The key distinguishing evidence is the interpolation barrier: 0.169 for the asymmetric model and 2.3×10^{-7} for the symmetric model, consistent specifically with the S_2 orbit structure, which would not produce a barrier in the symmetric model.

Table 2 examines the two independently trained seeds at the level of individual losses. If the two asymmetric seeds are orbit partners, functionally equivalent solutions in distinct weight-space locations, their test losses should be nearly identical, since the training objective cannot distinguish them. The symmetric model has no such orbit structure, so its two seeds are free to converge to genuinely different solutions, producing greater spread. The table confirms both predictions: the asymmetric seeds differ by only $\Delta = 0.0051$ in test loss, while the symmetric seeds differ by $\Delta = 0.0231$. The larger symmetric spread reflects internal neuron permutation within ϕ and ρ, a known weight-space symmetry of MLPs that is independent of the S_2 endpoint orbit.

Figure 3 compares the per-edge loss under three conditions: the symmetrised prediction $(\hat{y}_{uv} + \hat{y}_{vu})/2$, the asymmetric prediction for ordering (u, v), and the asymmetric prediction for ordering (v, u). The symmetrised loss (0.418) is consistently lower than both asymmetric losses (0.458 and 0.468), confirming that the model optimises on the combined loss of both orderings rather than on a single ordering. This is visible in the scatter plot (log scale): most points lie below the diagonal, meaning the symmetrised prediction is closer to the true label than either asymmetric prediction individually. Here $\hat{y}_{uv} = g^{(k)}_{\{u,v\}}$ denotes the edge embedding computed from ordering (u, v), and the symmetrised prediction $(\hat{y}_{uv} + \hat{y}_{vu})/2$ is a post-hoc averaging of both orderings, distinct from the node-level symmetric prediction ψ_θ which is symmetric by construction through the node update.

Figure 4 shows the evolution of $\overline{|W_1^{h_u} - W_1^{h_v}|}$ during training for two models: one with standard random initialisation and one initialised symmetrically ($\theta = \theta'$, i.e. the h_u and h_v blocks of W_1 set to their average). Symmetric initialisation is not preserved during training: the model breaks endpoint symmetry actively, converging to a similar level of asymmetry as the randomly initialised model by epoch 60. The difference $W_1^{\text{random}} - W_1^{\text{sym init}}$ after training is unstructured noise with no block pattern, confirming both models converge to similarly asymmetric but independently different solutions. This shows that the asymmetry in trained models is not an artefact of random initialisation but emerges actively during training, reinforcing the need for a structural solution such as the Deep Sets parameterisation.

F. Power Grid Application: Details and Results

Task and dataset. We evaluate on n-1 contingency assessment in medium-voltage (MV) power distribution grids. Each cable (edge) must be classified as critical (its single failure causes an unrecoverable supply interruption under the n-1 criterion) or non-critical. Recall of critical edges is the primary operational metric: a missed critical cable can lead to blackouts, while a false alarm triggers preventive action. The dataset comprises 159 real MV grid topologies from a Dutch

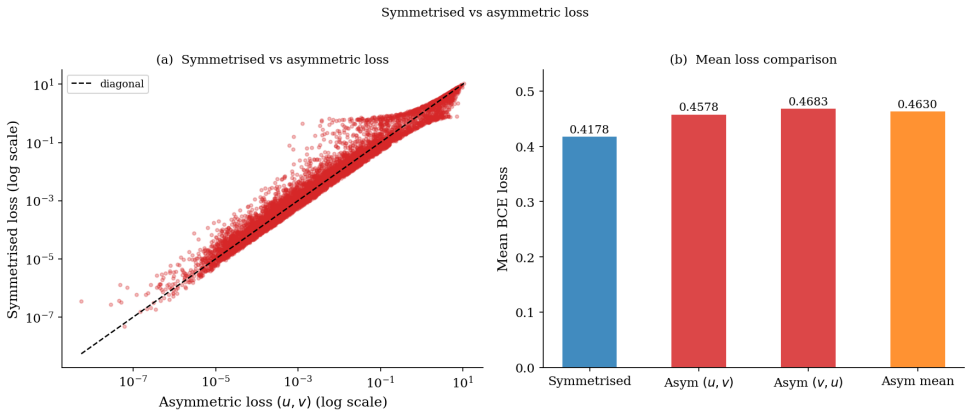

Figure 3. Symmetrised vs asymmetric loss. (a) Per-edge symmetrised loss vs asymmetric loss for ordering (u, v) (log scale). (b) Mean BCE loss for symmetrised and asymmetric predictions.

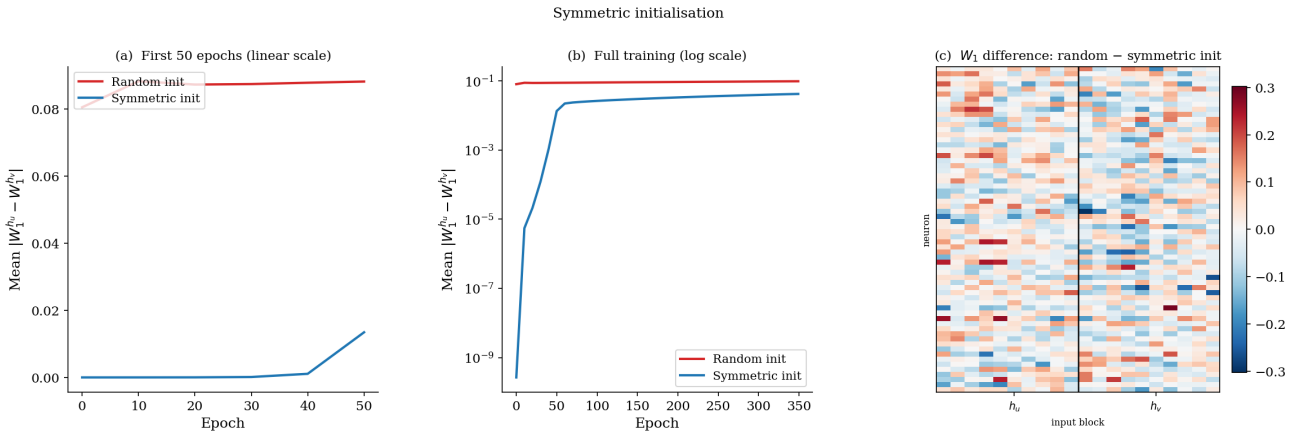

Figure 4. Symmetric initialisation experiment. (a) Mean absolute difference $\overline{|W_1^{h_u} - W_1^{h_v}|}$ over the first 50 epochs (linear scale). (b) Mean absolute difference over full training (log scale). (c) Signed difference $W_1^{\text{random}} - W_1^{\text{sym init}}$ in the h_u and h_v blocks after training.

Distribution System Operator (DSO) (van de Sande et al., 2017), each assessed under 11 distinct operating scenarios, yielding 1,749 graph instances (mean 30 nodes, 32 edges). Critical edges are a minority class; node features encode substation load and voltage, edge features encode cable impedance, nominal current, and loading state.

Models and training. Both models use 15 message-passing layers, hidden dimension 64, and are trained with focal loss (Lin et al., 2017) and Adam with learning rate 10^{-3}. Results are averaged over 10 random splits of the 159 grid topologies (80/20 train/test). We set $m = d = d_e = 64$, giving both models the same representational budget per endpoint. The asymmetric MLP takes input of dimension $64 + 128 = 192$; the symmetric MLP takes $64 + 64 = 128$, reducing the first-layer width by one third. The asymmetric edge MLP has layers $192 \to 128 \to 64$; the symmetric model applies $\phi : 64 \to 64$ to each endpoint, sums, then applies $\rho : 128 \to 128 \to 64$. All other choices (node update MLP, readout, batch normalisation, learning rate schedule) are identical.

Results. The symmetric variant achieves higher recall with strictly lower variance across random splits and one-third fewer parameters in the edge update network. A paired Wilcoxon signed-rank test on recall across the 10 splits gives $p < 0.05$. Balanced accuracy and F1 show the same ordering, with symmetric outperforming asymmetric on every metric.

The synthetic experiment (Section 4, Appendix E) isolates the three theoretical predictions under controlled conditions where the ground truth is perfectly symmetric and no confounding factors are present. The power grid experiment demonstrates that the same mechanism operates in a setting with real-world noise, class imbalance, and complex graph topology: the

Table 3. Edge classification on test set (mean $\pm$ std, 10 splits).

Model	Recall	Bal. Acc.	F1
Asymmetric	0.872 ± 0.014	0.886 ± 0.016	0.872 ± 0.010
Symmetric	$\mathbf{0.881 \pm 0.011}$	$\mathbf{0.891 \pm 0.013}$	$\mathbf{0.879 \pm 0.010}$

Table 4. Parameters in the edge update network ($d = d_e = m = 64$).

Model	Edge update params	Reduction
Asymmetric	24,640	–
Symmetric	16,448	-33%

theoretical benefit (orbit collapse) translates into measurable operational benefits (lower variance, fewer parameters, higher recall) even when other sources of variation are present.

The recall improvement, while statistically significant across the 10 splits, is modest relative to the total recall achieved by both models. The main contribution of the symmetric parameterisation in this domain is therefore not a dramatic performance jump but a better parameterisation with measurable efficiency gains. Whether the principle generalises to other edge-centric GNN applications (link prediction, molecular property prediction, traffic forecasting) remains to be tested; we expect the variance and parameter-efficiency gains to be consistent across tasks, while the magnitude of the recall improvement depends on how much of the original asymmetric model's capacity was being spent on redundantly encoding endpoint symmetry.

G. Code

The full experiment is available at `https://github.com/charlottecvn/icml-wss-GNNsymmetry`.