INVARIANT GRAPH PROPERTIES FOR NEURAL COMPUTATION GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Representing neural networks as graphs allows their architecture and learned structure to be studied using tools from network science, topological data analysis, discrete geometry, and graph representation learning. Yet the graph associated with a trained network is not uniquely determined by the function it computes: exact parameter symmetries can alter edge weights and graph structure without changing the network's predictions. We study invariance under common symmetries, including hidden permutations, positive rescaling, and general linear changes of basis. For positive rescaling, we identify minimal coordinates and a canonical representative on each fixed support, and show why zero weights preclude a single continuous exact representation across supports. For general linear symmetries, we derive operator representations that remain complete when ranks drop, including an operator graph for multi-head attention. We then connect this symmetry dependence to the reliability of predictions and interventions. Experiments on collections of MLP and ViT checkpoints show that raw graph analyses can differ substantially across equivalent parameterizations, whereas symmetry-aware methods remain stable and preserve predictions and transported interventions. The appropriate graph representation therefore depends on both the parameter symmetries and the target of the analysis.

1 INTRODUCTION

Graph representations of neural networks encode intermediate states or operations as vertices, trainable maps as edges, and checkpoint parameters as edge weights and other attributes. These representations support architecture prediction and structural analysis (You et al., 2020; Kadlecová et al., 2024), persistent and geometric diagnostics (Rieck et al., 2019; Girrbach et al., 2023; Tan et al., 2024), pruning and circuit analysis (Tan et al., 2026), and learned graph encoders (Lim et al., 2024; Kofinas et al., 2024).

The graph representation is not uniquely determined by the function the network computes. Relabeling hidden coordinates is a hidden permutation. Multiplying a ReLU unit's incoming weights and bias by $r > 0$ while dividing its outgoing weights by r is a positive hidden-unit rescaling. Factorized maps and attention heads also admit an invertible general linear gauge that changes the latent basis. These transformations preserve every input–output value but can change raw graph attributes. We call a change of representative within a declared symmetry orbit an *equivalence shift*. An equivalence shift tests whether a graph analysis reflects the network's computation or merely its particular parameterization: the inputs and predictions remain unchanged, while the analysis may assign different importance to parts of the network or suggest different modifications. Good performance on ordinary checkpoints alone does not resolve this ambiguity.

For a given graph construction and symmetry group, we determine which graph properties are invariant, the smallest complete invariant representation, and which parts of the representation persist when a weight vanishes or a factor loses rank. Our goal is to remove this representative dependence while retaining complete information about the symmetry orbit. We use different graphs to analyze symmetry in different architectures: Typed graph isomorphism handles permutations, orthogonal projection of log magnitudes handles positive rescaling, and operator products with image and kernel subspaces handle dense changes of basis. An analysis intended to describe the represented function should therefore use an invariant graph, or report how much its output changes along the orbit.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

How can we represent neural networks as graphs that are invariant under parameter symmetries, yet capable of distinguishing parameters in different orbits?

Contributions Our main contributions are:

1. We formalize invariance, equivariance, and orbit completeness for neural graph representations, and use this framework to analyze commonly used graph descriptors.
2. For positive rescaling, we characterize the minimal Euclidean quotient on each fixed support, derive a canonical representative and the full polynomial invariant ring, and identify the obstruction introduced by zero weights.
3. For permutation and general linear symmetries, we construct complete graph and operator representations, including rank-deficient attention, and extend the local-frame construction to LayerNorm, RMSNorm, and BatchNorm.
4. We connect dependence on parameterization to prediction and intervention error, and evaluate the resulting symmetry-aware methods on independently trained collections of MLP and ViT checkpoints.

2 Related work

Neural networks as graphs. Architecture graphs have been analyzed using structural features such as path length, clustering, and operation counts to characterize architecture families and predict performance (You et al., 2020; Kadlecová et al., 2024). Weight- and activation-dependent constructions include persistent topology (Rieck et al., 2019; Girrbach et al., 2023), activation graphs (Gebhart et al., 2019; Arnal et al., 2023), and curvature for robustness or pruning (Tan et al., 2024; 2026). Related work also studies the geometry of learned representations through graphs constructed from layerwise activations (Hehl et al., 2026). Graph Metanetworks and related methods instead use a neural computation graph as input to a learned graph encoder (Lim et al., 2024; Kofinas et al., 2024). We study how these analyses behave under exact function-preserving reparameterizations.

Parameter symmetries and learning on weights. Classical identifiability results already established uniqueness up to finite symmetries in shallow tanh or sigmoidal networks (Sussmann, 1992; Fefferman, 1994; Kůrková & Kainen, 1994). More recent work analyzes permutation, scaling, and activation-dependent symmetry groups through identifiability, intertwiner groups, and loss geometry (Phuong & Lampert, 2020; Godfrey et al., 2022; Grigsby et al., 2023; Zhao et al., 2026). Permutation alignment is now used in mode connectivity and model merging (Ainsworth et al., 2023). Weight-space networks explicitly encode such symmetries. Permutation-equivariant neural functionals, Neural Functional Transformers, Graph Metanetworks, ScaleGMN, Monomial-NFN, and transformer-specific NFNs cover increasingly broad architecture classes and group actions (Navon et al., 2023; Zhou et al., 2023a;b; Lim et al., 2024; Kofinas et al., 2024; Kalogeropoulos et al., 2024; Tran et al., 2024; 2025). Our quotient constructions provide a complementary route and can be supplied to hand-designed descriptors or generic graph encoders.

3 Neural computation graphs and parameter symmetries

We construct two representations of the same computation. The block representation keeps each state space and linear map intact. Choosing a basis expands each block into coordinate vertices and scalar-weight edges, which is the input used by most graph descriptors. Positive rescaling preserves these coordinate axes; a dense basis change does not. This distinction determines which representation is intrinsic.

3.1 Computation graphs and graph representations

A *typed feed-forward architecture* is a finite directed acyclic multigraph $Q = (V, E, s, t)$. Each vertex v carries a finite-dimensional state space H_v and, except at inputs, an activation $\rho_v : H_v \to H_v$. Each edge e carries a linear map $W_e : H_{s(e)} \to H_{t(e)}$, and each non-input vertex has a bias $b_v \in H_v$.

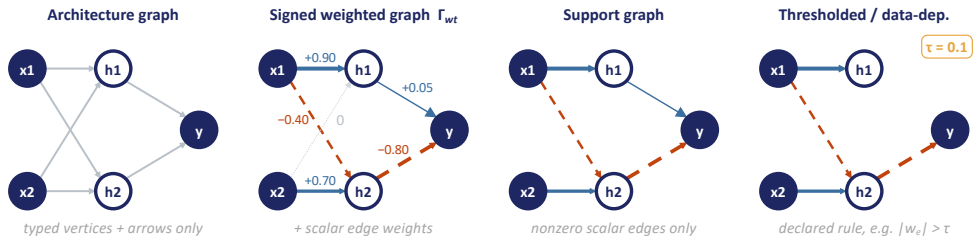


Figure 1: Different graph constructions of a ReLU network: architecture graph, weighted graph Γ_{wt}, support graph, and thresholded graph ($|w_e| > \tau = 0.1$).

The network evaluates the graph in topological order,

$$z_v = b_v + \sum_{e:t(e)=v} W_e h_{s(e)}, \qquad h_v = \rho_v(z_v), \qquad F_\theta(x) = (h_v)_{v \in V_{\mathrm{out}}}.$$

We represent each bias by an edge from a fixed source $\star$ with $h_\star \equiv 1$. Inputs, outputs, and $\star$ form the fixed boundary. After bases are chosen, each basis coordinate becomes a vertex and each matrix entry becomes a typed scalar edge with weight w_e.

We use four graph constructions (Figure 1). The *architecture graph* contains only the typed vertices and arrows. The *signed weighted graph* $\Gamma_{\mathrm{wt}}(\theta)$ adds the scalar edge weights. The *support graph* keeps exactly the nonzero scalar edges. A *thresholded or data-dependent graph* applies a declared rule that may use a threshold, input, dataset, or loss. Section 4 and the ReLU experiments use the signed weighted graph and its support. Section 5 and the transformer experiment use block or operator attributes.

3.2 Parameter symmetries and invariant properties

The full set of functionally equivalent parameters may be larger than the symmetry group under study. We therefore fix a declared group action $\mathcal{G} \curvearrowright \Theta$ satisfying $F_{g\theta} = F_\theta$ for every $g \in \mathcal{G}$. A broad class comes from hidden coordinate changes M_v that commute with the local activation. With boundary frames fixed,

$$(g \cdot W)_e = M_{t(e)} W_e M_{s(e)}^{-1}, \qquad (g \cdot b)_v = M_v b_v.$$

Hidden permutations, positive rescalings of degree-one homogeneous units, sign changes for odd activations, and changes of basis in linear latent spaces are standard examples (Phuong & Lampert, 2020; Navon et al., 2023; Zhao et al., 2026). We illustrate them in Figure 2. Appendix C.3 extends the rule to LayerNorm, RMSNorm, and BatchNorm.

For a declared graph construction Γ, a scalar property Φ is $\mathcal{G}$-invariant when

$$\Phi(\Gamma(g\theta)) = \Phi(\Gamma(\theta)) \qquad \text{for every } g, \theta.$$

Structured outputs such as nodes, edges, masks, or circuits should instead transform equivariantly. An invariant map I is *orbit complete* when $I(\theta) = I(\theta')$ if and only if $\theta' \in \mathcal{G} \cdot \theta$.

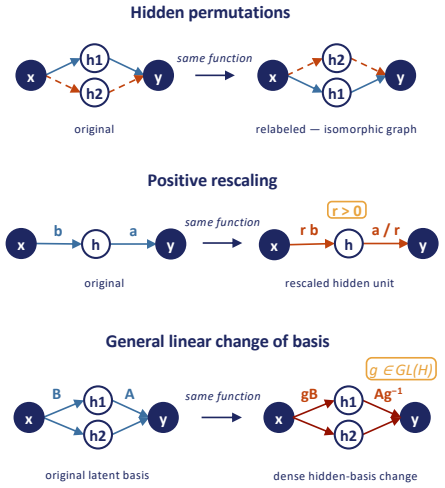


Figure 2: Three common symmetries: a permutation relabels hidden units; rescaling by $r > 0$ changes weights but preserves ab; a basis change $g \in \mathrm{GL}(H)$ changes A, B but preserves AB.

The hidden-permutation group $P = \prod_{v \in V_{\mathrm{hid}}} \mathfrak{S}_{d_v}$ only relabels coordinate vertices. Its orbits are exactly the typed weighted graph isomorphism classes under maps that fix the boundary and preserve parent-block and parent-edge types (Proposition C.1).

3.3 TRANSFORMATION RULES AND NEURAL-GRAPH DESCRIPTORS

Positive rescaling is simplest in log magnitude. Let $r_v > 0$ be the scale of hidden vertex v, let $\alpha_v := \log r_v$, and let $y_e := \log |w_e|$. On a fixed support S,

$$y' = y + B_S^{\mathsf{T}} \alpha, \tag{1}$$

where B_S is the hidden-node incidence matrix and $(B_S^{\mathsf{T}}\alpha)_e = \alpha_{t(e)} - \alpha_{s(e)}$. A log-linear quantity $c^{\mathsf{T}} y$ is therefore invariant exactly when $B_S c = 0$. Boundary-to-boundary path products satisfy this condition because the hidden-node factors cancel. The two-edge network $f_{a,b}(x) = a\,\mathrm{ReLU}(bx)$ gives a simple example. The transformation $(a, b) \mapsto (a/r, rb)$ changes both edge magnitudes but preserves ab and $\log |a| + \log |b|$. Raw weighted degree has no such cancellation and generally changes along the orbit.

Dense gauges mix scalar coordinates, so individual entries and scalar support are no longer intrinsic; operator products and subspaces take their place. Appendix E.2 derives the remaining transformation rules. Appendix F then applies them to centrality, spectra, curvature, persistence, activation graphs, pruning, and learned encoders.

4 COMPLETE INVARIANTS FOR POSITIVE RESCALING

Equation (1) makes the rescaling directions explicit. We first remove them on a fixed support, then show why no single continuous exact representation can retain all orbit information when weights cross zero. Appendix B.2 gives the associated sensitivity and repair results.

Fix a support S and sign pattern σ, and let $\Omega_{S,\sigma} := \{w : \mathrm{supp}(w) = S,\ \mathrm{sgn}(w_S) = \sigma\}$. Let $\mathcal{D}_+ := (\mathbb{R}_{>0})^{V_H}$ be the positive hidden-rescaling group. Let B_S be the hidden-node incidence matrix and define

$$q_S := |S| - \operatorname{rank} B_S, \qquad \Pi_S := I - B_S^{\mathsf{T}}(B_S B_S^{\mathsf{T}})^{\dagger} B_S, \qquad T_S := I - \Pi_S. \tag{2}$$

Here Π_S projects onto $\ker B_S$, the intrinsic log-weight coordinates, while T_S projects onto $\operatorname{im} B_S^{\mathsf{T}}$, the pure rescaling directions. A support is *boundary complete* when every supported edge lies on a directed path between fixed boundary vertices.

4.1 THE QUOTIENT ON A FIXED SUPPORT

Prior work introduces invariants – path products, path norms, and path lifting (Neyshabur et al., 2015; Meng et al., 2019; Stock & Gribonval, 2023) – and a preferred representative, the ordinary-ℓ_2 balanced point of Equi-normalization (Stock et al., 2019), but metric on orbits, minimal coordinate count, and explicit projector remain unknown. Theorem 4.1 identifies this orbit space as a log-Euclidean quotient and gives its canonical representative, computable by one sparse Laplacian solve. Corollary 4.2 gives the minimal coordinate count and a polynomial-size family from which complete path coordinates can be selected.

Theorem 4.1 (Fixed-support rescaling quotient)**.** *On $\Omega_{S,\sigma}$:*

(a) ***Quotient coordinates.*** *With*

$$d_Q([w], [w']) := \inf_{\alpha} \| \log |w'_S| - \log |w_S| - B_S^{\mathsf{T}}\alpha \|_2, \tag{3}$$

the map $[w] \mapsto \Pi_S \log |w_S|$ is an isometry from $\Omega_{S,\sigma}/\mathcal{D}_+$ onto $\ker B_S \cong \mathbb{R}^{q_S}$.

(b) ***Canonical representative.*** *Each orbit contains a unique log-Euclidean minimum-norm point $\mathrm{Can}_+(w)$, given by*

$$\log |\,\mathrm{Can}_+(w)_S| = \Pi_S \log |w_S|, \qquad \mathrm{sgn}(\mathrm{Can}_+(w)_S) = \sigma, \tag{4}$$

and equivalently characterized by $B_S \log |\,\mathrm{Can}_+(w)_S| = 0$.

Proof sketch. The log-space orbits are affine translates of $\operatorname{im} B_S^{\mathsf{T}}$. Orthogonal decomposition gives $\mathbb{R}^S = \operatorname{im} B_S^{\mathsf{T}} \oplus \ker B_S$, which yields the isometry and canonical representative. Appendix B.1 gives the full argument and the dimension-minimality proof used below.

The projector removes exactly the part of the log weights generated by hidden rescaling. The canonical representative is the unique orbit point whose incoming and outgoing log magnitudes balance at every hidden vertex.

Consequently, any typed graph encoder H becomes rescaling invariant after canonicalization:

$$w \longmapsto H(\Gamma_{\mathrm{wt}}(\mathrm{Can}_+(w)))\,. \tag{5}$$

Appendix B.2 gives the full orbit-completeness statement and further consequences.

The graph structure also determines the quotient dimension. Let $c_0(S)$ count weak components of $(V_H \cup V_B, S)$ that contain no boundary vertex, including isolated hidden vertices.

Corollary 4.2 (Quotient dimension and path coordinates)**.** *The quotient has dimension*

$$q_S = |S| - |V_H| + c_0(S), \tag{6}$$

and no continuous orbit-complete map into $\mathbb{R}^k$ exists for $k < q_S$. If S is boundary complete, then

$$\ker B_S = \mathrm{span}_{\mathbb{R}}\{\chi_\gamma : \gamma \textit{ is a boundary path in } S\}, \tag{7}$$

and q_S path-product magnitudes together with the fixed edge signs form a complete invariant.

Proof sketch. The rank formula follows from the reduced incidence matrix, and invariance of domain gives the coordinate lower bound. Vectors in $\ker B_S$ are conserved edge flows. On an acyclic boundary-complete graph, every nonnegative conserved flow decomposes into boundary paths; taking differences spans the full kernel. Appendix B.1 gives the complete proof and a polynomial-size candidate family from which a basis can be selected.

A small path basis therefore represents the quotient even when the network contains exponentially many complete paths.

4.2 Zero weights

Log coordinates require a fixed nonzero support, but pruning, masking, and thresholding all create zero weights. Path products are the classical remedy, polynomial invariants defined for every weight vector (Neyshabur et al., 2015; Meng et al., 2019; Stock & Gribonval, 2023), and for torus actions such as $\mathcal{D}_+$, invariant theory describes the invariant ring and the closed orbits in general (Sturmfels, 1996; Mumford et al., 1994; Kempf & Ness, 1979). Still, neither answers the two questions that matter for neural graphs: which invariants extend through zero weights, and whether exact coordinates can be continuous across supports.

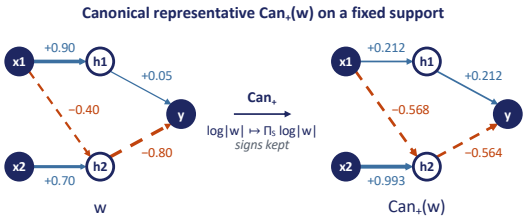


Figure 3: The canonical representative $\mathrm{Can}^+(w)$ projects $\log|w|$ onto $\ker B_S$, keeping signs. Incoming and outgoing magnitudes balance at every hidden unit (0.212=0.212; 0.568×0.993=0.564).

For a boundary path γ, write $p_\gamma(w) = \prod_{e\in\gamma} w_e$.

Theorem 4.3 (Global polynomial invariants)**.** *For the untied scalar-edge parameterization,*

$$\mathbb{R}[w_e]^{\mathcal{D}_+} = \mathbb{R}[p_\gamma : \gamma \textit{ is a boundary path}]. \tag{8}$$

The path monomials are the indecomposable invariant monomials and form a minimal algebra-generating family.

Proof sketch. An invariant monomial exponent is a nonnegative integer flow conserved at every hidden vertex. Acyclic flow decomposition writes it as a sum of boundary paths.

Every polynomial invariant is therefore a polynomial in boundary-path products. On a fixed nonzero support, far fewer continuous coordinates may suffice. A bias-free scalar-input, scalar-output, width-two network with L hidden layers has $2L$ quotient dimensions but 2^L boundary paths.

For a support S, let S_{bb} contain the edges lying on boundary paths within S, and let w_{bb} retain those edges and set the rest to zero. We obtain a simple graph criterion and an explicit limiting support from closed-orbit principles for torus actions (Mumford et al., 1994; Kempf & Ness, 1979).

Theorem 4.4 (Closed-orbit core and zero-weight barrier)**.** (a) *The orbit of w is closed if and only if its support is boundary complete.*

(b) *Every orbit closure contains the unique closed orbit of w_{bb}.*

(c) *If the rescaling action is nontrivial, the exact orbit space across all supports is non-Hausdorff; no continuous Hausdorff-valued invariant separates all exact orbits through zero-weight transitions.*

Proof sketch. Boundary completeness is equivalent to the existence of a strictly positive conserved flow. When it fails, a theorem of alternatives produces a hidden potential whose one-parameter rescaling contracts every noncore edge. Invariant path products force any closed limit to retain exactly the boundary-path core. Appendix B.3 contains the proofs, and Appendix B.4 gives the functional-completeness consequence on identifiable loci.

5 General linear symmetries

A dense linear gauge changes the coordinate axes themselves, so scalar-edge attributes are no longer intrinsic. Operator products and image and kernel subspaces remain complete at rank loss. Unlike positive-rescaling canonicalization, this changes the representation rather than only its edge attributes.

5.1 A two-factor linear gauge

Let B map an input space X into a latent space H, and let A map H to an output space Y. Changing the basis of H by $g \in \mathrm{GL}(H)$ gives $(A, B) \mapsto (Ag^{-1}, gB), g \in \mathrm{GL}(H)$. The product AB is always invariant. The theorem states exactly what must be added when a factor loses rank.

Theorem 5.1 (Orbit classification for a two-factor gauge)**.** *Two pairs (A, B) and (A', B') lie in the same $\mathrm{GL}(H)$-orbit if and only if*

$$AB = A'B', \qquad \operatorname{im} A = \operatorname{im} A', \qquad \ker B = \ker B'. \tag{9}$$

Hence $(AB, \operatorname{im} A, \ker B)$ is orbit complete at every rank.

Proof sketch. Equality of $\ker B$ identifies the reachable hidden subspaces through $Bx \mapsto B'x$. Equality of AB makes this map intertwine A and A', and equality of the output images extends it to an invertible map on all of H.

The singular behavior is captured by a short corollary. A *minimal factorization* is one for which both factor ranks equal $\operatorname{rank}(AB)$.

Corollary 5.2 (Closed orbits of the factorization gauge)**.** *The orbit of (A, B) is closed if and only if $\operatorname{rank} A = \operatorname{rank} B = \operatorname{rank}(AB)$. Every orbit closure contains the unique closed orbit of a minimal factorization of AB. Every continuous Hausdorff-valued invariant is constant on each fiber of the product map.*

Proof sketch. A four-part decomposition of the latent space separates directions transmitted by both factors, reached but annihilated, observed but unreached, and unused. Rescaling the middle two classes produces the minimal factorization in the orbit closure.

When both factors have full latent rank, AB determines the remaining two spaces. At rank loss, the same product can come from factors with different output images or input kernels, so the product alone is incomplete.

5.2 Operator graphs for attention

Each attention head contains two instances of the same factorization pattern. The query–key product determines attention scores, and the value–output product determines the head contribution. For each head, define

$$\mathcal{I}_h^{QK} := (M_h^{QK}, \operatorname{im} W_h^Q, \ker(W_h^K)^\mathsf{T}), \qquad M_h^{QK} := W_h^Q (W_h^K)^\mathsf{T},$$
$$\mathcal{I}_h^{VO} := (M_h^{VO}, \operatorname{im} W_h^V, \ker W_h^O), \qquad M_h^{VO} := W_h^V W_h^O.$$

The within-head gauges are

$$W_h^Q \mapsto W_h^Q G_h, \quad W_h^K \mapsto W_h^K G_h^{-\mathsf{T}}, \quad W_h^V \mapsto W_h^V H_h, \quad W_h^O \mapsto H_h^{-1} W_h^O, \tag{10}$$

followed by head permutation.

Corollary 5.3 (Complete attention operator graph)**.** *The unordered head graph with attributes $(\mathcal{I}_h^{QK}, \mathcal{I}_h^{VO})$ is complete for the declared within-head gauges and head permutations. On the full-latent-rank locus*

$$\operatorname{rank} W_h^Q = \operatorname{rank} W_h^K = r_q, \qquad \operatorname{rank} W_h^V = \operatorname{rank} W_h^O = r_v, \tag{11}$$

the products (M_h^{QK}, M_h^{VO}) suffice. On the generic identifiable locus of standard attention, this product-only graph is also complete for the represented attention function (Tran et al., 2026).

Proof sketch. Apply Theorem 5.1 independently to the query–key and value–output factorizations, then quotient the finite head permutation. On the full-rank locus, each product already determines the image and kernel data.

Query, key, value, and output coordinates depend on the chosen latent bases, but the operator products and the required image/kernel data do not. Fixed latent operators such as RoPE restrict the gauge to their centralizer. Appendix C.2 derives the corresponding complete operator contractions.

The within-head gauges cancel inside each attention head and do not cross LayerNorm (Appendix C.3). Section 7.2 uses the operator products to transport whole-head interventions between equivalent representatives of trained transformers.

6 Reliability consequences and invariantization

We separate graph analysis into two stages. The map ϕ constructs a graph or descriptor from parameters, and h turns it into the reported output. For prediction, h is a regressor. For pruning or circuit analysis, h selects an object that is then applied to the model. A change in ϕ is harmless only if the complete pipeline has the required invariance or equivariance.

Let G be the declared symmetry group, let d be a metric on $\mathcal{Y}$, and let $p = h \circ \phi : \Theta \to \mathcal{Y}$ be a graph-based predictor with target $\tau : \Theta \to \mathcal{Y}$. For any $f : \Theta \to \mathcal{Y}$, define

$$\operatorname{diam}_G f(\theta) := \sup_{g_1, g_2 \in G} d(f(g_1\theta), f(g_2\theta)),$$
$$\operatorname{err}_G(p, \tau; \theta) := \sup_{g \in G} d(p(g\theta), \tau(g\theta)).$$

Proposition 6.1 (Orbit-diameter mismatch bound)**.** *Whenever the diameters are finite,*

$$\operatorname{err}_G(p, \tau; \theta) \geq \frac{1}{2} \left| \operatorname{diam}_G p(\theta) - \operatorname{diam}_G \tau(\theta) \right|. \tag{12}$$

In particular, if τ is G-invariant, then

$$\sup_g d(p(g\theta), \tau(\theta)) \geq \tfrac{1}{2} \operatorname{diam}_G p(\theta). \tag{13}$$

Any observed prediction diameter therefore certifies a representative with substantial error against an invariant target. Appendix D gives the symmetric inequalities and a random-shift bias–variance decomposition.

6.1 Selections and interventions

A mask or circuit must move with the parameter representative before it is applied. Let $\mathcal{M}$ denote the intervention space. For $m \in \mathcal{M}$, let $\mathcal{I}_m(\theta)$ be the model obtained by applying m to parameters θ. Assume that G also acts on $\mathcal{M}$ and on intervened models.

Proposition 6.2 (Equivariant selection gives equivalent interventions)**.** *Suppose the intervention rule and selector satisfy $\mathcal{I}_{gm}(g\theta) = g\mathcal{I}_m(\theta)$ and $S(g\theta) = gS(\theta)$. Then $\mathcal{I}_{S(g\theta)}(g\theta) = g\mathcal{I}_{S(\theta)}(\theta)$. If the action on intervened models preserves realization, the two interventions compute the same function.*

Under positive rescaling, edge identities are fixed, so invariant scores produce the same mask. Under a dense gauge, coordinate deletion is not intrinsic; an intervention should act on a subspace, projector, or operator component.

6.2 Choosing a representation

The target determines the appropriate representation. For a target that depends only on the represented function, one can use invariant attributes, canonicalize before applying an existing method, or build the symmetry into an equivariant model. For a target that depends on implementation, such as latency, conditioning, compression, or future optimization, the relevant gauge-sensitive information should be retained and the parameterization should be declared. Appendix G.6 illustrates this distinction by resuming optimization from equivalent ViT checkpoints.

7 Experiments

To test whether an equivalence shift alters a graph-based prediction or intervention, we study performance prediction for ReLU MLPs and head matching followed by ablation for trained transformers. Appendix G gives implementation details and additional experiments.

7.1 MLP prediction under rescaling

We train 80 ReLU MLPs with three 32-unit hidden layers on Fashion-MNIST. We retain ten checkpoints per run and split entire runs into 48 training, 16 validation, and 16 test runs. The same graph encoder architecture predicts each checkpoint's test negative log-likelihood (NLL) from raw weight signs and log magnitudes, canonical log magnitudes $\Pi_S \log |w_S|$ with the same signs, or raw graphs with random rescaling during predictor training.

We measure root mean squared error (RMSE) on untransformed checkpoints (IID), 32 random rescalings per checkpoint, and rescalings chosen to increase prediction error. The predeclared primary comparison uses hidden-unit log scales α with $\|\alpha\|_2 \le R = 2$. The search takes 60 projected Adam steps from each of four random starts. Its errors are observed lower bounds on the worst-case error, not certified maxima. Rescaling preserves probe logits to relative error below 10^{-9}.

Results. Raw and canonical predictors have similar IID RMSEs, 0.0269 and 0.0260, compared with 0.0685 for the mean baseline. The search raises raw RMSE to 0.2156, while canonical RMSE remains 0.0260. Defining the shift penalty as the increase over IID RMSE, the paired run-level difference between raw and canonical is $\Delta_{\text{shift}} = 0.1887$, with 95% CI $= [0.1676, 0.2072]$. We compute confidence intervals by resampling entire test runs. Figure 4 shows results across radii. Appendix G.1 reports larger search budgets and results using only final checkpoints.

Raw predictions also change without an optimized rescaling: random rescaling gives RMSE 0.0412, and canonicalizing the input to the trained raw predictor changes its output by 0.0530 on average. Augmentation reduces error under random rescaling but remains sensitive to the search (Figure 4). Appendix G.1 gives full numerical results. A predictor using only the rescaling-dependent component reaches IID $R^2 = 0.753$. Scales that can change without altering the function still predict NLL in this collection. Low error on the original checkpoints therefore does not ensure consistent predictions after exact rescaling.

7.2 Circuit transport on trained transformers

We match attention heads between a checkpoint and transformed copies of it, then ablate the matched heads. Correct matching will preserve the intervention's effect.

We fine-tune 40 ViT-B/16 models on CIFAR-100 for eight epochs. Runs are split into 24 training, 4 validation, and 12 test runs. The architecture includes residual connections, MLP blocks, and LayerNorm. For each test model, we rank heads by the NLL change caused by their ablation on 1,024 calibration examples and select the most-important 10%. We evaluate these ablations and 16 size-matched random controls on 2,048 disjoint examples.

For each test checkpoint, we apply head permutation, scalar rescaling, orthogonal changes of basis, and general-linear changes of basis with condition numbers 10, 100, and 300. Before ablation, all transformations pass the relative logit error threshold of 5×10^{-5}. GL-100 is the primary comparison.

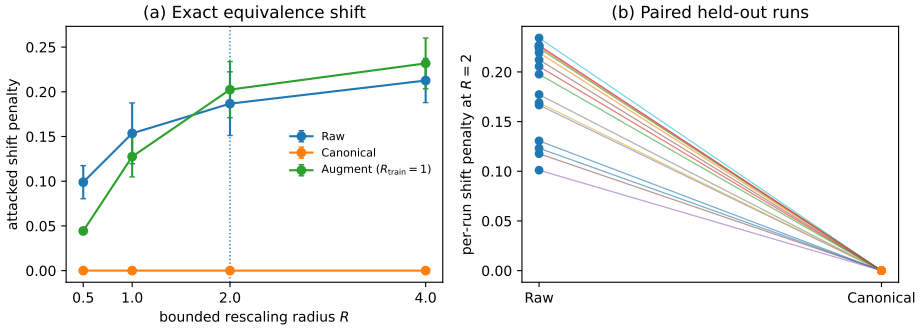


Figure 4: MLP prediction after exact rescaling. The shift penalty is attacked RMSE minus RMSE on untransformed checkpoints. **Left:** penalty versus rescaling radius; error bars show standard deviation over three predictor initializations. **Right:** paired penalties for held-out runs at $R = 2$, averaged over checkpoints and predictor initializations.

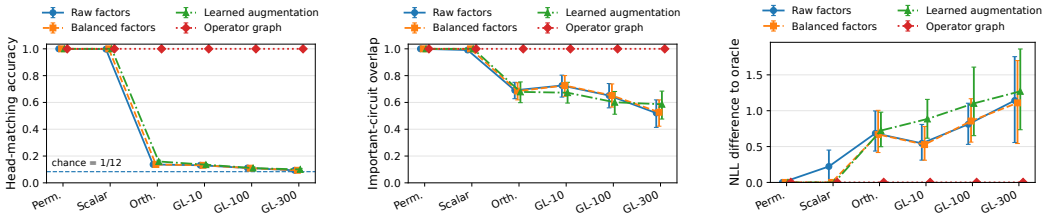


Figure 5: Matching heads and transporting ablations between equivalent ViT-B/16 checkpoints. **Left:** head-matching accuracy; the dashed line is chance. **Middle:** overlap of the most-important 10% head set. **Right:** absolute NLL difference between ablations using the inferred and oracle assignments. Error bars are 95% confidence intervals clustered by training run.

Within each layer, the Hungarian algorithm assigns each original head to one head in the transformed model. We compare compact summaries (sketches) of the raw head matrices, factors after scalar balancing, a Siamese matcher trained with transformed checkpoints from the training/validation runs, and the query–key and value–output operator products from Section 5.2. Query, key, and value biases are included through a constant input coordinate. The known head permutation gives the correct assignment (oracle), used only for evaluation.

Results. At GL-100, raw head-matching accuracy is 0.109 (95% CI $[0.098, 0.119]$), close to chance 0.083; balancing and the learned matcher do not improve it (Figure 5; Appendix G.3). Ablation using the raw matching has an absolute NLL difference of 0.811 from oracle ablation. Operator products match every head correctly on every held-out run and in every condition (Figure 5). Zero error means agreement with oracle transport, not exact floating-point equality between checkpoints. Ablating the most-important 10% raises NLL by 2.733, versus 0.336 for size-matched random sets (Appendix G.3).

8 Conclusion

We characterize complete graph invariants for the studied parameter symmetries and connect them to prediction and intervention reliability. Our findings suggest that equivalence-shift tests should become a standard control for analyses intended to describe a network's function. Future work could identify circuits as subspaces or operator components whose intervention effects survive allowed changes of basis. Another direction is to develop compact invariant representations that remain stable as weights vanish or ranks change. The broader goal is to distinguish properties of computation from choices of parameterization.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in writing proofs, providing feedback on experiment designs, coding, gathering past work that uses graph descriptors in neural network analysis, and improving writing in general. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

Marco Armenta, Thomas Brüstle, Souheila Hassoun, and Markus Reineke. Double framed moduli spaces of quiver representations. *Linear Algebra and its Applications*, 650:98–131, 2022. doi: 10.1016/j.laa.2022.05.018.

Charles Arnal, Felix Hensel, Mathieu Carrière, Théo Lacombe, Hiroaki Kurihara, Yuichi Ike, and Frédéric Chazal. Magdiff: Covariate data set shift detection via activation graphs of deep neural networks. *arXiv preprint arXiv:2305.13271*, 2023.

Amir Asiaee. Causal mechanism reduction: Mechanism replacement for neural network pruning and abstraction. *arXiv preprint arXiv:2602.24266*, 2026.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. URL `https://arxiv.org/abs/1607.06450`.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37(1):103–120, 2007. doi: 10.1007/s00454-006-1276-5.

Ciprian A. Corneanu, Sergio Escalera, and Aleix M. Martinez. Computing the testing error without a testing set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2674–2682, 2020. doi: 10.1109/CVPR42600.2020.00275.

Charles Fefferman. Reconstructing a neural net from its output. *Revista Matemática Iberoamericana*, 10(3):507–555, 1994. doi: 10.4171/RMI/160.

Thomas Gebhart, Paul Schrater, and Alan Hylton. Characterizing the shape of activation space in deep neural networks. In *2019 18th IEEE International Conference on Machine Learning and Applications*, pp. 1537–1542. IEEE, 2019.

Leander Girrbach, Anders Christensen, Ole Winther, Zeynep Akata, and A. Sophia Koepke. Addressing caveats of neural persistence with deep graph persistence. *Transactions on Machine Learning Research*, 2023.

Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the symmetries of deep learning models and their internal representations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11893–11905, 2022.

J. Elisenda Grigsby, Kathryn Lindsey, and David Rolnick. Hidden symmetries of ReLU networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11734–11760. PMLR, 2023.

Moritz Hehl, Max von Renesse, and Melanie Weber. Neural feature geometry evolves as discrete ricci flow. In *Forty-third International Conference on Machine Learning*, 2026.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015.

Gabriela Kadlecová, Jovita Lukasik, Martin Pilát, Petra Vidnerová, Mahmoud Safari, Roman Neruda, and Frank Hutter. Surprisingly strong performance prediction with neural graph features. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22771–22816. PMLR, 2024.

Ioannis Kalogeropoulos, Giorgos Bouritsas, and Yannis Panagakis. Scale equivariant graph metanetworks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Abdullah Karasan. Representation measurements under function-preserving reparameterizations. *arXiv preprint arXiv:2608.27020*, 2026. URL https://arxiv.org/abs/2608.27020. Concurrent work.

George Kempf and Linda Ness. The length of vectors in representation spaces. In Knud Lønsted (ed.), *Algebraic Geometry*, volume 732 of *Lecture Notes in Mathematics*, pp. 233–243. Springer, Berlin, 1979. doi: 10.1007/BFb0066647.

James A King, Arshdeep Singh, and Mark D Plumbley. Compressing audio cnns with graph centrality based filter pruning. In *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, 2023.

Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J. Burghouts, Efstratios Gavves, Cees G. M. Snoek, and David W. Zhang. Graph neural networks for learning equivariant representations of neural networks. In *International Conference on Learning Representations*, 2024.

Věra Kůrková and Paul C. Kainen. Functionally equivalent feedforward neural networks. *Neural Computation*, 6(3):543–558, 1994. doi: 10.1162/neco.1994.6.3.543.

Théo Lacombe, Yuichi Ike, Mathieu Carrière, Frédéric Chazal, Marc Glisse, and Yuhei Umeda. Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 2666–2672. International Joint Conferences on Artificial Intelligence Organization, 2021. doi: 10.24963/ijcai.2021/367.

Lieven Le Bruyn and Claudio Procesi. Semisimple representations of quivers. *Transactions of the American Mathematical Society*, 317(2):585–598, 1990. doi: 10.2307/2001477.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Derek Lim, Haggai Maron, Marc T. Law, Jonathan Lorraine, and James Lucas. Graph metanetworks for processing diverse neural architectures. In *International Conference on Learning Representations*, 2024.

Qi Meng, Shuxin Zheng, Huishuai Zhang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, Nenghai Yu, and Tie-Yan Liu. $\mathcal{G}$-SGD: Optimizing ReLU neural networks in its positively scale-invariant space. In *International Conference on Learning Representations*, 2019.

Bamdev Mishra, Gilles Meyer, Silvère Bonnabel, and Rodolphe Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3–4):591–621, 2014. doi: 10.1007/s00180-013-0464-z.

David Mumford, John Fogarty, and Frances Kirwan. *Geometric Invariant Theory*, volume 34 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, Berlin, 3 edition, 1994. doi: 10.1007/978-3-642-57916-5.

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25790–25816. PMLR, 2023.

Behnam Neyshabur, Ruslan Salakhutdinov, and Nathan Srebro. Path-SGD: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Mary Phuong and Christoph H. Lampert. Functional vs. parametric equivalence of ReLU networks. In *International Conference on Learning Representations*, 2020.

Theo Moe Putterman, Derek Lim, Yoav Gelberg, Stefanie Jegelka, and Haggai Maron. Learning on LoRAs: GL-equivariant processing of low-rank weight spaces for large finetuned models. *arXiv preprint arXiv:2410.04207*, 2024.

Bastian Rieck, Matteo Togninalli, Christian Bock, Michael Moor, Max Horn, Thomas Gumbsch, and Karsten Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations*, 2019.

Tananun Songdechakraiwut and Yutong Wu. Functional connectomes of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20558–20566, 2025.

Erich Stiemke. Über positive lösungen homogener linearer gleichungen. *Mathematische Annalen*, 76:340–342, 1915. doi: 10.1007/BF01458147.

Pierre Stock and Rémi Gribonval. An embedding of ReLU networks and an analysis of their identifiability. *Constructive Approximation*, 57(2):853–899, 2023. doi: 10.1007/s00365-022-09578-1.

Pierre Stock, Benjamin Graham, Rémi Gribonval, and Hervé Jégou. Equi-normalization of neural networks. In *International Conference on Learning Representations*, 2019.

Bernd Sturmfels. *Gröbner Bases and Convex Polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, Providence, RI, 1996. doi: 10.1090/ulect/008.

Héctor J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input–output map. *Neural Networks*, 5(4):589–593, 1992. doi: 10.1016/S0893-6080(05)80037-1.

Edric Tam and David Dunson. Fiedler regularization: Learning neural networks with graph sparsity. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9346–9355. PMLR, 2020.

Shuhang Tan, Jayson Sia, Paul Bogdan, and Radoslav Ivanov. Analyzing neural network robustness using graph curvature. In *2024 International Conference on Assured Autonomy (ICAA)*, pp. 110–113. IEEE, 2024. doi: 10.1109/ICAA64256.2024.00026.

Shuhang Tan, Jayson Sia, Paul Bogdan, and Radoslav Ivanov. Post-training neural network pruning using graph curvature. *Transactions on Machine Learning Research*, 2026. URL `https://openreview.net/forum?id=kwACVY73Ug`.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Hoang V. Tran, Thieu Vo, An Nguyen The, Tho Tran Huu, Minh-Khoi Nguyen-Nhat, Thanh Tran, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant neural functional networks for transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

Viet-Hoang Tran, Thieu Vo, Tho Tran-Huu, An T Nguyen, and Tan Nguyen. Monomial matrix group equivariant neural functional networks. *Advances in Neural Information Processing Systems*, 37: 48628–48665, 2024.

Viet-Hoang Tran, Vinh Khanh Bui, Van-Hoan Trinh, Tan Lai Ngoc, and Tan M. Nguyen. Functional equivalence in attention: A comprehensive study with applications to linear mode connectivity. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. arXiv:2606.17830.

Fatemeh Vahedian, Ruiyu Li, Puja Trivedi, Di Jin, and Danai Koutra. Leveraging the graph structure of neural network training dynamics. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 4545–4549, 2022. doi: 10.1145/3511808.3557628.

Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10881–10891. PMLR, 2020.

Olivia T. Zahn, Thomas L. Daniel, and J. Nathan Kutz. Motif distribution and function of sparse deep neural networks. *arXiv preprint arXiv:2403.00974*, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Bo Zhao, Robin Walters, and Rose Yu. Symmetry in neural network parameter spaces. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=jLpWq5QY6I. First posted as arXiv:2506.13018.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J. Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. In *Advances in Neural Information Processing Systems*, volume 36, 2023a.

Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J. Zico Kolter, and Chelsea Finn. Neural functional transformers. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

A Additional related work

Rescaling invariants. Path products, path norms, and path lifting remove positive rescaling (Neyshabur et al., 2015; Meng et al., 2019; Stock & Gribonval, 2023). Equi-normalization selects an ordinary-ℓ_2 balanced representative (Stock et al., 2019). Our fixed-support construction instead gives a log-Euclidean quotient, its minimal continuous coordinate count, and an explicit projector. The polynomial-invariant and closed-orbit results specialize classical invariant theory to the network's boundary paths.

Learning on weights and low-rank factors. ScaleGMN and Monomial-NFN encode scaling and permutation symmetries in learned weight-space models (Kalogeropoulos et al., 2024; Tran et al., 2024). Learning on LoRAs uses canonicalization, invariant features, and general-linear equivariance for low-rank factors (Putterman et al., 2024). Fixed-rank factorization and quiver representations provide the underlying orbit geometry (Mishra et al., 2014; Le Bruyn & Procesi, 1990; Armenta et al., 2022). Our factor representation retains the image and kernel information needed to distinguish orbits at every rank.

Attention and normalization. Tran et al. (2026) study attention identifiability and the restrictions imposed by rotary positional operators. Appendix C.2 gives complete attributes for the corresponding full-rank factor problem. Appendix C.3 separately treats normalization, including the distinction between core normalization maps and explicit affine parameters.

Measurements and interventions. Asiaee (2026) compares pruning selections under exact rescaling and uses function-based scores to avoid dependence on parameter scale. Concurrent work tests representation measurements under function-preserving basis changes, including normalization-compatible orthogonal transformations (Karasan, 2026). We connect these questions to graph representations, complete invariants, and the transport of interventions.

B Proofs and supplementary results for Section 4

Throughout this appendix, B_S denotes the hidden-node incidence matrix on a scalar support S, and boundary-node potentials are fixed to zero.

The following two elementary results will be used repeatedly in later proofs.

Lemma B.1 (Acyclic flow decomposition)**.** *Let S be a finite directed acyclic graph with hidden vertices V_H and fixed boundary vertices V_B. If $c \in \mathbb{R}_+^S$ satisfies $B_S c = 0$, then there are boundary-to-boundary directed paths $\gamma_1, \dots, \gamma_N$ and coefficients $\lambda_i > 0$ such that*

$$c = \sum_{i=1}^{N} \lambda_i \chi_{\gamma_i}.$$

If c is integral, the coefficients may be chosen as positive integers.

Proof. Assume $c \neq 0$ and choose an edge of positive flow. Extend it backwards while its current tail is hidden. Conservation at a hidden tail with positive outgoing flow guarantees a positive-flow incoming edge. Since the graph is acyclic, this process terminates at a boundary vertex. Extending forwards analogously terminates at a boundary vertex and produces a boundary-to-boundary path γ whose edges all have positive flow. Let

$$\lambda := \min_{e \in \gamma} c_e > 0.$$

Then $c - \lambda \chi_\gamma$ is nonnegative, remains conserved at every hidden vertex, and has strictly fewer positive coordinates. Iteration terminates after finitely many steps. When c is integral, every chosen λ is integral. □

Lemma B.2 (Stiemke's theorem of alternatives (Stiemke, 1915))**.** *For every real matrix A, exactly one of the following systems has a solution:*

$$Ax = 0, \qquad x > 0; \tag{14}$$

$$A^{\mathsf{T}} y \geq 0, \qquad A^{\mathsf{T}} y \neq 0. \tag{15}$$

B.1 THE EXACT RESCALING QUOTIENT

Proof of Theorem 4.1. Fix the sign pattern and identify $\Omega_{S,\sigma}$ with $\mathbb{R}^S$ through the diffeomorphism $w \mapsto y = \log|w_S|$. Equation (1) shows that the rescaling orbits are precisely the affine spaces

$$y + \operatorname{im} B_S^{\mathsf{T}}. \tag{16}$$

The fundamental theorem of linear algebra gives the orthogonal decomposition

$$\mathbb{R}^S = \operatorname{im} B_S^{\mathsf{T}} \oplus \ker B_S.$$

The projector onto the second summand is

$$\Pi_S = I - B_S^{\mathsf{T}}(B_S B_S^{\mathsf{T}})^{\dagger} B_S.$$

Therefore $\Pi_S y = \Pi_S y'$ if and only if $y' - y \in \operatorname{im} B_S^{\mathsf{T}}$, proving that the map

$$[w] \longmapsto \Pi_S \log|w_S| \tag{17}$$

is well defined, injective on orbits, and surjective onto $\ker B_S$.

For $d := y' - y$, orthogonal projection also gives

$$\begin{aligned}\inf_{\alpha} \|d - B_S^{\mathsf{T}}\alpha\|_2 &= \operatorname{dist}(d, \operatorname{im} B_S^{\mathsf{T}}) \\ &= \|\Pi_S d\|_2.\end{aligned}$$

This is the Euclidean distance between $\Pi_S y$ and $\Pi_S y'$, so the quotient map is an isometry.

Let C_S have orthonormal columns spanning $\ker B_S$. The map $z \mapsto C_S z$ is an isometry from $\mathbb{R}^{q_S}$ onto $\ker B_S$, and $C_S C_S^{\mathsf{T}} = \Pi_S$. Hence $C_S^{\mathsf{T}} y$ is orbit complete. To prove minimality, let $J : \Omega_{S,\sigma} \to \mathbb{R}^k$ be continuous, invariant, and orbit complete. It induces a continuous injective map $\bar{J} : \mathbb{R}^{q_S} \to \mathbb{R}^k$. If $k < q_S$, compose with the standard inclusion $\iota : \mathbb{R}^k \hookrightarrow \mathbb{R}^{q_S}$. Brouwer's invariance-of-domain theorem would make the image of $\iota \circ \bar{J}$ open in $\mathbb{R}^{q_S}$, yet that image lies in the proper linear subspace $\iota(\mathbb{R}^k)$, a contradiction.

The point $\Pi_S y$ is the unique minimum-norm point in the affine space (16). Restoring the fixed signs proves $\log|\operatorname{Can}_+(w)_S| = \Pi_S \log|w_S|, \qquad \operatorname{sgn}(\operatorname{Can}_+(w)_S) = \sigma$. Since $\Pi_S y \in \ker B_S$, the canonical representative satisfies

$$B_S \log|\operatorname{Can}_+(w)_S| = 0. \tag{18}$$

Conversely, an orbit point satisfying (18) belongs to both $y + \operatorname{im} B_S^{\mathsf{T}}$ and $\ker B_S$, whose intersection contains exactly $\Pi_S y$. In coordinates, the vth balance equation is

$$\sum_{e:t(e)=v} \log|w_e| = \sum_{e:s(e)=v} \log|w_e|,$$

which is equivalent to equality of the products of incoming and outgoing magnitudes.

□

Proof of Corollary 4.2. A vector $\alpha \in \ker B_S^{\mathsf{T}}$ has equal values at the endpoints of every support edge, where boundary values are fixed at zero. Thus α is zero on every weakly connected support component meeting V_B and is constant on every component containing only hidden vertices. Therefore

$$\dim \ker B_S^{\mathsf{T}} = c_0(S), \qquad \operatorname{rank} B_S = |V_H| - c_0(S),$$

which proves (6).

Assume now that S is boundary complete. For every edge e, choose a boundary-to-boundary path γ_e containing it and define

$$c := \sum_{e \in S} \chi_{\gamma_e}.$$

Then $c > 0$ and $B_S c = 0$. Given any $x \in \ker B_S$, choose $M > 0$ so large that $x + Mc \geq 0$. Lemma B.1 decomposes both $x + Mc$ and Mc into nonnegative linear combinations of boundary

paths. Their difference expresses x as a real linear combination of path-incidence vectors, proving (7).

Choose q_S paths whose incidence vectors form a basis of $\ker B_S$. Equality of their path-product magnitudes for w and w' implies

$$\chi_\gamma^\mathsf{T}(y' - y) = 0$$

for a basis of $\ker B_S$. Hence $y' - y \in (\ker B_S)^\perp = \operatorname{im} B_S^\mathsf{T}$. Together with equality of the fixed edge-sign pattern, this proves that w and w' are rescaling equivalent. The converse is the telescoping identity (38).

Finally, choose for each hidden vertex v incident to an edge of S one boundary-to-v prefix P_v and one v-to-boundary suffix Q_v, and use empty prefixes or suffixes at boundary endpoints. Consider

$$\mathcal{R} := \{P_v Q_v : v \in V_H \text{ is incident to } S\} \cup \{P_{s(e)} e Q_{t(e)} : e \in S\}.$$

It contains at most $|V_H| + |S|$ paths. If δ is orthogonal to every incidence vector in $\mathcal{R}$, set

$$a_v := \sum_{e \in P_v} \delta_e, \qquad b_v := \sum_{e \in Q_v} \delta_e.$$

Orthogonality to $P_v Q_v$ gives $a_v + b_v = 0$, while orthogonality to $P_{s(e)} e Q_{t(e)}$ gives $a_{s(e)} + \delta_e + b_{t(e)} = 0$. Therefore

$$\delta_e = a_{t(e)} - a_{s(e)} = (B_S^\mathsf{T} a)_e,$$

with boundary potentials set to zero. Thus

$$\operatorname{span}\{\chi_\gamma : \gamma \in \mathcal{R}\}^\perp = \operatorname{im} B_S^\mathsf{T} = (\ker B_S)^\perp,$$

so the family spans $\ker B_S$. Gaussian elimination selects a basis from it. □

Computational cost. The incidence matrix has at most two nonzero entries per edge. Canonicalization therefore requires the sparse right-hand side $B_S \log |w_S|$, one solve with the hidden-node Laplacian $L_S := B_S B_S^\mathsf{T}$ on the orthogonal complement of its nullspace, and one sparse multiplication by B_S^T. For a fixed support, a sparse factorization or preconditioner for L_S is reused across checkpoints; the dense projector Π_S is never formed. Iterative Laplacian solvers provide an alternative for larger graphs. Appendix G.5 reports the timing study.

B.2 Canonical graph encoders and descriptor sensitivity

The orthogonal decomposition into invariant and rescaling directions gives

$$\|\Delta y\|_2^2 = d_Q([w], [w'])^2 + \|T_S \Delta y\|_2^2, \qquad \Delta y := \log |w'_S| - \log |w_S|. \tag{19}$$

Canonicalization can be used before any graph encoder. The next result states when the resulting representation is invariant and when it also distinguishes symmetry orbits. Write $\mathcal{E}_H(w) := H(\Gamma_{\mathrm{wt}}(\mathrm{Can}_+(w)))$.

Corollary B.3 (Canonicalized graph encoders)**.** *Let H be invariant under type-preserving attributed-graph isomorphism. On a fixed support-and-sign stratum, the map in* (5) *is positive-rescaling invariant. On a union of strata closed under hidden permutations, it is also permutation invariant. If H is injective on the resulting canonical attributed-graph isomorphism classes, the composition is orbit complete for the declared rescaling–permutation group.*

Proof. Canonicalization is constant on every positive-rescaling orbit by Theorem 4.1. Let $p \in P$ act by permutation matrices P_H on hidden vertices and P_E on scalar edges. Then

$$B_{pS} = P_H B_S P_E^\mathsf{T}, \qquad \Pi_{pS} = P_E \Pi_S P_E^\mathsf{T}.$$

Thus $\mathrm{Can}_+(p \cdot w) = p \cdot \mathrm{Can}_+(w)$, so the canonical graphs are isomorphic and $\mathcal{E}_H$ is permutation invariant.

If $\mathcal{E}_H(w) = \mathcal{E}_H(w')$ and H is injective on canonical graph isomorphism classes, some hidden permutation p satisfies $\mathrm{Can}_+(w') = p \cdot \mathrm{Can}_+(w)$. Applying p^{-1} shows that $p^{-1} \cdot w'$ and w have the same canonical representative and hence lie in the same positive-rescaling orbit. Therefore w' and w lie in the same $\mathcal{D}_+ \rtimes P$ orbit. □

Sensitivity to rescaling. For a continuously differentiable descriptor $f : \mathbb{R}^S \to \mathbb{R}^k$, invariance is equivalent to $Df(y)T_S = 0$ for every y. Indeed, rescaling moves y only along $\operatorname{im} T_S$. Integrating the derivative along such a direction u gives

$$\|f(y+u) - f(y)\|_2 \le \|u\|_2 \int_0^1 \|Df(y+tu)T_S\|_{\mathrm{op}}\, dt. \tag{20}$$

Projecting a descriptor. For a linear descriptor L, invariance is equivalent to $L = L\Pi_S$. The closest invariant linear map in Frobenius norm is $L\Pi_S$: for every $K = K\Pi_S$, orthogonality gives $\|L - K\|_F^2 = \|L\Pi_S - K\|_F^2 + \|LT_S\|_F^2$. More generally, composing an L_f-Lipschitz map f with Π_S gives an invariant descriptor satisfying $\|f(\Pi_S y) - f(\Pi_S y')\|_2 \le L_f d_Q([w], [w'])$.

B.3 The global invariant ring and support transitions

Proof of Theorem 4.3. Write $E := \bar{E}_{\mathrm{arch}}$ and let

$$f(w) = \sum_{a \in \mathbb{N}^E} c_a w^a$$

be a polynomial, with only finitely many nonzero coefficients. Under the rescaling $\alpha \in \mathbb{R}^{V_H}$,

$$(g_\alpha \cdot w)^a = \exp(\alpha^{\mathsf{T}} B a) w^a.$$

Group the monomials by the character $\lambda = Ba$. Distinct exponential characters $\alpha \mapsto e^{\alpha^{\mathsf{T}}\lambda}$ are linearly independent. Therefore $f(g_\alpha \cdot w) = f(w)$ for every α if and only if every monomial appearing in f satisfies

$$a \in \mathcal{K} := \ker B \cap \mathbb{N}^E.$$

By Lemma B.1, every $a \in \mathcal{K}$ is a finite sum of incidence vectors of boundary-to-boundary paths. Consequently,

$$w^a = \prod_i p_{\gamma_i}(w),$$

which proves (8).

We next identify the indecomposable elements of $\mathcal{K}$. A path vector χ_γ with no internal boundary vertex is indecomposable: if $\chi_\gamma = a + b$ for $a, b \in \mathcal{K}$, then the supports of a and b partition the edges of a unit-flow path. Conservation at every internal hidden vertex forces any nonempty part to continue through the entire path, so one summand must be zero. Conversely, the flow decomposition of any nonzero $a \in \mathcal{K}$ writes it as a sum of elementary path vectors. If a is indecomposable, that sum has exactly one term. Hence the elementary path vectors are precisely the atoms of $\mathcal{K}$.

It follows that the path monomials are the unique minimal monomial generators of the semigroup algebra $\mathbb{R}[\mathcal{K}]$. One can also see that no arbitrary algebra-generating family can use fewer generators. Let $\mathfrak{m}$ be the ideal of invariant polynomials with zero constant term. The quotient $\mathfrak{m}/\mathfrak{m}^2$ has as a basis the classes of the atom monomials, because decomposable monomials lie in $\mathfrak{m}^2$ and distinct monomials are linearly independent. The nonconstant parts of any algebra-generating family must span $\mathfrak{m}/\mathfrak{m}^2$. Therefore the number of elementary boundary paths is the minimum possible number of algebra generators. □

Proof of Theorem 4.4. For part (a), boundary completeness is equivalent to the existence of $c > 0$ with $B_S c = 0$. In one direction, sum the incidence vectors of boundary paths covering all support edges. In the other, decompose such a positive flow using Lemma B.1; every edge must occur in a boundary path.

Assume such a positive flow exists, and let $w_n = g_n \cdot w$ be a convergent sequence in the orbit, with limit z. In log magnitudes,

$$y_n = y + B_S^{\mathsf{T}} \alpha_n.$$

Because w_n converges to a finite vector, each coordinate of y_n is bounded above. Furthermore,

$$c^{\mathsf{T}} y_n = c^{\mathsf{T}} y$$

for every n. If some supported coordinate of z were zero, the corresponding coordinate of y_n would tend to $-\infty$. Since $c > 0$ and all other coordinates are bounded above, this would force $c^{\mathsf{T}} y_n \to -\infty$, a contradiction. Hence z has support S and the same signs as w. Since $\operatorname{im} B_S^{\mathsf{T}}$ is closed and $y_n - y \to \log|z_S| - y$, the limit difference belongs to $\operatorname{im} B_S^{\mathsf{T}}$. Therefore z lies in the orbit, proving that the orbit is closed.

Suppose no such positive flow exists. Lemma B.2, applied to B_S, gives α such that

$$d := B_S^{\mathsf{T}} \alpha \le 0, \qquad d \neq 0,$$

where we have changed the sign of α if necessary. The orbit curve

$$w_e(t) := e^{t d_e} w_e, \qquad t \ge 0,$$

converges to a finite vector with strictly smaller support, which is not in the orbit. Hence the orbit is not closed. This proves the converse and completes part (a).

We next prove part (b). Let

$$F := V_B \cup \{v \in V_H : v \text{ is incident to an edge of } S_{\mathrm{bb}}\}.$$

Consider the directed graph with edge set $S \setminus S_{\mathrm{bb}}$ and contract all vertices in F to a single vertex $*$. The contracted graph is acyclic. Indeed, a directed cycle not using $*$ would already be a cycle in the original DAG, while a cycle through $*$ would lift to a directed path, using at least one noncore edge, from one vertex of F to another. Prefixing and suffixing this path by boundary paths inside S_{bb} would place every one of its edges on a boundary-to-boundary path in S, a contradiction.

Choose a topological numbering τ of the contracted graph, strictly increasing along every directed edge. Set $\alpha_v = 0$ for hidden $v \in F$ and

$$\alpha_v := \tau(*) - \tau([v])$$

for hidden $v \notin F$, where $[v]$ denotes its contracted vertex. Both endpoints of a core edge belong to F, so its character is zero. For every noncore edge $e : u \to v$,

$$(B_S^{\mathsf{T}} \alpha)_e = \alpha_v - \alpha_u = -\big(\tau([v]) - \tau([u])\big) < 0,$$

with the same formula when one endpoint is a boundary vertex because boundary potentials are fixed at zero. Thus the rescaling character is zero on every core edge and strictly negative on every noncore edge. Coordinatewise, the associated one-parameter rescaling converges to w_{bb}. Part (a) shows that the orbit of w_{bb} is closed.

To prove uniqueness, let $z \in \overline{\mathcal{D}_+ \cdot w}$ and suppose that $\mathcal{D}_+ \cdot z$ is closed. Coordinates that vanish at w remain zero along the orbit and in its closure, so $\operatorname{supp}(z) \subseteq S$. If $e \in S_{\mathrm{bb}}$, choose a boundary path $\gamma \subseteq S$ containing e. The polynomial p_γ is continuous and invariant, and $p_\gamma(w) \neq 0$; hence $p_\gamma(z) = p_\gamma(w) \neq 0$, so every edge of γ, including e, belongs to $\operatorname{supp}(z)$. Thus $S_{\mathrm{bb}} \subseteq \operatorname{supp}(z)$. Closedness and part (a) imply that $\operatorname{supp}(z)$ is boundary complete. Every one of its edges therefore lies on a boundary path contained in S, so $\operatorname{supp}(z) \subseteq S_{\mathrm{bb}}$. Hence $\operatorname{supp}(z) = S_{\mathrm{bb}}$.

Nonzero coordinate signs are preserved under positive rescaling and under nonzero limits, and all boundary-path products of z equal those of w_{bb}. Corollary 4.2, applied on the boundary-complete support S_{bb}, now shows that z and w_{bb} lie in the same orbit. Thus this is the unique closed orbit in the closure.

For part (c), if the declared parameter space contains an edge with a nontrivial rescaling character, choose a parameter vector supported only on that edge. Scaling in a direction that decreases its character sends the orbit to the zero vector, which is not in the orbit. Thus the quotient contains a nonclosed singleton and is not T_1, hence not Hausdorff.

Finally, suppose a continuous invariant map F_0 into a Hausdorff space separated all exact orbits. Hausdorff spaces are T_1, so $\{F_0(w)\}$ is closed. Orbit separation gives

$$F_0^{-1}(\{F_0(w)\}) = \mathcal{D}_+ \cdot w,$$

which would make every orbit closed, contradicting the preceding construction. Therefore no such global continuous complete descriptor exists. □

B.4 Functional completeness

Orbit completeness is not, by itself, completeness for the represented function. Let $G = \mathcal{D}_+ \rtimes P$ and suppose a G-invariant set U satisfies the identifiability condition

$$F_{\mathcal{A},\theta} = F_{\mathcal{A},\theta'} \quad \Longleftrightarrow \quad \theta' \in G \cdot \theta, \qquad \theta, \theta' \in U. \tag{21}$$

On U, equality of functions, equality of symmetry orbits, and equality of any orbit-complete invariant are equivalent. In particular, the canonical graph isomorphism class is functionally complete there by Corollary B.3. The identifiability assumption is essential; the quotient construction does not establish it.

C Proofs and supplementary results for Section 5

C.1 Permutation and factorization gauges

Let $P = \prod_{v \in V_{\mathrm{hid}}} \mathfrak{S}_{d_v}$ act by

$$W_e \mapsto P_{t(e)} W_e P_{s(e)}^{\mathsf{T}}, \qquad b_v \mapsto P_v b_v. \tag{22}$$

Proposition C.1 (Permutation quotient)**.** *Two parameter points θ and θ' lie in the same hidden-permutation orbit if and only if*

$$\Gamma_{\mathrm{wt}}(\theta) \cong \Gamma_{\mathrm{wt}}(\theta')$$

by a type-preserving attributed-graph isomorphism that fixes the boundary coordinates.

Proof of Proposition C.1. A tuple of hidden permutation matrices (P_v) acts on coordinate vertices by $(v, i) \mapsto (v, \pi_v(i))$. Equation (22) shows that this vertex map preserves every typed scalar edge and its transformed attribute, including the constant-source edges representing biases. Hence points in the same P-orbit give isomorphic typed attributed graphs.

Conversely, an admissible type-preserving isomorphism fixes every boundary coordinate and restricts on each hidden block v to a permutation π_v. Because the parent-block and parent-arrow labels are retained, preservation of all scalar-edge attributes is exactly

$$W'_e = P_{t(e)} W_e P_{s(e)}^{\mathsf{T}}, \qquad b'_v = P_v b_v.$$

Thus θ' belongs to the hidden-permutation orbit of θ. □

We next prove the factorization theorem. The following decomposition makes both the orbit classification and the closure statement transparent.

Lemma C.2 (Four-part decomposition of a factorization)**.** *Let $B : X \to H$ and $A : H \to Y$, with $\dim H = r$, and set $a := \operatorname{rank} A$, $b := \operatorname{rank} B$, and $k := \operatorname{rank}(AB)$. Then H admits a direct-sum decomposition*

$$H = H_{11} \oplus H_{10} \oplus H_{01} \oplus H_{00}$$

with dimensions

$$\dim H_{11} = k, \quad \dim H_{10} = b - k, \quad \dim H_{01} = a - k, \quad \dim H_{00} = r - a - b + k,$$

such that

$$\operatorname{im} B = H_{11} \oplus H_{10},$$
$$\ker A = H_{10} \oplus H_{00},$$

$A|_{H_{11}}$ is an isomorphism onto $\operatorname{im}(AB)$, and $A|_{H_{01}}$ is an isomorphism onto a complement of $\operatorname{im}(AB)$ in $\operatorname{im} A$.

Proof. Set $C := \operatorname{im} B$ and $N := \ker A$. Since $A(C) = \operatorname{im}(AB)$,

$$\dim(C \cap N) = \dim C - \operatorname{rank}(A|_C) = b - k.$$

Choose $H_{10} := C \cap N$, a complement H_{11} of H_{10} in C, and a complement H_{00} of H_{10} in N. Then

$$\dim H_{11} = k, \qquad \dim H_{00} = (r - a) - (b - k) = r - a - b + k.$$

Choose H_{01} as a complement of $C + N$ in H. Since

$$\dim(C + N) = b + (r - a) - (b - k) = r - a + k,$$

we have $\dim H_{01} = a - k$. The restriction $A|_{H_{11}}$ is injective and has image $A(C) = \mathrm{im}(AB)$. The restriction $A|_{H_{01}}$ is injective because $H_{01} \cap N = \{0\}$. If $Ah \in A(C)$ for $h \in H_{01}$, then $A(h-c) = 0$ for some $c \in C$, whence $h \in C + N$ and therefore $h = 0$. Thus $A(H_{01})$ is complementary to $A(C)$ in $\mathrm{im}\, A$. □

Proof of Theorem 5.1. The product, output image, and input kernel in (9) are invariant under $(A, B) \mapsto (Ag^{-1}, gB)$, so necessity is immediate.

For sufficiency, suppose these three quantities agree. Write

$$C := \mathrm{im}\, B, \quad C' := \mathrm{im}\, B', \quad N := \ker A, \quad N' := \ker A'.$$

Equality $\ker B = \ker B'$ makes

$$\phi : C \to C', \qquad \phi(Bx) := B'x,$$

well defined and bijective. Equality of the products gives

$$A'\phi(Bx) = A'B'x = ABx = A(Bx),$$

so $A'\phi = A$ on C, and $\phi(C \cap N) = C' \cap N'$.

Choose the decompositions of Lemma C.2 for both pairs, with $H'_{10} = \phi(H_{10})$ and $H'_{11} = \phi(H_{11})$. Since $\mathrm{im}\, A = \mathrm{im}\, A'$, choose H_{01} and H'_{01} so that $A(H_{01}) = A'(H'_{01})$, a common complement of $\mathrm{im}(AB)$ in the shared output image. Define g to equal ϕ on $H_{11} \oplus H_{10}$, choose any isomorphism $H_{00} \to H'_{00}$, and set

$$g|_{H_{01}} := (A'|_{H'_{01}})^{-1} \circ A|_{H_{01}}.$$

Then g is invertible, $gB = B'$, and $A'g = A$ on all four summands. Hence $(A', B') = (Ag^{-1}, gB)$.

□

Proof of Corollary 5.2. Use Lemma C.2 and, for $t \geq 0$, let g_t act as the identity on $H_{11} \oplus H_{00}$, as $e^{-t}I$ on H_{10}, and as $e^{t}I$ on H_{01}. Then

$$g_t \cdot (A, B) \longrightarrow (A_{\min}, B_{\min}), \tag{23}$$

where

$$A_{\min}B_{\min} = AB, \qquad \mathrm{rank}\, A_{\min} = \mathrm{rank}\, B_{\min} = \mathrm{rank}(AB).$$

If either factor rank exceeds $\mathrm{rank}(AB)$, the limit is outside the orbit, so the orbit is not closed.

Conversely, assume $\mathrm{rank}\, A = \mathrm{rank}\, B = \mathrm{rank}(AB) = k$, and let a sequence in the orbit converge to $(\bar{A}, \bar{B})$. Its product remains AB, while rank cannot increase under limits. Thus

$$k = \mathrm{rank}(\bar{A}\bar{B}) \leq \mathrm{rank}\, \bar{A} \leq k, \qquad k \leq \mathrm{rank}\, \bar{B} \leq k.$$

Therefore $\mathrm{im}\, \bar{A} = \mathrm{im}(AB) = \mathrm{im}\, A$ and $\ker \bar{B} = \ker(AB) = \ker B$. The orbit classification already proved places $(\bar{A}, \bar{B})$ in the original orbit. Hence the orbit is closed. The limit in (23) is therefore a closed orbit; any other closed orbit in the same closure has the same product and minimal factor ranks, and is the same orbit by (9). This proves uniqueness.

Finally, let f be a continuous invariant on these factor pairs, with values in a Hausdorff space Z. Along (23), invariance makes $f(g_t \cdot (A, B))$ constant and continuity makes it converge to $f(A_{\min}, B_{\min})$. Uniqueness of limits in Z gives

$$f(A, B) = f(A_{\min}, B_{\min}).$$

Minimal factorizations of the same product lie in the same orbit, so $f(A, B)$ depends only on AB. □

Proof of Corollary 5.3. Apply Theorem 5.1 to each query–key pair with

$$A = W_h^Q, \qquad B = (W_h^K)^\mathsf{T},$$

and to each value–output pair with

$$A = W_h^V, \qquad B = W_h^O.$$

Replacing the group variable by its inverse matches the convention in (10). The two rank-aware triples are complete for the two independent general linear factors of head h. Let $N_H := \prod_{h=1}^{H}(\mathrm{GL}_{r_q} \times \mathrm{GL}_{r_v})$. Taking all head attributes is complete for N_H. If two unordered head graphs agree, first match their heads by a permutation. Equality of each pair of triples then supplies the two within-head gauges, proving the first assertion.

Under (11),

$$\operatorname{im} W_h^Q = \operatorname{im} M_h^{QK}, \qquad \ker(W_h^K)^\mathsf{T} = \ker M_h^{QK},$$
$$\operatorname{im} W_h^V = \operatorname{im} M_h^{VO}, \qquad \ker W_h^O = \ker M_h^{VO}.$$

Thus the products retain the complete triples. The query–key quotient is the rank-r_q manifold in $\mathbb{R}^{d\times d}$, of dimension $r_q(2d - r_q)$, and the value–output quotient is the rank-r_v manifold in $\mathbb{R}^{d\times d_o}$, of dimension $r_v(d + d_o - r_v)$. Summing over heads gives dimension $H[r_q(2d - r_q) + r_v(d + d_o - r_v)]$; quotienting by the finite group $\mathfrak{S}_H$ does not change local dimension.

The final functional statement combines this orbit completeness with the identifiability theorem for vanilla multi-head attention under full rank, a common head dimension, and pairwise-distinct query–key products in Tran et al. (2026). □

The product invariant for vanilla attention extends to any gauge that must commute with fixed latent operators. A basis of the generated operator algebra gives a finite complete family.

C.2 Gauges restricted by fixed latent operators

Let $\mathcal{A} \subseteq \mathbb{R}^{r\times r}$ be the unital algebra generated by the fixed latent operators. Its centralizer is the group $Z_{\mathrm{GL}_r}(\mathcal{A}) = \{G \in \mathrm{GL}_r : GT = TG \text{ for every } T \in \mathcal{A}\}$. Such a G preserves the inserted operators as well as the factor product.

Theorem C.3 (Complete composites for a centralizer gauge)**.** *Let $T_0 = I, T_1, \ldots, T_{s-1}$ be a basis of $\mathcal{A}$, and let $Q, K \in \mathbb{R}^{d\times r}$ have full column rank. Under*

$$(Q, K) \longmapsto (QG, KG^{-\mathsf{T}}), \qquad G \in Z_{\mathrm{GL}_r}(\mathcal{A}),$$

the tuple

$$\Phi_{\mathcal{A}}(Q, K) := \left(QT_jK^\mathsf{T}\right)_{j=0}^{s-1} \tag{24}$$

is orbit complete.

Proof of Theorem C.3. Every component of (24) is invariant because, for $G \in Z_{\mathrm{GL}_r}(\mathcal{A})$,

$$(QG)T_j(KG^{-\mathsf{T}})^\mathsf{T} = QGT_jG^{-1}K^\mathsf{T} = QT_jK^\mathsf{T}.$$

Suppose two full-column-rank pairs (Q, K) and (Q', K') have the same tuple. Equality of the component $T_0 = I$ gives

$$QK^\mathsf{T} = Q'(K')^\mathsf{T}.$$

The full-rank case of Theorem 5.1 yields a unique $G \in \mathrm{GL}_r$ such that

$$Q' = QG, \qquad K' = KG^{-\mathsf{T}}.$$

For every j, equality of the corresponding composite gives

$$Q(GT_jG^{-1} - T_j)K^\mathsf{T} = 0.$$

Since Q has a left inverse and K^T has a right inverse,

$$GT_jG^{-1} = T_j.$$

The T_j span $\mathcal{A}$, so $G \in Z_{\mathrm{GL}_r}(\mathcal{A})$. This proves orbit completeness. □

Rotary operators. For $r = 2s$, consider $R = \mathrm{diag}(R(\omega_1), \ldots, R(\omega_s))$, where $R(\omega)$ is the two-dimensional rotation through angle ω and the $\omega_i \in (0, \pi)$ are distinct. Let P_i project onto plane i, and let J_i be rotation through $\pi/2$ on that plane and zero elsewhere. The algebra generated by R is $\mathrm{span}_{\mathbb{R}}\{P_i, J_i\}_{i=1}^{s}$, and its invertible commuting maps are

$$G = \sum_{i=1}^{s} (a_i P_i + b_i J_i), \qquad (a_i, b_i) \neq (0, 0) \text{ for each } i.$$

Indeed, the distinct complex eigenvalues $e^{\pm \mathrm{i}\omega_i}$ force a commuting map to preserve each rotary plane; a real commuting block has the form $a_i I + b_i J$. The same eigenvalues show that the generated algebra has dimension $2s$. Thus Theorem C.3 gives the complete full-rank attributes $(QP_iK^{\mathsf{T}}, QJ_iK^{\mathsf{T}})_i$. Repeated frequencies allow mixing between equal-frequency planes and require the larger commuting algebra. This example uses the rotary symmetry structure studied by Tran et al. (2026).

C.3 Normalization layers

Normalization restricts which coordinate changes preserve a block. Its trainable affine parameters and, for BatchNorm, running statistics must also be included (Ba et al., 2016; Zhang & Sennrich, 2019; Ioffe & Szegedy, 2015).

A module ρ_η with parameters or state η is compatible with a coordinate change M when

$$\rho_{M\cdot\eta}(Mz) = M\rho_\eta(z). \tag{25}$$

Together with the local-frame rule in Section 3.2, this preserves the network function by the induction in Appendix E. All branches entering an addition must use the same coordinate change.

Exact frames for LayerNorm and RMSNorm. Let $\mathbf{1} \in \mathbb{R}^d$ be the all-ones vector, $P_1 := d^{-1}\mathbf{1}\mathbf{1}^{\mathsf{T}}$, and $C_1 := I - P_1$. For the core normalizers with the practical assumption $\varepsilon > 0$, write

$$\mathrm{LN}^0_\varepsilon(x) := \frac{C_1 x}{\sqrt{d^{-1}\|C_1 x\|_2^2 + \varepsilon}}, \qquad \mathrm{RMS}^0_\varepsilon(x) := \frac{x}{\sqrt{d^{-1}\|x\|_2^2 + \varepsilon}}. \tag{26}$$

Theorem C.4 (Maximal linear frames of the core normalizers)**.** *The maximal groups of invertible linear maps satisfying $N(Mx) = MN(x)$ for every x are*

$$G_{\mathrm{RMS}} = O(d), \tag{27}$$

$$G_{\mathrm{LN}} = \{aP_1 + QC_1 : a \in \mathbb{R} \setminus \{0\},\ Q \in O(d),\ Q\mathbf{1} = \mathbf{1}\}. \tag{28}$$

Thus RMSNorm permits any orthogonal hidden frame. LayerNorm permits an orthogonal frame on the centered subspace $\mathbf{1}^\perp$ and an arbitrary nonzero rescaling of the mean direction, which the module discards.

Table 1: Coordinate changes compatible with normalization. The last column applies uniformly to arbitrary diagonal gains and biases.

Module	Core map	With diagonal gain and bias
RMSNorm	$O(d)$	signed permutations
LayerNorm	$\mathbb{R}^\times \times O(d-1)$	permutations composed with one global sign
BatchNorm	signed channel permutations	same, with statistics transformed

Proof. For RMSNorm, equivariance forces $\|Mx\|_2 = \|x\|_2$ for every x, hence $M \in O(d)$; the converse is immediate. For LayerNorm, applying equivariance to $\mathbf{1}$ gives $M\mathbf{1} \in \mathrm{span}\{\mathbf{1}\}$. Applying it to nonzero $x \in \mathbf{1}^\perp$ gives $Mx \in \mathbf{1}^\perp$ and $\|Mx\|_2 = \|x\|_2$. Thus M is arbitrary and invertible on the mean line and orthogonal on $\mathbf{1}^\perp$, which is exactly (28); every such map is sufficient. □

Standard affine normalization has $N_{\gamma,\beta}(x) = \mathrm{diag}(\gamma)N^0(x) + \beta$. The monomial elements of the core groups give simple exact frames valid for arbitrary gains: $M = DP$ for RMSNorm, with $D_{ii} \in \{\pm 1\}$, and $M = sP$ for LayerNorm, with $s \in \{\pm 1\}$. In both cases, setting $\gamma' = P\gamma$ and

$\beta' = M\beta$ gives $N_{\gamma',\beta'}(Mx) = MN_{\gamma,\beta}(x)$. These are signed permutations for RMSNorm and permutations composed with one global sign for LayerNorm; parameter-specific repeated gains admit larger stabilizers.

For function-level analysis, affine normalization parameters can often be removed from the symmetry question altogether. If every consumer has an affine bias edge (or $\beta = 0$), fold them exactly into each consumer:

$$W\big(\mathrm{diag}(\gamma)N^0(x) + \beta\big) + b = (W\,\mathrm{diag}(\gamma))N^0(x) + (b + W\beta). \tag{29}$$

The full core groups then act on the folded effective maps. This is an exact representation of the same block, not a claim that an arbitrary orthogonal frame preserves the raw diagonal-affine coordinates. Multiple affine consumers are handled by folding into each one.

Complete operator graphs for normalized blocks. The orthogonal core groups yield a simple all-rank representation. Let $\bar{B} : X \oplus \mathbb{R} \to \mathbb{R}^d$ include the incoming bias, and let $A : \mathbb{R}^d \to Y$ stack all outgoing affine consumers after the folding in (29).

Theorem C.5 (Complete normalized-block operators)**.** *For an RMSNorm block, the tuple*

$$\mathcal{I}_{\mathrm{RMS}}(A, \bar{B}) := \big(\bar{B}^\mathsf{T}\bar{B}, A\bar{B}, AA^\mathsf{T}\big) \tag{30}$$

is orbit complete for $(A, \bar{B}) \mapsto (AQ^\mathsf{T}, Q\bar{B})$, $Q \in O(d)$, at every rank.

For a LayerNorm block, the module discards the incoming mean and its output lies in $\mathbf{1}^\perp$. After identifying

$$\bar{B} \sim \bar{B} + \mathbf{1}a^\mathsf{T}, \qquad A \sim A + u\mathbf{1}^\mathsf{T},$$

the tuple

$$\mathcal{I}_{\mathrm{LN}}(A, \bar{B}) := \big(\bar{B}^\mathsf{T}C_1\bar{B}, AC_1\bar{B}, AC_1A^\mathsf{T}\big) \tag{31}$$

is orbit complete for the orthogonal frame on $\mathbf{1}^\perp$, again at every rank.

Proof. For RMSNorm, stack the maps as $Z = [\bar{B}\ A^\mathsf{T}]$. The three entries in (30) are the blocks of $Z^\mathsf{T}Z$. Two real matrices have the same Gram matrix if and only if they differ by a left orthogonal map: the induced isometry between their column spans extends to $Q \in O(d)$. This proves necessity and sufficiency without a rank assumption. For LayerNorm, use $Z = [C_1\bar{B}\ C_1A^\mathsf{T}]$, whose columns lie in $\mathbf{1}^\perp$, and apply the same argument in that $(d-1)$-dimensional space. □

A pre-LayerNorm residual block. Consider

$$T(x) = x + W_2\phi\big(W_1\,\mathrm{LN}^0_\varepsilon(x) + b_1\big) + b_2.$$

For any $M \in G_{\mathrm{LN}}$, set

$$W_1' = W_1M^{-1}, \qquad W_2' = MW_2, \qquad b_1' = b_1, \qquad b_2' = Mb_2.$$

Then $T'(Mx) = MT(x)$. Applying one frame throughout a residual stream and transforming its embedding and readout at the boundaries therefore preserves the complete model function. The within-head query–key and value–output gauges of Section 5.2 are separate: they cancel inside each attention head and do not cross LayerNorm.

BatchNorm. Training-mode BatchNorm is a map on a batch. Let $X \in \mathbb{R}^{d\times m}$ collect examples by columns and normalize each channel using its batch mean and variance. For every signed channel permutation $M = DP$, where P is a permutation and $D_{ii} \in \{\pm 1\}$,

$$\mathrm{BN}^0_\varepsilon(MX) = M\,\mathrm{BN}^0_\varepsilon(X). \tag{32}$$

With affine parameters and inference statistics, exact equivariance is obtained by

$$\mu' = M\mu, \qquad v' = Pv, \qquad \gamma' = P\gamma, \qquad \beta' = M\beta. \tag{33}$$

Equation (32) follows because channel means transform by M and channel variances by P; substituting the transformed affine parameters and running statistics gives (33). Training-mode BatchNorm removes every pre-normalization bias exactly. At inference, it depends on the pre-normalization bias and running mean only through their difference, so $b \mapsto b + c$, $\mu \mapsto \mu + c$ is an exact state symmetry.

Arbitrary channel scaling is exact only in the idealized $\varepsilon = 0$ limit. For a centered channel u with variance v and $c \neq 0$,

$$\mathrm{BN}^0_\varepsilon(cu) = \mathrm{sgn}(c)\sqrt{\frac{v+\varepsilon}{v+\varepsilon/c^2}}\,\mathrm{BN}^0_\varepsilon(u), \tag{34}$$

which gives the exact finite-ε deviation. If BatchNorm is followed by an elementwise ReLU, intersecting (32) with ReLU equivariance leaves channel permutations but removes sign flips.

D Reliability proofs and summary of graph representations

D.1 Prediction and intervention guarantees

Proof of Proposition 6.1. Fix θ and let $e = \mathrm{err}_G(p, \tau; \theta)$. For any $g_1, g_2 \in G$, the triangle inequality gives

$$d(p(g_1\theta), p(g_2\theta)) \leq 2e + d(\tau(g_1\theta), \tau(g_2\theta)).$$

Taking the supremum gives $\mathrm{diam}_G\, p(\theta) \leq 2e + \mathrm{diam}_G\, \tau(\theta)$. Interchanging p and τ gives the reverse inequality. Together they prove (12) when both diameters are finite. For an invariant target, $\mathrm{diam}_G\, \tau(\theta) = 0$, giving (13). □

Random rescalings. The usual bias–variance identity also separates prediction error from variation between equivalent parameters. Let $\mathcal{Y}$ be a real Hilbert space, let τ be invariant, and let ν be a probability distribution on G such that $p(g\theta)$ has finite second moment. With $\bar{p}_\nu(\theta) = \mathbb{E}_{g\sim\nu} p(g\theta)$,

$$\begin{aligned}\mathbb{E}_{g\sim\nu}\|p(g\theta) - \tau(\theta)\|^2 &= \|\bar{p}_\nu(\theta) - \tau(\theta)\|^2 \\ &\quad + \mathbb{E}_{g\sim\nu}\|p(g\theta) - \bar{p}_\nu(\theta)\|^2.\end{aligned} \tag{35}$$

To verify this, expand the square around $\bar{p}_\nu$; the cross term has zero expectation. Averaging over an arbitrary distribution of rescalings need not produce an invariant predictor. For a finite or compact group, averaging with its normalized Haar measure does, by the change of variable $g \mapsto gh$.

Proof of Proposition 6.2. The two equivariance assumptions give

$$\mathcal{I}_{S(g\theta)}(g\theta) = \mathcal{I}_{gS(\theta)}(g\theta) = g\,\mathcal{I}_{S(\theta)}(\theta).$$

If g preserves the function of the intervened model, both interventions compute the same function. □

D.2 Summary of graph representations

Table 2 groups the constructions most relevant to this paper. The method descriptions follow the cited sources; the symmetry statements follow from the transformation rules in Appendix E.2 and the examples in Appendix F. They concern the stated graph construction, not the usefulness of the original method on its reported task. A method can work well on trained checkpoints without being invariant to a change of parameterization.

Positive rescaling applies to compatible homogeneous units; dense basis changes apply to linear latent spaces. These are different architectural settings, not interchangeable transformations. Permutations only relabel the graphs when vertex types, boundary identities, and any additional data are carried along. Invariant graph-level outputs also require an isomorphism-invariant readout; selected nodes or masks require equivariant selection and tie handling.

Table 2: Graph representations and their behavior under parameter symmetries. Rescaling means independent positive hidden-unit rescaling, not a single global multiplier. Statements about dense basis changes apply only where that symmetry is admitted.

Construction and examples	Behavior under parameter symmetries
Architecture graphs You et al. (2020); Kadlecová et al. (2024)	Use wiring and operation types rather than learned weights. Fixed-architecture parameter symmetries leave these attributes unchanged. This does not address function-preserving rewrites that change the architecture.
Exact support graphs Motif counts in sparse networks (Zahn et al., 2024)	Positive rescaling preserves exact zeros, so support statistics are unchanged up to relabeling. Dense basis changes can alter scalar support. A support chosen by magnitude thresholding is not generally rescaling invariant.
Magnitude-weighted graphs Training summaries and spectral regularization (Vahedian et al., 2022; Tam & Dunson, 2020)	Strength, weighted centrality, and symmetrized graph-Laplacian spectra can change under rescaling. Normalizing by one common scale does not remove independent unit scales. Dense basis changes also alter these scalar weights.
Weight-based persistence Neural Persistence and Deep Graph Persistence (Rieck et al., 2019; Girrbach et al., 2023)	Global normalization and weight standardization do not enforce hidden-unit rescaling invariance. Filtration values and their order can change. Invariant edge scores are sufficient to make the resulting persistence summary invariant.
Weighted curvature Robustness and pruning (Tan et al., 2024; 2026)	Curvature depends on edge lengths and neighborhood measures. When these use raw magnitudes, rescaling can change curvature and pruning ranks. Curvature on an unweighted architecture graph has no such weight dependence.
Activation-weight graphs Topological Uncertainty and MAGDiff (Lacombe et al., 2021; Arnal et al., 2023)	An edge contribution $w_{uv}h_u(x)$ acquires the target unit's scale. Scores and distances built from these contributions therefore need not be invariant. Dense basis changes mix the coordinate contributions.
Activation-correlation graphs Error prediction and functional connectomes (Corneanu et al., 2020; Songdechakraiwut & Wu, 2025)	Pearson correlations are unchanged by positive unit scales when variances are nonzero; absolute correlations also remove compatible sign changes. Dense basis changes generally change coordinate correlations.
Filter-similarity graphs Cosine-based pruning (King et al., 2023)	Cosine similarity removes each filter's overall scale, but not nonuniform rescaling of its input channels. It therefore does not cover the full multilayer rescaling group.
Gradient-based scores Connection sensitivity and SynFlow (Lee et al., 2019; Tanaka et al., 2020)	For an invariant differentiable objective, $w_e \partial J / \partial w_e$ is unchanged by diagonal rescaling and relabeled by permutations. Raw gradients do not have this property. A coordinatewise score does not by itself handle dense basis changes.
Permutation-aware encoders Graph Metanetworks and neural-graph GNNs (Lim et al., 2024; Kofinas et al., 2024)	Respect the encoded graph relabelings. This alone does not imply invariance to weight-changing rescalings or dense basis changes.
Scaling-aware encoders ScaleGMN and Monomial-NFN (Kalogeropoulos et al., 2024; Tran et al., 2024)	Explicitly encode their declared scaling and permutation actions, with invariant or equivariant outputs as required. Dense factor-basis changes are outside these scalar symmetry guarantees.
Low-rank factor representations Learning on LoRAs (Putterman et al., 2024)	Uses canonicalization, invariant features, or equivariant layers for $(U, V) \mapsto (UG, G^{-1}V)$. This directly addresses dense factor symmetries; it is distinct from processing raw factor entries as invariant attributes.

E TRANSFORMATION RULES

E.1 SEMANTICS PRESERVATION UNDER LOCAL FRAME CHANGES

Let $H := \bigoplus_{v \in V} H_v$. The *block adjacency operator* $\mathbf{A}_\theta : H \to H$ applies all incoming maps at once,

$$(\mathbf{A}_\theta h)_v := \sum_{e:t(e)=v} W_e h_{s(e)}.$$

Its (v, u) block is the sum of the maps from H_u to H_v.

For $g = (M_v)$ acting by the local-frame rule in Section 3.2, assume inductively that $h_u(x; g \cdot \theta) = M_u h_u(x; \theta)$ for every predecessor u of v. Then

$$\begin{aligned} z_v(x; g \cdot \theta) &= M_v b_v + \sum_{e:t(e)=v} M_v W_e M_{s(e)}^{-1} M_{s(e)} h_{s(e)}(x; \theta) \\ &= M_v z_v(x; \theta), \end{aligned}$$

and $h_v(x; g \cdot \theta) = \rho_v(M_v z_v) = M_v \rho_v(z_v)$. Input and output frames are fixed, so the realized function is unchanged. The block similarity law $\mathbf{A}_{g\theta} = M \mathbf{A}_\theta M^{-1}$ follows by collecting the transformed edge blocks.

E.2 INCIDENCE AND MONOMIAL LAWS

For a scalar rescaling and hidden relabeling, the exact edge rule is

$$w'_{\pi(e)} = \frac{r_{t(e)}}{r_{s(e)}} w_e. \tag{36}$$

For $r_v > 0$, let $\alpha_v = \log r_v$ and $y_e = \log |w_e|$, with boundary log scales fixed to zero. Then

$$\log |w'_e| - \log |w_e| = \alpha_{t(e)} - \alpha_{s(e)},$$

and stacking the edge equations gives (1). For $c \in \mathbb{R}^S$,

$$\prod_{e \in S} |w'_e|^{c_e} = e^{\alpha^\mathsf{T} B_S c} \prod_{e \in S} |w_e|^{c_e}, \qquad B_S c = 0 \iff \text{invariance}. \tag{37}$$

Along a path $\gamma : u \rightsquigarrow v$, the node scales telescope:

$$p_\gamma(w') = \frac{r_v}{r_u} p_\gamma(w), \qquad p_\gamma(w) := \prod_{e \in \gamma} w_e. \tag{38}$$

Since $(\ker B_S)^\perp = \operatorname{im} B_S^\mathsf{T}$, equality of all log-linear invariants indexed by $c \in \ker B_S$ is equivalent to belonging to the same positive-rescaling orbit on a fixed nonzero support, after the edge sign pattern is fixed.

The corresponding activation and gradient laws are

$$\operatorname{Cov}'(u, v) = r_u r_v \operatorname{Cov}(u, v), \tag{39}$$

$$\rho'_{uv} = \operatorname{sgn}(r_u r_v) \rho_{uv}, \tag{40}$$

$$\left(\frac{\partial J}{\partial w_e}\right)' = \frac{r_{s(e)}}{r_{t(e)}} \frac{\partial J}{\partial w_e}, \qquad \left(w_e \frac{\partial J}{\partial w_e}\right)' = w_e \frac{\partial J}{\partial w_e}. \tag{41}$$

Thus absolute correlations are invariant when both activation variances are nonzero, and multiplicative-gate derivatives are invariant when J depends only on the represented function and fixed data.

F EXAMPLES OF GRAPH DESCRIPTORS

The same graph statistic can be invariant for one choice of edge attributes and sensitive to parameterization for another. The examples below explain the distinctions in Table 2. Throughout, boundary scales are one, and positive rescaling is used only where it preserves the network function.

F.1 Architecture, support, and thresholded graphs

An architecture graph contains no learned weights, so its degree counts, path lengths, clustering, and other structural statistics are unchanged by fixed-architecture parameter symmetries (You et al., 2020; Kadlecová et al., 2024). Exact support graphs also survive positive rescaling, since multiplying an entry by a nonzero number preserves whether it is zero. Dense basis changes need not preserve scalar support.

For a positive threshold τ, the graph with edges $\{e : |w_e| > \tau\}$ is different: rescaling can move an edge across the threshold. Statistics of that graph inherit this dependence. Canonicalizing before thresholding removes the rescaling ambiguity, as tested in Appendix G.4.

F.2 Weight magnitudes and path lengths

Raw incident strength sums the magnitudes of a unit's incoming and outgoing weights. The two-edge ReLU network gives a simple counterexample to its invariance:

$$f_{a,b}(x) = a\ \mathrm{ReLU}(bx), \qquad (a, b) \longmapsto (a/r, rb), \quad r > 0. \tag{42}$$

For $a, b \neq 0$, the function stays fixed while the incident strength $|a|/r + r|b|$ becomes arbitrarily large. This is the score used in the MLP intervention experiment.

Reciprocal-weight edge lengths also change:

$$\ell_e = |w_e|^{-1}, \qquad \ell'_e = \frac{r_{s(e)}}{r_{t(e)}}\ell_e. \tag{43}$$

The input–output length in (42) is $r/|a| + 1/(r|b|)$. In contrast, the boundary-path product ab and its log magnitude are invariant. This distinction separates an additive distance built from raw weights from a path product whose hidden scales cancel.

F.3 Spectra and centrality

The directed adjacency transforms by similarity, $A' = MAM^{-1}$, so its eigenvalues are invariant. For a feed-forward graph, however, this adjacency is nilpotent and all eigenvalues are zero. Thus an invariant statistic need not be informative. Symmetrizing absolute weights before forming a graph Laplacian removes this similarity relation; the resulting spectrum can change under rescaling. Such a weighted Laplacian can still serve as an implementation-dependent regularizer (Tam & Dunson, 2020).

Normalizing outgoing weights does not generally fix the problem. For a non-dangling vertex u, the PageRank transition probability transforms as

$$P_{vu} = \frac{|w_{u\to v}|}{\sum_z |w_{u\to z}|}, \qquad P'_{vu} = \frac{r_v|w_{u\to v}|}{\sum_z r_z|w_{u\to z}|}. \tag{44}$$

The source scale cancels, but the target scales do not. PageRank scores and rankings can therefore change. Likewise, shortest-path centralities depend on the chosen edge lengths. On an invariant graph, a centrality-based selector still needs a tie rule that respects relabeling to guarantee permutation-equivariant selections.

F.4 Curvature and persistence

Ollivier–Ricci curvature compares the distance between two vertices with the Wasserstein distance between their neighborhood measures. Changing raw-weight edge lengths or neighborhood weights can therefore change curvature. This applies to reciprocal-weight neural graphs used for robustness analysis and pruning (Tan et al., 2024; 2026). Curvature computed on an unweighted architecture graph is instead independent of the learned weights. The distinction is the input graph, not curvature itself.

For persistence, an edge score η defines a filtration $G_t = (V, \{e : \eta_e \geq t\})$. Equality of edge scores up to relabeling is *sufficient* for invariant persistence diagrams, but not necessary: different filtrations can have the same summary. A common multiplier rescales birth and death values; independent edge

changes can also alter insertion order. Stability to small filtration perturbations is not invariance to parameter symmetries (Cohen-Steiner et al., 2007).

Neural Persistence normalizes magnitudes by a maximum weight (Rieck et al., 2019). This removes a common multiplier, not independent hidden-unit scales. Deep Graph Persistence uses standardized weights and whole-network paths to address different limitations (Girrbach et al., 2023); those steps do not enforce the rescaling cancellation used in Section 4.

F.5 Activation and similarity graphs

For fixed data, an activation-weight contribution changes as

$$w'_{uv}h'_u(x) = r_v w_{uv} h_u(x).$$

Thus the activation graphs used in Topological Uncertainty and MAGDiff need not be rescaling invariant (Lacombe et al., 2021; Arnal et al., 2023). Binary ReLU activation indicators are different: $\mathbf{1}\{z_v(x) > 0\}$ is unchanged by $z'_v = r_v z_v$ for $r_v > 0$.

Pearson correlation is unchanged by independent positive scales when both variances are nonzero. An absolute-correlation graph therefore preserves its edge attributes under positive rescaling and compatible sign changes, up to relabeling (Corneanu et al., 2020; Songdechakraiwut & Wu, 2025). Dense basis changes generally alter these coordinate correlations.

Cosine similarity between incoming filter rows removes each row's overall scale, but rescaling predecessor channels changes the rows by a common nonuniform diagonal map. Their angles can then change. Cosine-based filter graphs (King et al., 2023) therefore respect a smaller set of transformations than full multilayer positive rescaling.

F.6 Gradient scores and learned encoders

For a differentiable invariant objective J, the chain rule gives

$$\left(\frac{\partial J}{\partial w_e}\right)' = \frac{r_{s(e)}}{r_{t(e)}}\frac{\partial J}{\partial w_e}.$$

The weight multiplier cancels the gradient multiplier, so

$$q_e = w_e \frac{\partial J}{\partial w_e} = \left.\frac{\partial J(w \odot \gamma)}{\partial \gamma_e}\right|_{\gamma=\mathbf{1}} \tag{45}$$

is invariant under positive rescaling and equivariant under parameter permutations. Absolute values and symmetric sums retain these properties. This first-order connection-sensitivity score, used by the gate-saliency baseline, is related to SNIP and SynFlow (Lee et al., 2019; Tanaka et al., 2020); it is not introduced here as a new pruning rule. Coordinatewise scores alone do not remove dense basis dependence.

A graph encoder inherits invariance from invariant input attributes and an appropriate readout. Permutation-aware neural-graph encoders do not automatically handle rescaling (Lim et al., 2024; Kofinas et al., 2024). ScaleGMN and Monomial-NFN encode broader scalar symmetry groups (Kalogeropoulos et al., 2024; Tran et al., 2024); canonicalization offers a complementary preprocessing approach.

G Experimental details and additional results

G.1 Prediction experiment

Networks and data split. We train 80 fully connected ReLU networks with hidden widths $(32, 32, 32)$ on Fashion-MNIST for 50 epochs and retain epochs $1, 2, 3, 5, 8, 12, 18, 25, 35, 50$. Each run samples an AdamW learning rate log-uniformly from $[3 \times 10^{-4}, 3 \times 10^{-3}]$, weight decay from $\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$, and label smoothing from $\{0, 0.05, 0.1\}$. The batch size is 256. Runs, rather than checkpoints, are the replication unit; the split contains 48 training, 16 validation, and 16 test runs.

Table 3: MLP prediction at rescaling radius $R = 2$. IID uses untransformed checkpoints; attacked uses the bounded search. Shift penalty is attacked minus IID RMSE. Estimated diameter is the largest minus the smallest prediction found. IID, attacked, and shift-penalty entries are run-clustered point estimates; random RMSE and estimated diameter are descriptive averages over the three predictor fits.

Representation	IID RMSE	Random RMSE	Attacked RMSE	Shift penalty	Est. diameter
Raw	0.0269	0.0412	0.2156	0.1887	0.2804
Canonical	0.0260	0.0258	0.0260	≈ 0	≈ 0
Augment, $R_{\text{train}} = 0.5$	0.0256	0.0348	0.2059	0.1803	0.2650
Augment, $R_{\text{train}} = 1$	0.0235	0.0300	0.2271	0.2037	0.2928
Augment, $R_{\text{train}} = 2$	0.0249	0.0278	0.2041	0.1793	0.2698
Gauge component only	0.0341	0.0710	0.2362	0.2021	0.2851

Graph predictor. The parameter graph contains a constant bias source, fixed identities for input pixels and output classes, and exchangeable hidden vertices. Edge features are parameter sign and log-magnitude. The encoder uses three edge-conditioned message-passing rounds, hidden and edge width 48, graph width 128, dropout 0.1, and a typed invariant readout. It is trained with AdamW, learning rate 5×10^{-4}, weight decay 10^{-4}, batch size 8, and early stopping. Feature statistics are fitted separately for each representation using only untransformed training checkpoints.

The canonical representation is

$$y_{\text{can}} = \left[I - B^{\mathsf{T}}(BB^{\mathsf{T}})^{\dagger}B\right] y, \qquad y = \log|w|.$$

All parameters in the experiment are nonzero, and no clipping is applied before taking logarithms. Transformations and canonicalization use float64. The maximum canonicalization error over the tested radii is below 2×10^{-13}.

Rescaling tests and confidence intervals. For every test checkpoint and radius $R \in \{0.5, 1, 2, 4\}$, we draw 32 random transformations. The predeclared attacked values use a 60-step, four-restart projected Adam search with step size $0.08R$ and include the untransformed checkpoint among the candidates. We optimize target error and separately maximize and minimize the prediction to estimate the bounded orbit diameter. These are observed lower bounds because global optimality is not certified. Confidence intervals use 5000 paired bootstrap samples over the 16 held-out runs. We subsequently reran the complete test set with up to 240 steps and eight restarts; Figure 6 reports the effect of the search budget.

Table 3 reports orbit evaluation at $R = 2$. Restricting the paired shift-penalty comparison to final checkpoints gives 0.1755 with 95% CI $[0.1371, 0.2082]$.

At $R = 2$, the search improves over the best of 32 random rescalings on 99.2% of raw checkpoints. Larger budgets find stronger raw and augmentation attacks, with only a small gain from the final budget doubling. These values remain lower bounds on the worst-case error.

G.2 MLP SELECTION AND INTERVENTION CONSISTENCY

We use the final checkpoint from each of the 16 MLP test runs and ablate the lowest-scoring hidden units under raw incident strength, canonicalized strength, or invariant gate saliency. Gate saliency is computed on a fixed 512-example training calibration set, and interventions are evaluated on 2048 test examples. For hidden unit v, raw and canonical strength sum the absolute incoming, bias, and outgoing parameters. Gate saliency sums $|w_e \partial J/\partial w_e|$ over the same incident edges. Ties are resolved by a fixed coordinate order after a relative tolerance of 10^{-10}. Ablation zeros each selected unit's outgoing weights.

For two masks M, M' of sizes k, k' among n units, the chance-adjusted overlap is

$$\frac{|M \cap M'| - kk'/n}{\min(k, k') - kk'/n},$$

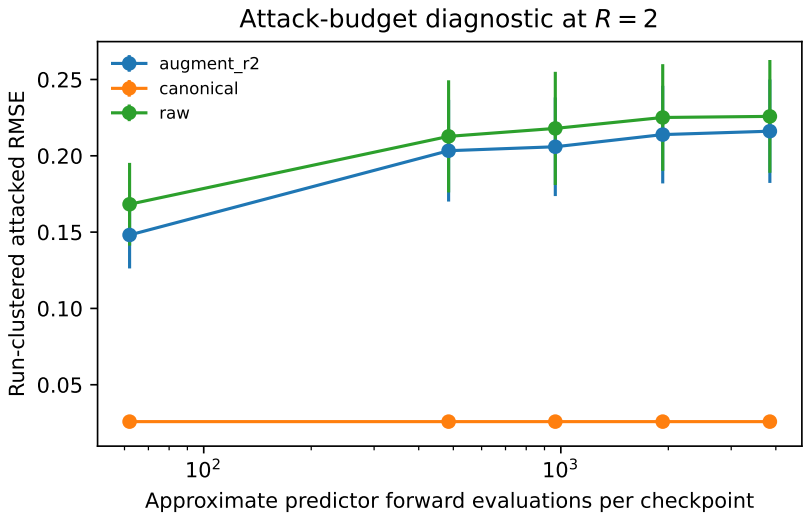


Figure 6: Attacked RMSE versus approximate predictor evaluations per checkpoint, averaged over three predictor initializations. Error bars show variation over initializations. These descriptive averages differ from the run-clustered main-text aggregation. Larger budgets increase raw and augmented errors; canonical errors remain unchanged.

with value one when the denominator vanishes. Metrics are computed layerwise and averaged. The relative logit discrepancy is

$$D_{\text{int}} := \frac{\|Z_{\text{independent}} - Z_{\text{transported}}\|_F}{\max\{\|Z_{\text{transported}}\|_F, 10^{-15}\}}.$$

We use 16 shared random transformations at $R = 2$. "Mean" averages all transformations. "Stress" selects, for each run and fraction, the common transformation with the largest raw-strength intervention discrepancy and evaluates every method on that transformation.

The unpruned final-checkpoint accuracy is 0.880. We report the primary 25% setting below. Figure 7 gives the full curves.

At 25% ablation, raw strength has mean Spearman correlation 0.329, chance-adjusted overlap 0.282, and relative intervention discrepancy $D_{\text{int}} = 1.012$. Under the common stress transformation, $D_{\text{int}} = 1.359$ with 95% CI $[1.151, 1.568]$. Canonical strength and gate saliency each give identical masks across equivalent representatives and intervention discrepancy at numerical precision.

Under the stress transformation, the mean absolute paired NLL difference is 0.509 (95% CI $[0.349, 0.681]$), and the mean absolute paired accuracy difference is 0.123 (95% CI $[0.091, 0.156]$). These are paired differences, not differences between separately averaged accuracies. The corresponding differences for canonical strength and gate saliency are numerical zero.

Consistency and pruning quality are distinct. Post-ablation accuracies on the untransformed models are 0.501 for raw strength, 0.529 for canonical strength, and 0.616 for gate saliency. Canonicalization makes the same strength-based graph score representative independent; gate saliency is the stronger pruning rule in this experiment.

Table 4: Head matching and ablation at GL-100 on 12 held-out runs. Chance head-matching accuracy is $1/12 = 0.083$. Overlap refers to the most-important 10% head set. Both error columns compare ablation using the inferred assignment with ablation using the known permutation (oracle).

Method	Head match	Important overlap	Excess logit error	$\lvert\Delta\text{NLL}\rvert$
Raw factors	0.109	0.652	0.875	0.811
Balanced factors	0.109	0.652	0.881	0.856
Learned augmentation	0.107	0.601	0.960	1.102
Operator products	1.000	1.000	0	0

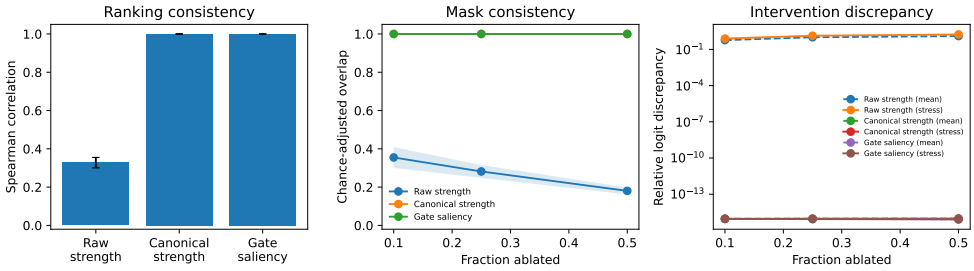


Figure 7: Selection and intervention consistency. Ranking and overlap average 16 transformations per run. At each sparsity and for each run, the stress curve uses the common transformation that maximizes raw-strength discrepancy; all methods are evaluated on that same transformation. Shaded regions are run-clustered 95% confidence intervals.

G.3 Trained-transformer transport: numerical results and controls

Section 7.2 gives the trained-transformer setup and matching procedure. Table 4 reports the full numerical comparison at the primary GL-100 condition; Figure 5 in the main text shows all transformation conditions.

Run variation and secondary selection. The 40 fine-tuning runs vary learning rate, weight decay, label smoothing, augmentation strength, and seed. The calibration ranking also selects the least-damaging 20% of heads, evaluated with 16 size-matched random controls on the same 2,048 disjoint examples.

Ablation controls. Ablating the most-important 10% raises NLL by 2.733 and lowers accuracy from 0.914 to 0.386, versus a mean NLL increase of 0.336 for random sets. It is more damaging than all 16 random controls on every test run. Ablating the least-damaging 20% raises NLL by 0.341 and leaves accuracy 0.837, versus a random-set NLL increase of 0.916; it is less damaging than all controls on 10 of 12 runs. Matching errors therefore change the effect of an ablation, even between checkpoints that compute the same function.

Numerical agreement. For each test checkpoint, we generate four transformed copies per condition; the first two are also used for interventions. Before ablation, relative logit error must be below 5×10^{-5}. All transformations pass; the maximum error is 3.50×10^{-5} overall and 1.51×10^{-5} at GL-100. For the most-important head set at GL-100, the oracle's logit discrepancy averages 9.74×10^{-6}, with maximum 1.63×10^{-5}. Zero transport error means agreement with the oracle assignment, not exact floating-point equality between checkpoints.

G.4 Thresholding and numerical rank

These sweeps are evaluated separately at each support or numerical rank, not as one continuous exact representation across transitions.

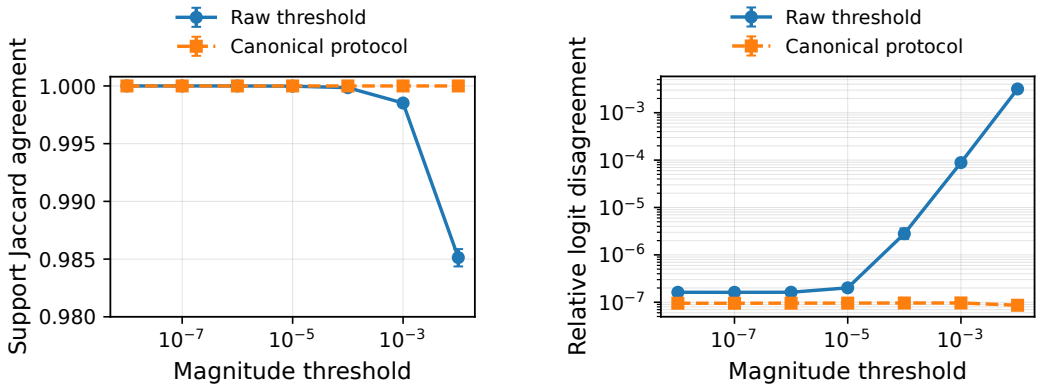


Figure 8: Thresholding across positive-rescaling orbits. **Left:** support agreement. **Right:** relative logit disagreement. Canonicalization before thresholding makes the selected support representative independent at every tested tolerance.

Numerical tolerances. On a fixed support, let $z = \Pi_S y$ and suppose a numerical solve gives $\widehat{z} = y - B_S^{\mathsf{T}}\widehat{\alpha}$. For $B_S \neq 0$, write $L_S = B_S B_S^{\mathsf{T}}$ and let $\lambda_+(L_S)$ be its smallest positive eigenvalue. Since $\widehat{z} - z \in \operatorname{im} B_S^{\mathsf{T}}$,

$$\|\widehat{z} - z\|_\infty \leq \frac{\|B_S \widehat{z}\|_2}{\sqrt{\lambda_+(L_S)}} =: \delta_{\mathrm{can}}. \tag{46}$$

This follows by inverting B_S on that subspace. A threshold decision is certain when $|\widehat{z}_e - \log \tau| > \delta_{\mathrm{can}}$; closer entries require more precision or an explicit uncertainty flag. For $B_S = 0$, the projection is the identity.

For a computed matrix $\widehat{M} = M + E$ with $\|E\|_{\mathrm{op}} \leq \delta$, the standard singular-value perturbation bound gives $|\sigma_i(\widehat{M}) - \sigma_i(M)| \leq \delta$. At an absolute threshold t, the number of singular values of M above t lies between $\#\{i : \widehat{\sigma}_i > t + \delta\}$ and $\#\{i : \widehat{\sigma}_i > t - \delta\}$. This bounds a thresholded *numerical* rank, not the exact rank. Thresholding a raw factor's singular values also need not respect a general linear gauge. The experiment below thresholds the invariant product operators instead.

Support transitions. We train 16 two-hidden-layer ReLU MLPs on Fashion-MNIST and apply 16 exact positive rescalings of radius $R = 2$ to each model. For thresholds $\tau \in \{10^{-8}, \ldots, 10^{-2}\}$, we compare raw magnitude thresholding with

$$\mathrm{Can}_+ \longrightarrow \text{threshold at } \tau \longrightarrow \mathrm{Can}_+ \text{ on the remaining support.}$$

Canonicalizing before thresholding gives support Jaccard agreement 1.000 and relative logit disagreement below 1.0×10^{-7} at every threshold. Raw thresholding becomes representative dependent: at $\tau = 10^{-2}$, mean support agreement is 0.985 and relative logit disagreement is 3.17×10^{-3}. The procedure is applied separately on each selected support; it does not give continuous exact coordinates across zero weights.

Numerical rank. On four held-out ViT-B/16 checkpoints, we compute thin SVDs of the affine query–key and value–output operators and declare numerical rank

$$r_\tau = \#\{i : \sigma_i/\sigma_1 > \tau\}.$$

We sweep τ from 10^{-8} to 10^{-2}, report the best rank-r_τ operator error and the local spectral boundary gap, and mark a rank decision ambiguous when that gap is below 10. The selected ranks remain at the latent dimension through $\tau = 10^{-6}$. At $\tau = 10^{-4}$, the mean query–key rank is $63.71/64$, mean relative operator error is 5.39×10^{-6}, and 6.9% of boundaries are ambiguous. At $\tau = 10^{-2}$, the corresponding values are 59.81, 1.92×10^{-3}, and 34.0%; value–output operators remain nearly full rank. Downstream NLL and accuracy are unchanged to the reported precision over the sweep. The spectral-gap flag is a heuristic for this sweep, not a certificate based on a measured backward-error bound.

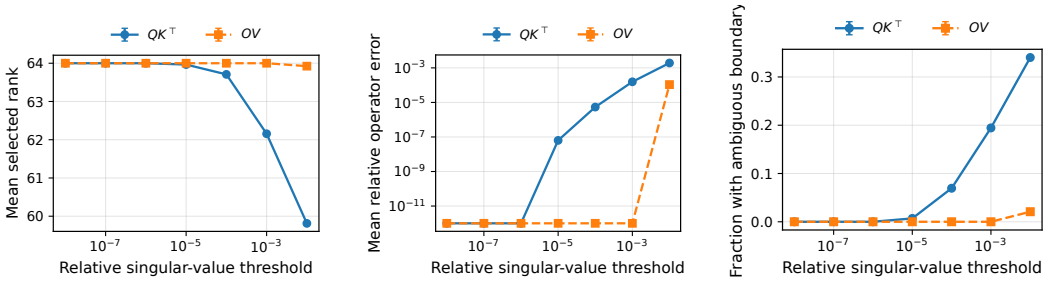


Figure 9: Numerical-rank sensitivity on real ViT attention operators. **Left:** selected rank. **Middle:** relative operator approximation error. **Right:** fraction of spectral boundaries marked ambiguous.

Table 5: CPU scaling of affine attention feature extraction, exact gauge application, and three matching rules.

Model	Params (M)	Features (s)	Gauge (s)	Match (s)	Pair total (s)
DeiT-T	5.7	0.035	0.016	0.003	0.090
DeiT-S	22.1	0.098	0.053	0.004	0.253
ViT-B	85.8	0.351	0.224	0.010	0.936

G.5 Computational cost

Transformer operator pipeline. The attention feature, transformation, and matching pipeline takes 0.936 s for a pair of 85.8M-parameter ViT-B checkpoints on CPU. Table 5 reports medians over five CPU repetitions with eight requested cores. "Pair total" is 2× feature extraction plus one gauge application and Hungarian matching for raw, balanced, and operator features. The median incremental process RSS is 8.2, 28.4, and 231.3 MiB for DeiT-T, DeiT-S, and ViT-B, respectively; this quantity includes allocator and cache effects and is not a hardware-independent peak-memory estimate.

Sparse rescaling canonicalization. Table 6 benchmarks canonicalization on layered ReLU networks. We use Jacobi-preconditioned conjugate gradients; the largest case takes 9 iterations and has projection residual 1.49×10^{-15}. The CG time covers the solve only. Direct-factorization times are amortized across checkpoints sharing a support and include the right-hand side and final projection.

G.6 Resuming training from equivalent checkpoints

To test whether the same current function implies the same behavior after further training, we start from four held-out ViT-B/16 checkpoints and exactly equivalent representatives, initialize fresh AdamW optimizers, and apply 25 updates using identical fixed minibatches. Before optimization, the mean relative logit disagreement is between 4.24×10^{-7} and 6.94×10^{-7}. After the identical updates, it is 2.77×10^{-4} for a head permutation, 7.42×10^{-3} for an orthogonal gauge, 1.54×10^{-2} for GL-10, and 1.55×10^{-1} for scalar-100; the corresponding scalar-100 NLL difference is 0.0701. The current functions agree to numerical precision, but the subsequent training trajectories need not. An invariant representation of the current function can therefore discard information relevant to predicting future optimization.

Table 6: Sparse incidence–Laplacian scaling. The direct solver is skipped for the largest case by a predeclared size guard.

Architecture	Moving edges	Build (s)	CG solve (s)	Direct amort. (s)	RSS (MiB)
MLP-32×3	27,552	0.0021	0.0002	0.0001	92.7
MLP-128×4	151,296	0.0119	0.0006	0.0012	105.5
MLP-256×6	532,480	0.0496	0.0037	0.0044	172.0
MLP-512×8	2,245,632	0.3025	0.0264	–	416.2