VECTOR-VALUED QUANTITATIVE BIPOLAR ARGUMENTATION FOR NODE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Node classification combines evidence from node attributes with potentially conflicting information from graph neighbors. We develop an end-to-end trainable architecture based on vector-valued quantitative bipolar argumentation frameworks (QBAFs). Feature-grounded arguments carry class-specific evidence, while dynamic support, neutral, and attack relations determine how that evidence revises neighboring arguments and class claims. Individual and jointly composed evidence contribute through the same role algebra, and prediction uses only the accumulated terminal claims. These operations expose feature contributions, relational actions, and score increments within the predictive computation itself. We prove that single-channel role interventions have class-specific spatial effects and derive downstream score bounds with conditions for preserving the predicted class. Against 16 comparison methods, QBAF achieves the highest tabulated mean on a majority of the benchmarks—six of nine datasets. Forty-eight matched ablation runs establish the contribution of local argument composition and measure the effects of relational conditioning and class weighting. Recorded traces reconstruct predictions, and controlled interventions identify evidence sustaining correct decisions or contributing to errors. The architecture connects graph learning and quantitative argumentation through a prediction mechanism whose reasoning can be reconstructed and intervened upon.

Keywords: graph representation learning, node classification, quantitative bipolar argumentation, interpretable deep learning

1 INTRODUCTION

Node classification requires deciding how evidence from a node's attributes and its graph neighbors should affect competing class claims. Attention learns neighbor-aggregation weights (Veličković et al., 2018; Brody et al., 2022), and adaptive propagation learns how different graph distances contribute (Chien et al., 2021). Explaining the resulting decision poses a further question: *which evidence acted on which claim, through what relational operation?*

Graph explanation methods recover useful evidence through feature and graph masks (Ying et al., 2019; Luo et al., 2020) or jointly learned explanatory subgraphs (Miao et al., 2022). Quantitative argumentation offers a complementary starting point: supporting and attacking relations are explicit components of the reasoning process (Cayrol & Lagasquie-Schiex, 2005; Baroni et al., 2018). Neural connections include argumentation interpretations of multilayer perceptrons (Potyka, 2021; Ayoobi et al., 2023) and learned case-based argumentative classifiers (Gould & Toni, 2025; 2026). For graph prediction, this perspective requires evidence that preserves its feature origin and class identity, relations that respond to both endpoints, and revisions that account for the final score.

We introduce *vector-valued quantitative bipolar argumentation* (QBAF) for node classification. Nodes maintain feature-grounded vector arguments and accumulated class claims. Each layer composes individual and joint evidence, transmits it through source–target support, neutral, and attack roles, and revises the claims. The role and the signed payload play distinct parts: the payload carries evidence, and the role governs its action. Terminal claims directly predict the class. Consequently, the quantities responsible for prediction also support reconstruction and controlled intervention.

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.

Our contributions are:

- **A predictive argumentation framework.** Class-indexed vector arguments, dynamic bipolar roles, and cumulative claims connect quantitative argumentation to end-to-end graph learning.
- **Intervention guarantees.** We prove class-specific spatial localization for single-channel role interventions and bound downstream score changes, yielding explicit conditions for preserving the winning class.
- **Predictive and explanatory evidence.** QBAF achieves the highest tabulated mean on a majority of the benchmarks: six of nine datasets. Forty-eight matched ablations test the argumentative operations; reconstructed traces and relational interventions explain both correct predictions and errors.

2 Related Work

Learning with arguments. Abstract and bipolar argumentation formalize attacks and supports (Dung, 1995; Cayrol & Lagasquie-Schiex, 2005); gradual semantics assign numerical strengths (Baroni et al., 2018). Neural argumentation connects these ideas to perceptrons, case-based classifiers, and concept-based models (Potyka, 2021; Gould & Toni, 2026; Chi et al., 2026). QBAF instead places class-indexed vector arguments on an observed graph and learns the relations that revise them.

Graph propagation and explanation. Signed propagation and adaptive hop weights address heterogeneous graph evidence (Bo et al., 2021; Chien et al., 2021); joint denoising and rewiring aligns graph and feature spectral spaces (Linkerhägner et al., 2025). Graph explainers identify influential masks, subgraphs, or coalitions (Ying et al., 2019; Luo et al., 2020; Yuan et al., 2021), while intrinsic approaches learn explanatory structure with the predictor (Miao et al., 2022). QBAF makes evidence, role action, and score revision the predictive operations themselves. Appendix C develops these connections.

3 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ have N nodes, features $X \in \mathbb{R}^{N \times F}$, and labels $y_i \in \{0, \dots, C-1\}$. In transductive node classification, the graph and attributes are available during learning. Disjoint sets $\mathcal{V}_{\mathrm{tr}}$, $\mathcal{V}_{\mathrm{val}}$, and $\mathcal{V}_{\mathrm{te}}$ specify training, validation, and evaluation nodes. Training labels supervise updates; validation labels select the model.

Node i maintains vector arguments $H_i^{(\ell)} \in \mathbb{R}^{K \times C}$ and accumulated class scores $q_i^{(\ell)} \in \mathbb{R}^C$. Each argument coordinate carries signed evidence for all classes, whereas $q_{ic}^{(\ell)}$ scores the claim that $y_i = c$. Both start from the node's own features:

$$H_i^{(0)} = \Phi_\theta(x_i), \qquad q_i^{(0)} = \Psi_\theta(x_i). \tag{1}$$

Each directed interaction $j \to i$, layer ℓ, head g, and class c has roles $\rho_{ji,g,c}^{(\ell)} = (p_S, p_N, p_A) \in \Delta^2$, the probability simplex over support, neutral, and attack. Current source and target states determine these roles. Their action on signed payload u is $2(p_S - p_A)u$: support preserves its sign, attack reverses it, and neutrality suppresses it.

Argument-to-claim relations turn updated arguments into revisions $r_i^{(\ell)} \in \mathbb{R}^C$, scaled by learned $\gamma_\ell \in \mathbb{R}$:

$$q_i^{(L)} = q_i^{(0)} + \sum_{\ell=1}^{L} \gamma_\ell r_i^{(\ell)}, \qquad \widehat{y}_i = \arg\max_c q_{ic}^{(L)}. \tag{2}$$

Binary ranking uses $q_{i1}^{(L)} - q_{i0}^{(L)}$. Thus, the terminal claims and the recorded reasoning share the same feature evidence, relation roles, and accumulated revisions.

4 VECTOR-VALUED QUANTITATIVE BIPOLAR ARGUMENTATION

We formulate node classification as the cumulative revision of class claims by feature-grounded arguments. Three criteria organize the construction: *feature grounding* connects initial arguments to observed attributes; *explicit relational action* separates the evidence carried by an interaction from its supporting, neutral, or attacking role; and *decision completeness* accounts for the final scores through successive argument revisions. Our central design choice is to carry argument content and relational roles separately, then accumulate their effects in the scores used for classification. Each layer composes local evidence, exchanges it through source–target relations, and revises class claims. This argumentation trajectory supplies both the prediction and the quantities needed to analyze relational interventions.

4.1 GROUNDING VECTOR ARGUMENTS IN NODE ATTRIBUTES

We ground arguments in two complementary forms of evidence: reusable feature patterns and signed contributions to individual classes. We learn $K = 64$ raw predicates $z_i = \mathrm{ReLU}(x_i U + b_U)$, with $U \in \mathbb{R}^{F\times K}$ and $b_U \in \mathbb{R}^K$. Four heads, indexed by $g \in \{0,1,2,3\}$, also collect signed feature-to-class contributions:

$$s_{igc} = \sum_{\substack{0\le f<F\\ f \bmod 4 = g}} x_{if} W^{\mathrm{src}}_{fc} + \frac{b^{\mathrm{src}}_c}{4}, \qquad H^{(0)}_{ikc} = \begin{cases} z_{ik}, & 0 \le k < 56, \\ s_{igc}/2, & k = 56+g \text{ or } k = 60+g. \end{cases} \tag{3}$$

Here $W^{\mathrm{src}} \in \mathbb{R}^{F\times C}$ and $b^{\mathrm{src}} \in \mathbb{R}^C$ are learned. The first 56 coordinates provide predicates shared across classes; the last eight embed four class-specific contributions as two half-valued copies. Summing these last eight coordinates recovers $x_i W^{\mathrm{src}} + b^{\mathrm{src}}$ exactly. Both forms of evidence thus enter the same argument space with an explicit feature origin.

The initial claims combine direct feature evidence with all K raw predicates:

$$q^{(0)}_i = \gamma_0\big(x_i W^{\mathrm{self}} + b^{\mathrm{self}} + z_i W^{\mathrm{pred}}\big), \tag{4}$$

where $W^{\mathrm{self}} \in \mathbb{R}^{F\times C}$, $b^{\mathrm{self}} \in \mathbb{R}^C$, and $W^{\mathrm{pred}} \in \mathbb{R}^{K\times C}$ are learned. The shared projection U grounds argument construction and initial claims in the same feature predicates. The source and self mappings are separate parameters, allowing transmitted evidence and initial judgments to be learned jointly.

4.2 A COMMON ALGEBRA FOR ARGUMENT ROLES

Explicit relational action separates a message's content from the rule governing its effect. For a context vector $v \in \mathbb{R}^{d_\eta}$, relation family η assigns roles through

$$\mathcal{R}_\eta(v) = \mathrm{softmax}_{S,N,A}\left(\omega_\eta + \beta_\eta + \frac{A_\eta v}{\sqrt{d_\eta}}\right), \qquad \nu(\rho) = 2(\rho_S - \rho_A). \tag{5}$$

The learned quantities are $\beta_\eta \in \mathbb{R}^3$ and $A_\eta \in \mathbb{R}^{3\times d_\eta}$; ω_η is a fixed role offset. Contexts contain explicit source values, target values, and their products. Softmax assigns the three roles within each interaction.

For a signed payload u, the resulting action is $\nu(\rho)u$: support preserves the payload's sign, attack reverses it, and pure neutrality suppresses its direct effect. The three pure roles give $2u$, $-2u$, and 0, respectively. Swapping support and attack reverses the action, and every mixture satisfies $|\nu(\rho)u| \le 2|u|$. This common algebra gives local composition, graph exchange, and claim revision the same operational semantics.

4.3 COMPOSING LOCAL EVIDENCE AND RELATIONAL MESSAGES

Individual and joint arguments. An argument can matter on its own or through its interaction with another argument. We retain both routes. At layer $\ell \in \{1,\dots,L\}$, let $b_{ikc} = \tanh H^{(\ell-1)}_{ikc}$. For $r \in \{0,\dots,5\}$, define the fixed partner $p_r(k) = k \oplus 2^r$, where $\oplus$ is bitwise XOR on the argument

coordinate. These partners form a connected, degree-six hypercube over the 64 coordinates. This topology gives every argument a fixed, traceable set of interaction partners, enabling sparse composition with six neighbors per coordinate rather than 63. Coordinates index learned evidence; the cube specifies their interaction structure. Atomic and joint payloads are

$$a_{ikcr} = b_{i,p_r(k),c}, \qquad j_{ikcr} = b_{i,p_r(k),c}\, b_{i,p_{(r+1) \bmod 6}(k),c}. \tag{6}$$

With $\psi(u,v) = (u, v, uv)^\top$, the composed state is

$$\begin{aligned} \widetilde{H}^{(\ell)}_{ikc} = H^{(\ell-1)}_{ikc} &+ \frac{1}{\sqrt{6}} \sum_{r=0}^{5} \nu(\mathcal{R}_{\mathrm{at},r}(\psi(a_{ikcr}, b_{ikc}))\, a_{ikcr} \\ &+ \frac{1}{\sqrt{6}} \sum_{r=0}^{5} \nu(\mathcal{R}_{\mathrm{jt},r}(\psi(j_{ikcr}, b_{ikc}))\, j_{ikcr}. \end{aligned} \tag{7}$$

Atomic roles evaluate one source argument against the receiving argument; joint roles evaluate a two-argument product against that same receiver. The atomic route keeps individual evidence available when a product vanishes, while the joint route expresses interactions beyond the sum of its two inputs. Each partner has separate atomic and joint parameters, shared across nodes, classes, coordinates, and layers, with $\omega_{\mathrm{at}} = \omega_{\mathrm{jt}} = \log(1/4, 1/2, 1/4)$.

Source–target graph roles. A graph relation must account for the evidence offered by its source and the state receiving that evidence. We assign coordinate k to head $g(k) = k \bmod 4$ and compute

$$\bar{b}^{(\ell)}_{igc} = \frac{1}{16} \sum_{k:g(k)=g} \tanh \widetilde{H}^{(\ell)}_{ikc}, \qquad t^{(\ell)}_{ic} = \tanh q^{(\ell-1)}_{ic}. \tag{8}$$

For edge $j \to i$, the six-dimensional context is

$$\begin{aligned} \chi^{(\ell)}_{ji,g,c} &= \left(\bar{b}^{(\ell)}_{jgc},\, \bar{b}^{(\ell)}_{igc},\, \bar{b}^{(\ell)}_{jgc}\bar{b}^{(\ell)}_{igc},\, t^{(\ell)}_{jc},\, t^{(\ell)}_{ic},\, t^{(\ell)}_{jc}t^{(\ell)}_{ic}\right)^\top, \\ \rho^{(\ell)}_{ji,g,c} &= \mathcal{R}_{\mathrm{gr},\ell,g,c}(\chi^{(\ell)}_{ji,g,c}). \end{aligned} \tag{9}$$

Graph-role parameters are specific to the layer, head, and class, with $\omega_{\mathrm{gr}} = \log(5/8, 1/4, 1/8)$. The relation can change its role as evidence and judgments evolve at either endpoint. Its linear–bilinear context makes that dependence explicit for every class.

Let $\bar{A}_{ji}$ count directed $j \to i$ edge entries after setting each diagonal entry to one. Define the incoming degree $d_i = \sum_j \bar{A}_{ji}$, normalized weight $\alpha_{ji} = \bar{A}_{ji}/\sqrt{d_j d_i}$, and edge support $\overline{\mathcal{E}} = \{(j,i) : \bar{A}_{ji} > 0\}$. Graph revision is

$$\begin{aligned} m^{(\ell)}_{j\to i,k,c} &= \alpha_{ji}\, \nu\Big(\rho^{(\ell)}_{ji,g(k),c}\Big) \tanh \widetilde{H}^{(\ell)}_{jkc}, \\ H^{(\ell)}_{ikc} &= \widetilde{H}^{(\ell)}_{ikc} + \sum_{j:(j,i)\in\overline{\mathcal{E}}} m^{(\ell)}_{j\to i,k,c}. \end{aligned} \tag{10}$$

Head summaries determine the relational context; messages carry the full 64-coordinate argument state and preserve the identity of the class being revised. This separation shares role computation across related coordinates while retaining their distinct evidence throughout propagation.

4.4 Revising claims and explaining the resulting decision

The final decision is built from the same arguments that graph relations revise. After graph propagation, argument-to-claim roles evaluate the updated evidence against the current judgment:

$$\begin{aligned} \rho^{(\ell)}_{\mathrm{dec},ikc} &= \mathcal{R}_{\mathrm{dec},k,c}\Big(\psi(\tanh H^{(\ell)}_{ikc}, \tanh q^{(\ell-1)}_{ic})\Big), \\ r^{(\ell)}_{ic} &= \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \nu(\rho^{(\ell)}_{\mathrm{dec},ikc}) \tanh H^{(\ell)}_{ikc}, \qquad q^{(\ell)}_{ic} = q^{(\ell-1)}_{ic} + \gamma_\ell r^{(\ell)}_{ic}. \end{aligned} \tag{11}$$

Decision-role parameters are indexed by coordinate and class and shared across layers, with $\omega_{\mathrm{dec}} = 0$. The learned coefficient $\gamma_\ell \in \mathbb{R}$ scales the revision; its sign, the role coefficient, and the payload jointly determine the decision contribution. Prediction uses $q^{(L)}$ directly, as in Eq. 2.

Define the recorded decision contribution of coordinate k by

$$\xi_{ikc}^{(\ell)} = \frac{\gamma_\ell}{\sqrt{K}}\, \nu(\rho_{\mathrm{dec},ikc}^{(\ell)}) \tanh H_{ikc}^{(\ell)}. \tag{12}$$

These contributions expose how successive arguments change the margins between class claims. For a fixed pair c, a, their sum reconstructs the decision margin:

$$q_{ic}^{(L)} - q_{ia}^{(L)} = q_{ic}^{(0)} - q_{ia}^{(0)} + \sum_{\ell=1}^{L} \sum_{k=0}^{K-1} \left(\xi_{ikc}^{(\ell)} - \xi_{ika}^{(\ell)} \right). \tag{13}$$

The same forward pass records feature contributions, local compositions, edge roles, and transmitted payloads. These records connect the score decomposition to the interactions that generated it. Appendix A.3 proves reconstruction and node-relabeling equivariance of the full trace.

From recorded contributions to intervention effects. To determine how a relation affects the decision, we replace its role by $(0, 1, 0)$ and recompute the downstream states with fixed parameters and graph normalization. The next result locates the affected decisions and bounds their changes, accounting for the resulting changes in subsequent roles.

Proposition 1 (Localized and bounded role interventions)**.** *Consider deterministic evaluation in real arithmetic with fixed finite inputs and parameters. Neutralize the role of edge $j \to i$, head g, and class c at layer $s \in \{1, \dots, L\}$. Its direct argument perturbation has maximum absolute coordinate change*

$$\delta = \alpha_{ji} |\nu(\rho_{ji,g,c}^{(s)})| \max_{k:g(k)=g} |\tanh \widetilde{H}_{jkc}^{(s)}|. \tag{14}$$

All other classes' claim scores remain unchanged. For class c, only nodes reachable from i in at most $L - s$ directed edges can change. Let $\mathsf{M}_{\ell,c}$ be the nonnegative layer-sensitivity matrices and $u_{g,c}$ the decision sensitivity defined explicitly in Appendix A.4. With $\mathbf{e}_i \in \mathbb{R}^N$ the ith coordinate vector,

$$\left| q_{:c}^{(L),\mathrm{neut}} - q_{:c}^{(L)} \right| \preceq \underbrace{[0 \quad I_N]\, \mathsf{M}_{L,c} \cdots \mathsf{M}_{s+1,c} \begin{bmatrix} \delta \mathbf{e}_i \\ |\gamma_s| u_{g,c} \delta \mathbf{e}_i \end{bmatrix}}_{\varepsilon \in \mathbb{R}^N_{\geq 0}}, \tag{15}$$

where $\preceq$ is componentwise order and an empty matrix product is the identity. At node v, a unique winning class a is preserved whenever

$$\begin{cases} q_{vc}^{(L)} - \max_{b \neq c} q_{vb}^{(L)} > \varepsilon_v, & a = c, \\ q_{va}^{(L)} - q_{vc}^{(L)} > \varepsilon_v, & a \neq c. \end{cases} \tag{16}$$

Proof sketch. Every state update preserves the class index, and each layer traverses at most one graph edge, establishing class and spatial locality. For a role-context coordinate v_t, softmax gives $|\partial \nu(\mathcal{R}_\eta(v))/\partial v_t| \leq (\max_r A_{\eta,rt} - \min_r A_{\eta,rt})/\sqrt{d_\eta}$, where r indexes the three roles. Combining this bound with the unit Lipschitz constant of $\tanh$ and the product rule yields the nonnegative matrices $\mathsf{M}_{\ell,c}$. Propagating the perturbation in Eq. 14 proves Eq. 15; the unchanged competing scores give the margin conditions. Appendix A.4 supplies all constants and the full proof. □

The bound separates the evidence removed at the intervened relation from its subsequent amplification by learned reasoning steps. It connects an edge-level role to a class-specific decision effect through the actual propagation equations.

4.5 Learning arguments and relations jointly

Supervision acts on the final accumulated claims, so argument construction, role assignment, and decision revision learn through the same prediction objective. With $P_{\theta,ic} = \mathrm{softmax}(q_i^{(L)})_c$, we

minimize

$$\mathcal{L}_t(\theta) = -\frac{1}{|\mathcal{V}_{\text{tr}}|} \sum_{i \in \mathcal{V}_{\text{tr}}} w_{y_i}^{(t)} \log P_{\theta,iy_i}, \qquad w_c^{(t)} = \begin{cases} 1, & t \le 100, \\ 1 + \lambda(w_c - 1), & t > 100. \end{cases} \tag{17}$$

Here t indexes optimizer updates. After update 100, detached evaluation-mode predictions on non-training nodes estimate their class prior $\widehat{\pi}_U$ using 50 expectation–maximization updates initialized at the training prior π_T (Saerens et al., 2002). We form $w_c \propto \widehat{\pi}_{U,c}/\pi_{T,c}$, normalized so that $\sum_c \pi_{T,c} w_c = 1$, and fix these weights for the remaining updates; λ controls their strength. Adam (Kingma & Ba, 2015) continues with the same optimizer state across the two phases. The highest validation metric selects the checkpoint, and prediction uses its final claim scores $q^{(L)}$. Appendix A.5 specifies the estimator and selection rules; Appendix A.1 gives parameter initialization and sharing.

5 Experiments

We evaluate QBAF through node classification, matched component ablations, and decision-level reconstruction and intervention. The nine-dataset comparison measures predictive effectiveness; the ablations and traces connect these results to explicit evidence construction, relational action, and cumulative claim revision.

5.1 Experimental setup

Data and evaluation. We study Cora, Citeseer, Pubmed, Amazon Photo, WikiCS, Coauthor Physics, Actor, Texas, and Questions. Together, these datasets cover citation, co-purchase, co-authorship, hyperlink, and social graphs, including both homophilous and heterophilous benchmarks. For Cora, Citeseer, Pubmed, Photo, WikiCS, Physics, Actor, and Texas, we use class-quota random splits: each class contributes at most $\mathrm{round}(0.6N/C)$ training nodes, validation draws $\mathrm{round}(0.2N)$ nodes from the remaining pool, and the remaining nodes form the evaluation set. We use five fixed splits and two initializations per split. Questions uses five official masks with two initializations per mask (Platonov et al., 2023). Questions is evaluated by ROC-AUC using the continuous final margin $q_{i1}^{(L)} - q_{i0}^{(L)}$; the other eight datasets use accuracy. Dataset statistics and the exact repetition structure are given in Appendix B.

Learning and model selection. All datasets use the same dense QBAF construction: 64 argument coordinates, four heads, six XOR partners, source–target role functions, and prediction from $q^{(L)}$. We train with Adam for at most 2,000 updates, clip the gradient norm at 5, and retain the checkpoint with the highest validation metric, keeping the first checkpoint in an exact tie. Early stopping uses 200 updates without improvement in unweighted validation cross-entropy. After update 100, detached predictions on non-training nodes and training-label frequencies determine the weights in Eq. 17; the estimator reads no validation or test labels. Each dataset uses one validation-selected configuration across all its reported runs. Appendix B gives the selected hyperparameters and training details.

Comparisons and statistics. We compare against the 16 established methods in Table 1, spanning graph convolution, adaptive spectral propagation, and graph Transformers. QBAF entries report the mean and sample standard deviation of ten completed runs. Split-level analyses first average the two initializations within each split. Each QBAF column uses a single configuration and all ten of its results.

5.2 Predictive performance of explicit argumentation

QBAF leads the comparison on a majority of the benchmarks, attaining the highest mean on six of nine datasets: Cora, Citeseer, Pubmed, Photo, WikiCS, and Physics. The respective gains over the strongest comparison means are 0.65, 0.27, 0.84, 1.04, 3.23, and 1.05 percentage points. WikiCS reaches 88.38%, Photo 96.53%, and Physics 99.01%; their across-run standard deviations are 0.30, 0.29, and 0.09 points. On Actor, Texas, and Questions, roles act on class-specific evidence: attack reverses a signed payload, while subsequent revisions determine its final decision effect. The matched

Table 1: **Node classification with explicit argumentation.** Accuracy (%) on eight datasets and ROC-AUC (%) on Questions. QBAF reports the mean and sample SD of ten runs; bold marks the largest mean in each column.

Method	Cora	Citeseer	Pubmed	Photo	WikiCS
GCN	87.14 ± 1.01	79.86 ± 0.67	86.74 ± 0.27	88.26 ± 0.73	82.32 ± 0.69
GAT	88.03 ± 0.79	80.52 ± 0.71	87.04 ± 0.24	90.94 ± 0.68	83.22 ± 0.78
H2GCN	87.96 ± 0.37	80.90 ± 1.21	89.18 ± 0.28	95.45 ± 0.67	83.45 ± 0.26
HopGNN	88.68 ± 1.06	80.38 ± 0.68	89.15 ± 0.35	94.49 ± 0.33	84.73 ± 0.59
ChebyNet	86.67 ± 0.82	79.11 ± 0.75	87.95 ± 0.28	93.77 ± 0.32	82.95 ± 0.45
GPRGNN	88.57 ± 0.69	80.12 ± 0.83	88.46 ± 0.33	93.85 ± 0.28	82.58 ± 0.89
BernNet	88.52 ± 0.95	80.09 ± 0.79	88.48 ± 0.41	93.63 ± 0.35	83.56 ± 0.61
JacobiConv	88.98 ± 0.46	80.78 ± 0.79	89.62 ± 0.41	95.43 ± 0.23	84.13 ± 0.49
HiGCN	89.23 ± 0.23	81.12 ± 0.28	89.95 ± 0.13	95.33 ± 0.37	83.14 ± 0.78
Transformer	71.83 ± 1.68	70.55 ± 1.20	86.66 ± 0.50	89.58 ± 1.05	77.36 ± 1.25
GraphGPS	83.42 ± 1.22	75.87 ± 0.71	86.62 ± 0.53	94.35 ± 0.25	79.26 ± 0.57
NodeFormer	87.32 ± 0.92	79.56 ± 1.10	89.24 ± 0.23	95.27 ± 0.22	81.03 ± 0.94
SGFormer	87.87 ± 2.67	79.62 ± 1.63	89.07 ± 0.14	94.34 ± 0.23	82.71 ± 0.56
NAGphormer	88.15 ± 1.35	80.12 ± 1.24	89.70 ± 0.19	95.49 ± 0.11	83.41 ± 0.34
Specformer	88.57 ± 1.01	81.49 ± 0.94	90.61 ± 0.23	95.48 ± 0.32	85.15 ± 0.63
GPRGNN+JDR	89.33 ± 0.25	81.00 ± 0.28	89.24 ± 0.15	94.78 ± 0.08	—
QBAF	$\mathbf{89.98 \pm 1.73}$	$\mathbf{81.76 \pm 2.26}$	$\mathbf{91.45 \pm 0.33}$	$\mathbf{96.53 \pm 0.29}$	$\mathbf{88.38 \pm 0.30}$

Method	Physics	Actor	Texas	Questions
GCN	97.74 ± 0.35	33.23 ± 1.16	77.38 ± 3.28	76.09 ± 1.27
GAT	97.82 ± 0.28	33.93 ± 2.47	80.82 ± 2.13	$\mathbf{77.43 \pm 1.20}$
H2GCN	97.19 ± 0.13	36.31 ± 2.58	91.89 ± 3.93	63.59 ± 1.46
HopGNN	97.86 ± 0.16	39.33 ± 2.79	89.15 ± 4.04	—
ChebyNet	97.25 ± 0.78	37.61 ± 0.89	86.22 ± 2.45	—
GPRGNN	97.25 ± 0.13	39.92 ± 0.67	92.95 ± 1.31	74.36 ± 0.67
BernNet	97.36 ± 0.30	41.79 ± 1.01	93.12 ± 0.65	74.25 ± 0.73
JacobiConv	97.56 ± 0.28	41.17 ± 0.64	93.44 ± 2.13	—
HiGCN	97.65 ± 0.35	41.81 ± 0.52	92.15 ± 0.73	—
Transformer	OOM	39.95 ± 0.64	88.75 ± 6.30	OOM
GraphGPS	97.60 ± 0.05	37.68 ± 0.94	83.71 ± 5.85	71.73 ± 1.47
NodeFormer	96.45 ± 0.28	34.62 ± 1.82	84.63 ± 3.47	74.48 ± 1.32
SGFormer	97.96 ± 0.81	41.69 ± 0.63	92.46 ± 1.48	77.06 ± 1.20
NAGphormer	97.85 ± 0.26	40.08 ± 1.50	91.80 ± 1.85	75.13 ± 0.70
Specformer	97.75 ± 0.53	$\mathbf{41.93 \pm 1.04}$	88.23 ± 0.38	76.49 ± 0.58
GPRGNN+JDR	—	41.89 ± 0.24	$\mathbf{93.85 \pm 0.54}$	73.60 ± 0.86
QBAF	$\mathbf{99.01 \pm 0.09}$	39.89 ± 1.12	89.18 ± 4.03	73.88 ± 3.53

Table 2: **Matched component ablations.** Accuracy (%) averaged over the same two splits and two initializations for each row. Parentheses give the change from the matched full QBAF, in percentage points. Every ablation is trained from initialization; the full-model entries are the corresponding four-run subset of Table 1. "Source only" removes target conditioning in graph roles; "no local composition" removes atomic and joint local updates.

Dataset	Full QBAF	Unit weights	Source only	No local composition
Cora	90.60	87.19 (-3.41)	90.52 (-0.08)	89.53 (-1.07)
Photo	96.69	96.08 (-0.61)	96.67 (-0.01)	96.55 (-0.14)
Texas	92.21	89.34 (-2.87)	92.21 (0.00)	90.57 (-1.64)
Actor	40.18	40.14 (-0.05)	39.65 (-0.54)	39.85 (-0.33)

ablations and Actor interventions below resolve how these operations sustain correct decisions or contribute to errors.

5.3 Which argumentative operations contribute?

We isolate three factors while preserving the matched data, initialization, hyperparameters, training budget, and checkpoint rule. *Unit class weights* sets $\lambda = 0$ while retaining the two-phase training procedure. *Source-only roles* removes target-state conditioning from graph-role assignment. *No local argument composition* removes the local atomic and joint composition step. Each factor is evaluated on Cora, Photo, Texas, and Actor with two splits and two initializations, giving 48 ablation identities. The full-model reference is the corresponding four-run subset of the main experiment.

Composing arguments improves the predictor. Removing local atomic and joint composition decreases mean accuracy on all four datasets: by 1.07 points on Cora, 0.14 on Photo, 1.64 on Texas, and 0.33 on Actor. The strongest effects appear on Cora and Texas, which differ substantially in graph size and label connectivity. This matched comparison supports the role of Eq. 7: transforming individual and joint evidence before graph exchange contributes beyond propagating uncomposed arguments.

Conditioning and weighting have distinct empirical effects. Removing target conditioning produces its largest loss on Actor, at 0.54 points, while Texas has the same four-run mean. Class reweighting has a different profile: unit weights reduce Cora by 3.41 points and Texas by 2.87, compared with 0.61 on Photo and 0.05 on Actor. The matched comparisons separate structural and learning contributions: local composition and endpoint conditioning govern evidence interaction, whereas prior reweighting changes the supervised learning signal. Its substantial gains on Cora and Texas establish reweighting as an effective component of the complete method.

5.4 From argument traces to decision effects

The explanatory object is the computation of the prediction itself. For four Photo and Actor cases, we reconstruct predictions from feature-grounded arguments and layerwise claim updates, then intervene on the same edge roles to measure their effects on class margins. Equation 12 connects the recorded updates to the class-score decomposition in Eq. 13. The analysis follows evidence from its feature origin through a relation to its contribution to a competing class claim.

A prediction reconstructed from its arguments. In the correctly classified Photo case, the margin of the true class over a fixed competitor changes from -0.0250 at $q^{(0)}$ to 8.0630 at $q^{(L)}$. The corresponding Actor margin grows from 0.1089 to 0.8804. Replaying the recorded argument computations reconstructs graph-state updates and final scores for all four cases with zero observed absolute error. Reconstructing the initial margins from feature, predicate, and bias terms has maximum absolute error 1.49×10^{-8}. Thus, the analysis links the learned feature coefficients to the decision through the actual argument updates.

Relational interventions test the reasoning path. For the correctly classified Photo case, neutralizing the selected relation lowers the true-class margin by 2.2822, compared with 1.2747 for the

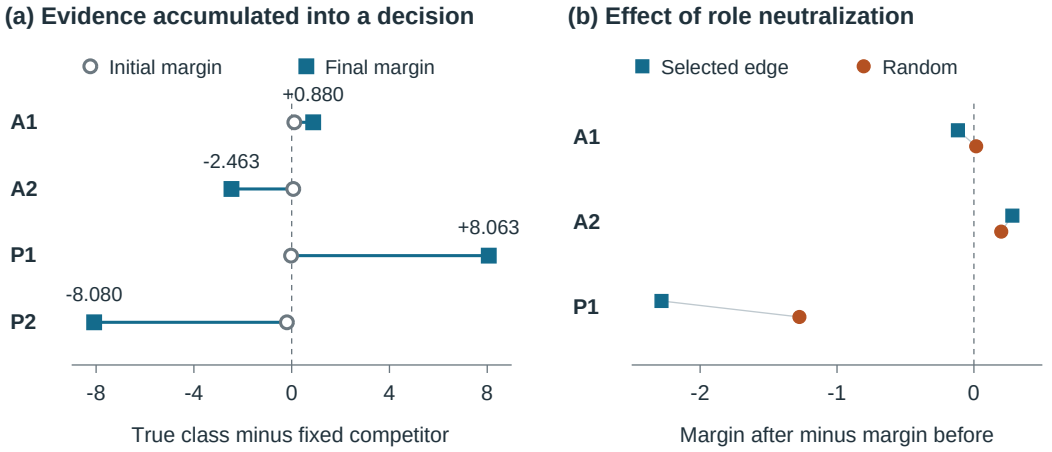


Figure 1: **Argument traces connect evidence to decision effects.** Left: initial and final true-class margins for four fixed validation cases, measured against the same competitor within each case; segments connect the measured endpoints. A1 and P1 are correctly classified Actor and Photo nodes; A2 and P2 are incorrectly classified nodes. Right: change in that margin after neutralizing the recorded selected edge, compared with fixed-seed, equal-budget random controls. Negative values mean the intervention reduces the true-class margin. Points show the recorded means; the control pools for A1, A2, and P1 contain three, one, and one distinct edges, respectively. P2 has no non-self incoming edge. All interventions recompute the downstream states with fixed model parameters.

matched random control. On the correct Actor case, the selected intervention lowers the margin by 0.1168, whereas the random-control mean increases it by 0.0165. For the incorrect Actor case, neutralizing the selected relation increases the true-class margin by 0.2808. The same explicit role operation therefore identifies both evidence sustaining a correct decision and evidence contributing to an error. Recomputing the model after each role change connects the recorded reasoning path to an observable change in the decision margin.

Neutralization and edge removal isolate role action and graph structure. Neutralization preserves normalization; deletion recomputes it. On the correct Photo case, their margin changes are -2.2822 and -0.6809, respectively. We neutralize the selected edge across all layers, heads, and classes; Proposition 1 characterizes the corresponding single-channel operation.

The cost of accessing the reasoning. On an RTX 5090D, median resident-model inference takes 79.5 ms for Photo and 32.0 ms for Actor. Replaying the saved contribution records on a two-thread CPU takes 44.3 ms and 26.1 ms, including trace loading and verification, with no model forward pass. The measured neutralization interface takes 127.5 ms and 59.5 ms on the GPU, including model loading, graph construction, transfers, and output copying. For fixed width and depth, computation and parameter count scale linearly with the number of classes (Appendix A.6). The argument state stores NKC scalars, and edge chunking limits transient message storage. Proposition 1 certifies score changes under role neutralization; replay reconstructs the recorded computation.

6 Conclusion

Vector-valued QBAF turns node classification into learned argument construction, relational action, and cumulative claim revision. Its role algebra supports both prediction and computational explanation, with spatial localization and score bounds for single-channel interventions. Six of nine datasets attain the highest tabulated mean; matched ablations, trace reconstruction, and interventions connect performance to the underlying argument operations. Learning argument structures that transfer across graphs while retaining this computational correspondence is a natural direction for further work.

AI USE STATEMENT

Generative AI tools assisted with code writing and manuscript language polishing and revision. Numerical claims have been verified against recorded experimental outputs, and method descriptions have been verified against the actual implementation. The authors take responsibility for the submitted text, claims, code, and experimental results.

REPRODUCIBILITY STATEMENT

Section 4 specifies the predictive computation and intervention result. Appendix A gives parameter initialization, sharing, complete proofs, and learning details; Appendix B gives data processing, selected configurations, matched ablations, intervention controls, and timing procedures.

REFERENCES

Caren Al Anaissy, Sandeep Suntwal, Mihai Surdeanu, and Srdjan Vesic. On the feasibility of learning bipolar gradual argumentation semantics using neural architectures. In *Agents and Artificial Intelligence: 16th International Conference, ICAART 2024, Revised Selected Papers*, volume 15592 of *Lecture Notes in Computer Science*, pp. 176–190. Springer, 2025. doi: 10.1007/978-3-031-87330-0_9. URL https://link.springer.com/chapter/10.1007/978-3-031-87330-0_9.

Hamed Ayoobi, Nico Potyka, and Francesca Toni. SpArX: Sparse argumentative explanations for neural networks. In *ECAI 2023*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, pp. 149–156. IOS Press, 2023. doi: 10.3233/FAIA230265. URL https://doi.org/10.3233/FAIA230265.

Pietro Baroni, Antonio Rago, and Francesca Toni. How many properties do we need for gradual argumentation? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 1736–1743, 2018. doi: 10.1609/aaai.v32i1.11544. URL https://ojs.aaai.org/index.php/AAAI/article/view/11544.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021. doi: 10.1609/aaai.v35i5.16514. URL https://ojs.aaai.org/index.php/AAAI/article/view/16514.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0pdSt3oyJa1.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2105.14491.

Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571 of *Lecture Notes in Computer Science*, pp. 378–389. Springer, 2005. doi: 10.1007/11518655_33. URL https://link.springer.com/chapter/10.1007/11518655_33.

Jie Chen, Zilong Li, Yin Zhu, Junping Zhang, and Jian Pu. From node interaction to hop interaction: New effective and scalable graph learning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7876–7885, 2023a. URL https://openaccess.thecvf.com/content/CVPR2023/html/Chen_From_Node_Interaction_To_Hop_Interaction_New_Effective_and_Scalable_CVPR_2023_paper.html.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. NAGphormer: A tokenized graph transformer for node classification in large graphs. In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=8KYeilT3Ow.

Haixiao Chi, Dawei Wang, Beishui Liao, Gaojie Cui, and Feng Mao. An interpretable model based on concept and argumentation for tabular data. *Scientific Reports*, 16:987, 2026. doi: 10.1038/s41598-025-30540-1. URL https://www.nature.com/articles/s41598-025-30540-1.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized PageRank graph neural network. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2006.07988.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995. doi: 10.1016/0004-3702(94)00041-X. URL https://www.sciencedirect.com/science/article/pii/000437029400041X.

Adam Gould and Francesca Toni. Neuro-argumentative learning with case-based reasoning. In *Proceedings of the 19th International Conference on Neurosymbolic Learning and Reasoning*, volume 284 of *Proceedings of Machine Learning Research*, pp. 1090–1106. PMLR, 2025. URL https://proceedings.mlr.press/v284/gould25a.html.

Adam Gould and Francesca Toni. Deep arguing. *arXiv preprint arXiv:2605.10569*, 2026. doi: 10.48550/arXiv.2605.10569. URL https://arxiv.org/abs/2605.10569.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. BernNet: Learning arbitrary graph spectral filters via Bernstein approximation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14239–14251, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/76f1cfd7754a6e4fc3281bcccb3d0902-Abstract.html.

Yiming Huang, Yujie Zeng, Qiang Wu, and Linyuan Lü. Higher-order graph convolutional network with flower-petals laplacians on simplicial complexes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12653–12661, 2024. doi: 10.1609/aaai.v38i11.29160. URL https://ojs.aaai.org/index.php/AAAI/article/view/29160.

Timotheus Kampik, Nico Potyka, Xiang Yin, Kristijonas Čyras, and Francesca Toni. Contribution functions for quantitative bipolar argumentation graphs: A principle-based analysis. *International Journal of Approximate Reasoning*, 173:109255, 2024. doi: 10.1016/j.ijar.2024.109255. URL https://doi.org/10.1016/j.ijar.2024.109255.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.02907.

Jonas Linkerhägner, Cheng Shi, and Ivan Dokmanić. Joint graph rewiring and feature denoising via spectral resonance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zBbZ2vdLzH.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e37b08dd3015330dcbb5d6663667b8b8-Abstract.html.

Péter Mernyei and Cătălina Cangea. Wiki-CS: A Wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*, 2020. doi: 10.48550/arXiv.2007.02901. URL https://arxiv.org/abs/2007.02901.

Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15524–15543. PMLR, 2022. URL https://proceedings.mlr.press/v162/miao22a.html.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2302.11640.

Nico Potyka. Interpreting neural networks as quantitative argumentation frameworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6463–6470, 2021. doi: 10.1609/aaai.v35i7.16801. URL https://ojs.aaai.org/index.php/AAAI/article/view/16801.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515, 2022. doi: 10.52202/068431-1054. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/5d4834a159f1547b267a05a4e2b7cf5e-Abstract-Conference.html.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002. doi: 10.1162/089976602753284446. URL https://doi.org/10.1162/089976602753284446.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018. doi: 10.48550/arXiv.1811.05868. URL https://arxiv.org/abs/1811.05868.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10903.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23341–23362. PMLR, 2022. URL https://proceedings.mlr.press/v162/wang22am.html.

Qitian Wu, Wentao Zhao, Zenan Li, David P. Wipf, and Junchi Yan. NodeFormer: A scalable graph structure learning transformer for node classification. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27387–27401, 2022. doi: 10.52202/068431-1986. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/af790b7ae573771689438bbcfc5933fe-Abstract-Conference.html.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. SGFormer: Simplifying and empowering transformers for large-graph representations. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64753–64773, 2023. doi: 10.52202/075280-2826. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/cc57fac10eacadb3b72a907ac48f9a98-Abstract-Conference.html.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12241–12252. PMLR, 2021. URL https://proceedings.mlr.press/v139/yuan21c.html.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/58ae23d878a47004366189884c2f8440-Abstract.html.

A Method Derivations and Learning Details

A.1 Argument initialization and parameter sharing

All feature, argument, and class indices retain the meanings introduced in Sections 3 and 4. The model uses $K = 64$ argument coordinates, $G = 4$ heads, and $R = 6$ local partners. Head g contains coordinates $g, g+4, \ldots, g+60$. The partner relation $p_r(k) = k \oplus 2^r$ changes argument coordinates within a node.

For the source mapping, let $x^+ = \max(x, 0)$ and $x^- = \max(-x, 0)$, and use the same notation for W^{src}. The implemented feature contributions are evaluated as

$$\begin{aligned} P_{igc} &= \sum_{f:f \bmod 4=g} \left(x^+_{if} W^{\mathrm{src},+}_{fc} + x^-_{if} W^{\mathrm{src},-}_{fc} \right) + \frac{b^{\mathrm{src}}_c}{8}, \\ N_{igc} &= \sum_{f:f \bmod 4=g} \left(x^+_{if} W^{\mathrm{src},-}_{fc} + x^-_{if} W^{\mathrm{src},+}_{fc} \right) - \frac{b^{\mathrm{src}}_c}{8}. \end{aligned} \tag{18}$$

Using $x = x^+ - x^-$ and $W = W^+ - W^-$ gives $P_{igc} - N_{igc} = s_{igc}$ in Eq. 3. Both source-coordinate copies equal $(P_{igc} - N_{igc})/2$. Summing over the four heads therefore recovers the full linear source evidence:

$$\sum_{g=0}^{3} \left(H^{(0)}_{i,56+g,c} + H^{(0)}_{i,60+g,c} \right) = \sum_{f=0}^{F-1} x_{if} W^{\mathrm{src}}_{fc} + b^{\mathrm{src}}_c. \tag{19}$$

This identity states precisely how the eight source coordinates preserve the original feature contributions.

The initial claim score also has an explicit decomposition. For any fixed classes c, a,

$$\begin{aligned} q^{(0)}_{ic} - q^{(0)}_{ia} = \gamma_0 \Bigg[& \sum_{f=0}^{F-1} x_{if} \left(W^{\mathrm{self}}_{fc} - W^{\mathrm{self}}_{fa} \right) + b^{\mathrm{self}}_c - b^{\mathrm{self}}_a \\ & + \sum_{k=0}^{K-1} z_{ik} \left(W^{\mathrm{pred}}_{kc} - W^{\mathrm{pred}}_{ka} \right) \Bigg]. \end{aligned} \tag{20}$$

Together, Eqs. 19 and 20 connect the argument state and initial decision to explicit feature and predicate coefficients.

The role families in Eq. 5 use the following parameter indexing:

Role family	d_η	Learned parameter indices	Fixed offset ω_η
Atomic composition	3	r	$\log(1/4, 1/2, 1/4)$
Joint composition	3	r	$\log(1/4, 1/2, 1/4)$
Graph interaction	6	ℓ, g, c	$\log(5/8, 1/4, 1/8)$
Argument-to-claim	3	k, c	$(0, 0, 0)$

Each entry specifies one $\beta_\eta \in \mathbb{R}^3$ and $A_\eta \in \mathbb{R}^{3 \times d_\eta}$. Parameters are shared across all indices absent from the corresponding row. In particular, atomic and joint parameters are distinct; local and argument-to-claim parameters are reused at every layer; and graph-role parameters vary across layers.

W^{self} and U use Xavier-uniform initialization, and their biases are zero. The source mapping starts with the same values as the self mapping and is then optimized as a separate parameter tensor. W^{pred} starts at zero. Local and graph learned offsets and context coefficients start at zero, giving an identity local composition and a unit signed graph-role coefficient. For decision roles, a Xavier-uniform matrix $B \in \mathbb{R}^{K \times C}$ initializes $\beta_{\mathrm{dec},k,c} = (B_{kc}, 0, -B_{kc})$, while $A_{\mathrm{dec},k,c} = 0$. Layer coefficients start at

$$\gamma_\ell = 0.1(0.9)^\ell \quad (0 \le \ell < L), \qquad \gamma_L = (0.9)^L, \tag{21}$$

and are subsequently learned as real-valued parameters.

A.2 Algebraic properties of relational action

For $\rho = (p_S, p_N, p_A) \in \Delta^2$ and payload $u \in \mathbb{R}$, define $\mathfrak{a}(\rho, u) = 2(p_S - p_A)u$.

Lemma 1 (Role action)**.** *The role action satisfies*

$$|\mathfrak{a}(\rho, u)| \le 2(1 - p_N)|u| \le 2|u|. \tag{22}$$

For $\tau(\rho) = (p_A, p_N, p_S)$,

$$\mathfrak{a}(\tau(\rho), u) = -\mathfrak{a}(\rho, u), \qquad \mathfrak{a}(\rho, -u) = -\mathfrak{a}(\rho, u). \tag{23}$$

At the three simplex vertices,

$$\mathfrak{a}((1,0,0), u) = 2u, \quad \mathfrak{a}((0,1,0), u) = 0, \quad \mathfrak{a}((0,0,1), u) = -2u. \tag{24}$$

Proof. Nonnegativity and unit sum give $|p_S - p_A| \le p_S + p_A = 1 - p_N$. Multiplication by $2|u|$ yields Eq. 22. Swapping p_S and p_A, changing the sign of u, and substituting the simplex vertices give Eqs. 23 and 24. □

All local, graph, and decision payloads are bounded by one in absolute value, since they are either a $\tanh$ value or a product of two such values. Lemma 1 therefore bounds each unnormalized role-weighted payload by two. Applied to Eq. 11, it also gives the finite-depth score-change bound

$$|r_{ic}^{(\ell)}| \le 2\sqrt{K}, \qquad \|q_i^{(L)} - q_i^{(0)}\|_\infty \le 2\sqrt{K} \sum_{\ell=1}^{L} |\gamma_\ell|. \tag{25}$$

This bound expresses the scale of accumulated revisions in terms of the learned layer coefficients.

A.3 Proof of complete and equivariant decision traces

We consider the real-arithmetic equations of the model with finite inputs, fixed finite parameters, and a deterministic forward evaluation. Let σ be a permutation of the nodes. Relabel the feature rows and both endpoints of every edge so that $X'_{\sigma(i):} = X_{i:}$. Feature coordinates, argument coordinates, heads, and classes retain their indices. A traced node i corresponds to node $\sigma(i)$ in the relabeled computation.

Proposition 2 (Equivariant and complete decision traces)**.** *Relabeling feature rows and directed edge endpoints by σ relabels argument states, role assignments, claim scores, and decision contributions by the same permutation. For every node and class, Eq. 28 reconstructs the final score from the initial claim and recorded decision contributions. For every fixed pair of classes, Eq. 13 reconstructs their final margin.*

Proof of Proposition 2. Initialization. The raw feature projection, source feature groups, and initial claim map apply shared parameters to individual feature rows. Eqs. 3 and 4 thus give

$$H'^{(0)}_{\sigma(i),kc} = H^{(0)}_{ikc}, \qquad q'^{(0)}_{\sigma(i),c} = q^{(0)}_{ic}. \tag{26}$$

Local composition. Assume corresponding argument and claim states agree after layer $\ell - 1$. The bounded states, XOR partner indices, and products in Eq. 6 are identical at nodes i and $\sigma(i)$. Their atomic and joint role parameters are shared across nodes. Applying Eq. 7 therefore yields identical local roles and $\widetilde{H}'^{(\ell)}_{\sigma(i),kc} = \widetilde{H}^{(\ell)}_{ikc}$.

Graph interactions. Let $\bar{A}_{ji}$ denote the edge multiplicity after replacing diagonal entries by one. Under relabeling, $\bar{A}'_{\sigma(j)\sigma(i)} = \bar{A}_{ji}$. Thus incoming degrees obey $d'_{\sigma(i)} = d_i$, and corresponding normalized coefficients satisfy $\alpha'_{\sigma(j)\sigma(i)} = \alpha_{ji}$. Head summaries depend on fixed argument coordinates, so $\bar{b}'^{(\ell)}_{\sigma(i),g,c} = \bar{b}^{(\ell)}_{igc}$. The preceding claim states agree by induction, establishing equality of all six context components in Eq. 9. Consequently,

$$\rho'^{(\ell)}_{\sigma(j)\sigma(i),g,c} = \rho^{(\ell)}_{ji,g,c}, \qquad m'^{(\ell)}_{\sigma(j)\to\sigma(i),k,c} = m^{(\ell)}_{j\to i,k,c}. \tag{27}$$

Relabeling is a bijection on incoming neighbors. Summing the corresponding messages in Eq. 10 gives $H'^{(\ell)}_{\sigma(i),kc} = H^{(\ell)}_{ikc}$.

Claim revisions. The decision roles depend on the updated argument, the preceding class score, and parameters shared across nodes. Therefore, $\rho'^{(\ell)}_{\mathrm{dec},\sigma(i),k,c} = \rho^{(\ell)}_{\mathrm{dec},ikc}$. The shared coefficient γ_ℓ then gives $\xi'^{(\ell)}_{\sigma(i),kc} = \xi^{(\ell)}_{ikc}$ and $q'^{(\ell)}_{\sigma(i),c} = q^{(\ell)}_{ic}$. This completes the induction for the entire argument and decision trace.

Score and margin completeness. Eqs. 11 and 12 imply

$$q^{(\ell)}_{ic} - q^{(\ell-1)}_{ic} = \sum_{k=0}^{K-1} \xi^{(\ell)}_{ikc}.$$

Summing over ℓ telescopes to

$$q^{(L)}_{ic} = q^{(0)}_{ic} + \sum_{\ell=1}^{L} \sum_{k=0}^{K-1} \xi^{(\ell)}_{ikc}. \tag{28}$$

Subtracting this identity for fixed classes c and a proves Eq. 13. Reconstructed scores determine the same prediction under the same class-index tie rule. □

The identities specify the quantities used in numerical replay. Replay error is measured by comparing reconstructed states and scores with the recorded forward computation; edge reductions are evaluated in their recorded order. For an intervention, the selected relation is changed and the model recomputes its states with fixed parameters. Role neutralization preserves the graph and its degree weights, while edge deletion rebuilds those weights on the modified graph.

A.4 Propagation and certification of role interventions

This section gives the sensitivity matrices in Proposition 1. They bound a complete recomputation after neutralization, including changes in subsequent role assignments. Throughout, model parameters, input features, and normalized graph weights remain fixed; evaluation is deterministic and all quantities are finite. Absolute values and inequalities between vectors are componentwise.

Sensitivity of the role law. For each family and parameter index η, define the coefficient oscillations

$$\kappa_{\eta,t} = \frac{\max_{r\in\{S,N,A\}} A_{\eta,rt} - \min_{r\in\{S,N,A\}} A_{\eta,rt}}{\sqrt{d_\eta}}, \qquad 1 \le t \le d_\eta. \tag{29}$$

Lemma 2 (Role sensitivity)**.** *For all role contexts v, v',*

$$|\nu(\mathcal{R}_\eta(v)) - \nu(\mathcal{R}_\eta(v'))| \le \sum_{t=1}^{d_\eta} \kappa_{\eta,t} |v_t - v'_t|. \tag{30}$$

For a role family with $d_\eta = 3$ and $|u|, |u'|, |w|, |w'| \le 1$, the action $f_\eta(u, w) = u\nu(\mathcal{R}_\eta(\psi(u, w)))$ satisfies

$$|f_\eta(u, w) - f_\eta(u', w')| \le (2 + \kappa_{\eta,1} + \kappa_{\eta,3})|u - u'| + (\kappa_{\eta,2} + \kappa_{\eta,3})|w - w'|. \tag{31}$$

Proof. Write $p = \mathcal{R}_\eta(v)$ and let $a_S = 2$, $a_N = 0$, and $a_A = -2$, so that $\nu(p) = \sum_r p_r a_r$. Differentiating softmax gives

$$\frac{\partial \nu(\mathcal{R}_\eta(v))}{\partial v_t} = \frac{1}{\sqrt{d_\eta}} \operatorname{Cov}_{r\sim p}(a_r, A_{\eta,rt}). \tag{32}$$

A random variable in an interval of width w has variance at most $w^2/4$: its variance is the minimum mean squared distance to a constant, and its distance to the interval midpoint is at most $w/2$. Cauchy–Schwarz thus bounds the covariance of two bounded variables by one quarter of the product of their interval widths. The widths in Eq. 32 are 4 and $\max_r A_{\eta,rt} - \min_r A_{\eta,rt}$, giving the derivative bound $\kappa_{\eta,t}$. Integrating along the straight segment between v and v' proves Eq. 30. Finally, $|uw - u'w'| \le |u - u'| + |w - w'|$ and $|\nu| \le 2$ yield Eq. 31 by adding and subtracting $u'\nu(\mathcal{R}_\eta(\psi(u, w)))$. □

The constants depend on differences between role coefficients. Adding the same context coefficient to all three logits leaves both the role probabilities and these bounds unchanged.

The coordinate-derivative constant is sharp uniformly over the allowable role coefficients and finite learned offsets. To see this, set $v = 0$, choose $A_{\eta,St} = a > b = A_{\eta,At}$ and $A_{\eta,Nt} \in [b, a]$, and choose the total offset $\omega_\eta + \beta_\eta = (0, -M, 0)$. Every finite M is admissible. Then $p_S = p_A = 1/(2 + e^{-M})$, so Eq. 32 gives

$$\left.\frac{\partial \nu(\mathcal{R}_\eta(v))}{\partial v_t}\right|_{v=0} = \frac{2(a-b)}{(2+e^{-M})\sqrt{d_\eta}} \xrightarrow[M\to\infty]{} \kappa_{\eta,t}. \tag{33}$$

Hence no factor strictly below one can reduce the coordinate-derivative bound uniformly over this family. This establishes the sharp constant for the role law used to construct the propagation certificate.

Local composition. Fix a class c and compare two executions. Let $h_i = \max_k |H_{ikc} - H'_{ikc}|$ and $e_i = |q_{ic} - q'_{ic}|$. Define $\widetilde{h}_i = \max_k |\widetilde{H}_{ikc} - \widetilde{H}'_{ikc}|$ for the locally composed states. Since $\tanh$ is 1-Lipschitz, atomic payload differences are at most h_i, joint payload differences at most $2h_i$, and receiving-argument differences at most h_i. Define

$$\begin{aligned} a_{\mathrm{loc}} = 1 + \frac{1}{\sqrt{6}} \sum_{r=0}^{5} \Big[& 2 + \kappa_{\mathrm{at},r,1} + \kappa_{\mathrm{at},r,2} + 2\kappa_{\mathrm{at},r,3} \\ & + 4 + 2\kappa_{\mathrm{jt},r,1} + \kappa_{\mathrm{jt},r,2} + 3\kappa_{\mathrm{jt},r,3} \Big]. \end{aligned} \tag{34}$$

Here $\kappa_{\mathrm{at},r,t}$ abbreviates Eq. 29 for partner r, and similarly for joint roles. Lemma 2, the residual term, and the triangle inequality give $\widetilde{h} \preceq a_{\mathrm{loc}} h$.

Graph exchange. Let $\mathsf{T}_{ij} = \alpha_{ji}$ and $\mathsf{D} = \mathrm{diag}(\mathsf{T}\mathbf{1})$, where $\mathbf{1} \in \mathbb{R}^N$ is the all-ones vector. For layer ℓ and class c, define

$$\begin{aligned} a^{\mathrm{src}}_{\ell,c} &= \max_g (2 + \kappa_{\mathrm{gr},\ell,g,c,1} + \kappa_{\mathrm{gr},\ell,g,c,3}), \\ a^{\mathrm{tgt}}_{\ell,c} &= \max_g (\kappa_{\mathrm{gr},\ell,g,c,2} + \kappa_{\mathrm{gr},\ell,g,c,3}), \\ b^{\mathrm{src}}_{\ell,c} &= \max_g (\kappa_{\mathrm{gr},\ell,g,c,4} + \kappa_{\mathrm{gr},\ell,g,c,6}), \\ b^{\mathrm{tgt}}_{\ell,c} &= \max_g (\kappa_{\mathrm{gr},\ell,g,c,5} + \kappa_{\mathrm{gr},\ell,g,c,6}). \end{aligned} \tag{35}$$

A head average changes by at most the largest argument change at that node. Applying Eq. 30 to the six graph-context entries and separating payload and role changes bounds the message difference by

$$|m_{j\to i,k,c} - m'_{j\to i,k,c}| \le \alpha_{ji} \left(a^{\mathrm{src}}_{\ell,c} \widetilde{h}_j + a^{\mathrm{tgt}}_{\ell,c} \widetilde{h}_i + b^{\mathrm{src}}_{\ell,c} e_j + b^{\mathrm{tgt}}_{\ell,c} e_i \right). \tag{36}$$

Define the nonnegative matrices

$$\begin{aligned} \mathsf{B}_{\ell,c} &= a_{\mathrm{loc}} \left(I_N + a^{\mathrm{tgt}}_{\ell,c} \mathsf{D} + a^{\mathrm{src}}_{\ell,c} \mathsf{T} \right), \\ \mathsf{C}_{\ell,c} &= b^{\mathrm{tgt}}_{\ell,c} \mathsf{D} + b^{\mathrm{src}}_{\ell,c} \mathsf{T}. \end{aligned} \tag{37}$$

Summing Eq. 36 and adding the residual yields $h^{(\ell)} \preceq \mathsf{B}_{\ell,c} h^{(\ell-1)} + \mathsf{C}_{\ell,c} e^{(\ell-1)}$. The diagonal D retains the actual row sums of the degree-normalized graph.

Claim revision. For the decision-role coefficients, set

$$\begin{aligned} u_c &= \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} (2 + \kappa_{\mathrm{dec},k,c,1} + \kappa_{\mathrm{dec},k,c,3}), \\ v_c &= \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} (\kappa_{\mathrm{dec},k,c,2} + \kappa_{\mathrm{dec},k,c,3}), \\ u_{g,c} &= \frac{1}{\sqrt{K}} \sum_{k:g(k)=g} (2 + \kappa_{\mathrm{dec},k,c,1} + \kappa_{\mathrm{dec},k,c,3}). \end{aligned} \tag{38}$$

Lemma 2 applied coordinate by coordinate gives

$$e^{(\ell)} \preceq |\gamma_\ell| u_c h^{(\ell)} + (1 + |\gamma_\ell| v_c) e^{(\ell-1)}. \tag{39}$$

Consequently, with $z^{(\ell)} = (h^{(\ell)\top}, e^{(\ell)\top})^\top \in \mathbb{R}^{2N}$,

$$z^{(\ell)} \preceq \mathsf{M}_{\ell,c} z^{(\ell-1)}, \qquad \mathsf{M}_{\ell,c} = \begin{bmatrix} \mathsf{B}_{\ell,c} & \mathsf{C}_{\ell,c} \\ |\gamma_\ell| u_c \mathsf{B}_{\ell,c} & (1 + |\gamma_\ell| v_c) I_N + |\gamma_\ell| u_c \mathsf{C}_{\ell,c} \end{bmatrix}. \tag{40}$$

Proof of Proposition 1. The two executions agree through local composition at layer s. Replacing one graph role by $(0, 1, 0)$ removes exactly $\alpha_{ji} \nu(\rho^{(s)}_{ji,g,c}) \tanh \widetilde{H}^{(s)}_{jkc}$ from $H^{(s)}_{ikc}$ for each k in head g, leaving every other node, coordinate, and class unchanged at that stage. Equation 14 is therefore the exact maximum absolute argument difference at node i. The preceding claims agree, so Eq. 31, summed only over the changed head, gives

$$z^{(s)} \preceq \begin{bmatrix} \delta \mathbf{e}_i \\ |\gamma_s| u_{g,c} \delta \mathbf{e}_i \end{bmatrix}. \tag{41}$$

For each later layer the executions apply identical update functions to their respective current states. Applying Eq. 40 recursively is valid because every $\mathsf{M}_{\ell,c}$ is nonnegative and thus preserves componentwise inequalities. Selecting the final claim block proves Eq. 15.

Every role and payload in the forward equations depends only on the class coordinate being updated. Fixed-parameter recomputation therefore leaves all classes other than c unchanged. At layer s, the perturbation is confined to node i. Local composition and claim revision act within a node; graph exchange can pass a perturbation only from an affected source to an adjacent target, or retain it at an already affected target. Induction shows that after layer $\ell \geq s$ the affected nodes lie in the directed $(\ell - s)$-hop neighborhood of i. This proves spatial locality. Subsequent local compositions may distribute the perturbation across argument heads, and the bound tracks all K coordinates through h.

At any node v, if the unique original winner is c, its score can decrease by at most ε_v, while every competing score remains fixed. If the unique winner is $a \neq c$, its score and all competitors other than c remain fixed, while class c can increase by at most ε_v. The respective strict margin conditions in the proposition therefore preserve the winner. □

The injection also obeys $\delta \leq 2\alpha_{ji}(1 - \rho^{(s)}_{N,ji,g,c}) \max_{k:g(k)=g} |\tanh \widetilde{H}^{(s)}_{jkc}|$. Thus the role mixture and transmitted content determine the initial disturbance, while the learned downstream coefficients and graph paths determine its propagation. The certificate can be evaluated by successive sparse matrix–vector products using T, without forming dense $2N \times 2N$ matrices.

A sharper certificate for the last layer. At $s = L$, the preceding claim at the affected node is fixed. Set $d = \tanh q^{(L-1)}_{ic}$, and let $A_{\mathrm{dec},k,c,:,t}$ denote the role-coefficient column for context entry t. With $\mathrm{osc}(v) = \max_r v_r - \min_r v_r$, define

$$\zeta_{kc}(d) = 2 + \frac{\mathrm{osc}\,(A_{\mathrm{dec},k,c,:,1} + d A_{\mathrm{dec},k,c,:,3})}{\sqrt{3}}. \tag{42}$$

Then the sole potentially changed final claim score satisfies

$$\left| q^{(L),\mathrm{neut}}_{ic} - q^{(L)}_{ic} \right| \leq \frac{|\gamma_L| |\alpha_{ji}| |\nu(\rho^{(L)}_{ji,g,c})|}{\sqrt{K}} \sum_{k:g(k)=g} \zeta_{kc}(d) \left| \tanh \widetilde{H}^{(L)}_{jkc} \right|. \tag{43}$$

To prove this refinement, hold d fixed and differentiate the decision action with respect to $u = \tanh H^{(L)}_{ikc}$. Equation 32 bounds the derivative of its net role coefficient by $\mathrm{osc}(A_{:,1} + d A_{:,3})/\sqrt{3}$. The derivative of the payload-weighted action is bounded by $\zeta_{kc}(d)$, using $|u| \leq 1$ and $|\nu| \leq 2$. Since $\tanh$ is 1-Lipschitz, applying this bound to each coordinate's exact removed message and summing yields Eq. 43. This refinement uses the recorded preceding claim and each transmitted payload individually.

A.5 Prior estimation, optimization, and validation

Let $T = \mathcal{V}_{\mathrm{tr}}$ and $U_{\mathrm{nodes}} = \mathcal{V} \setminus T$. The training class prior is

$$\pi_{T,c} = \frac{|\{i \in T : y_i = c\}|}{|T|}, \qquad \pi_{T,c} > 0.$$

The first 100 optimizer updates minimize ordinary cross-entropy on T. At the end of update 100, evaluation without feature dropout produces fixed probabilities

$$p_{uc} = \mathrm{softmax}\left(q^{(L)}_{\theta_{100},u}(X, \mathcal{G})\right)_c, \qquad u \in U_{\mathrm{nodes}}.$$

The estimator receives detached logits, training-node membership, and π_T; it evaluates the corresponding log-probabilities in CPU float64 arithmetic. Following the prior-shift EM procedure of Saerens et al. (2002), set $\pi^{(0)} = \pi_T$ and perform

$$R^{(s)}_{uc} = \frac{p_{uc}\pi^{(s)}_c/\pi_{T,c}}{\sum_{a=0}^{C-1} p_{ua}\pi^{(s)}_a/\pi_{T,a}}, \qquad \pi^{(s+1)}_c = \frac{1}{|U_{\mathrm{nodes}}|}\sum_{u \in U_{\mathrm{nodes}}} R^{(s)}_{uc}, \quad s = 0, \ldots, 49. \tag{44}$$

The original estimator interprets these updates through a change in class priors with fixed class-conditional distributions. Here its output $\widehat{\pi}_U = \pi^{(50)}$ supplies fixed training weights:

$$w_c = \frac{\widehat{\pi}_{U,c}/\pi_{T,c}}{\sum_{a=0}^{C-1} \pi_{T,a}(\widehat{\pi}_{U,a}/\pi_{T,a})}, \qquad \widetilde{w}_c = 1 + \lambda(w_c - 1), \quad \lambda \in \{0, \tfrac{1}{2}, 1\}. \tag{45}$$

Consequently, $\sum_c \pi_{T,c}\widetilde{w}_c = 1$. The loss from update 101 onward is the mean of $\widetilde{w}_{y_i}[-\log P_{\theta,iy_i}]$ over training nodes. The case $\lambda = 0$ executes the same estimation step and uses unit effective weights.

EM runs in the log domain with CPU float64 arithmetic. Before taking a prior's logarithm, the value is bounded below by the smallest positive normal float64 value. Responsibilities are computed by softmax, and the final weights are normalized as in Eq. 45. The estimate remains fixed during subsequent gradient updates.

Training uses feature dropout on the input, Adam with coupled weight decay, and global gradient-norm clipping at five. The same optimizer and its moment estimates continue across update 100. After each update, evaluation on the validation nodes uses the full feature matrix. The checkpoint changes only when the registered validation metric strictly increases; an exact tie keeps the earlier checkpoint. Early stopping independently tracks strict improvements in unweighted validation cross-entropy, with patience 200 and a maximum of 2000 updates. The registered metric is accuracy for the multiclass benchmarks and ROC-AUC of $q^{(L)}_{i1} - q^{(L)}_{i0}$ for Questions. Final inference uses the selected checkpoint's original accumulated scores.

A.6 Computation and parameterization

The local computation evaluates R atomic and R joint relations for each node, argument coordinate, and class, costing $O(NKCR)$ per layer. Graph contexts are computed once per node and head; evaluating edge-role mixtures costs $O(|\overline{\mathcal{E}}|GC)$, and transmitting their coordinate-wise payloads costs $O(|\overline{\mathcal{E}}|KC)$. Argument-to-claim updates cost $O(NKC)$. Including feature grounding, the forward complexity is

$$O\big(NF(K + C) + L\left[NKCR + |\overline{\mathcal{E}}|KC\right]\big). \tag{46}$$

Role-context sharing reduces repeated edge-level computations while retaining the full argument state. Edge messages can be evaluated in chunks and accumulated at their targets, with the same equations and parameters.

The number of learned scalar parameters is

$$FK + 2FC + K + 2C + 13KC + 21LGC + 24R + (L + 1). \tag{47}$$

The terms account for the predicate projection, two feature-to-class mappings, their biases, predicate-to-class and decision-role parameters, graph roles, two local role families, and layer coefficients. For $K = 64$, $G = 4$, and $R = 6$, this becomes $64F + 2FC + (834 + 84L)C + L + 209$.

B Experimental Details

B.1 Datasets, graph construction, and repetition structure

The datasets connect argumentation to several established graph-learning settings. Amazon Photo and Coauthor Physics belong to the benchmark collections studied by Shchur et al. (2018); WikiCS combines hyperlink structure with text-derived node features (Mernyei & Cangea, 2020); and Questions supplies official masks and a binary ROC-AUC task from the heterophily benchmark of Platonov et al. (2023). Table 3 describes the actual inputs and evaluation identities used here.

Table 3: **Dataset characteristics and evaluation identities.** T/V/E counts are shown for the first registered split or mask. "5×2" means five splits or official masks with two initializations each. Accuracy is used except on Questions, where the metric is ROC-AUC.

Dataset	Nodes	Features	Classes	T/V/E	Runs
Cora	2,708	1,433	7	1,557/542/609	5×2
Citeseer	3,327	3,703	6	1,929/665/733	5×2
Pubmed	19,717	500	3	11,829/3,943/3,945	5×2
Photo	7,650	745	8	4,144/1,530/1,976	5×2
WikiCS	11,701	300	10	6,079/2,340/3,282	5×2
Physics	34,493	8,415	5	18,689/6,899/8,905	5×2
Actor	7,600	932	5	4,501/1,520/1,579	5×2
Texas	183	1,703	5	85/37/61	5×2
Questions	48,921	301	2	24,460/12,230/12,231	5×2

Input graphs are symmetrized by taking the union of each edge and its reverse and removing duplicates. The model replaces existing self-loops with exactly one self-loop per node and uses the normalization in Eq. 10. Cora, Citeseer, Pubmed, Photo, Actor, and Texas use row-sum feature normalization; Physics uses the registered NormalizeFeatures-equivalent transform. WikiCS and Questions retain their original feature values. Questions retains the node ordering bound to its official masks.

For the eight class-quota datasets, the realized training fraction follows class availability. Texas, for example, has 85 training, 37 validation, and 61 evaluation nodes; Cora has 1,557, 542, and 609. Questions uses official masks 0–4.

B.2 Training configuration

Table 4: **Validation-selected configurations.** One configuration is used across all ten runs of each dataset. L is the number of reasoning layers, η the Adam learning rate, w_d its weight decay, and p feature dropout. Numerical entries are rounded for display; the experiment configurations retain full training precision. All models use dense predicate projection, 64 coordinates, four heads, and six XOR partners.

Dataset	L	λ	η	w_d	p
Cora	10	1	7.63380×10^{-4}	4.29659×10^{-3}	0.146110
Citeseer	10	1	6.81919×10^{-4}	5.39008×10^{-5}	0.794365
Pubmed	10	1	6.90491×10^{-3}	1.13087×10^{-4}	0.423811
Photo	10	1	10^{-2}	5×10^{-4}	0.5
WikiCS	10	1	10^{-2}	5×10^{-4}	0.5
Physics	10	1	10^{-2}	5×10^{-4}	0.5
Actor	10	1	1.55731×10^{-2}	4.54298×10^{-4}	0.399290
Texas	4	1	1.2×10^{-2}	10^{-3}	0.5
Questions	10	1	10^{-2}	5×10^{-4}	0.5

Table 4 specifies one validation-selected configuration per dataset across all ten runs. The candidate ranges originate from the preceding development stage, which used validation and evaluation feedback.

Each training run initializes its own model and optimizer. The final class scores supervise every update. The class-prior estimator combines detached predictions on non-training nodes with the training-label prior. After 100 updates, 50 EM iterations produce fixed class weights, and Adam continues with its existing state. The highest validation metric determines the retained checkpoint; early stopping tracks strict improvements in unweighted validation cross-entropy, as specified in Appendix A.5.

B.3 Comparison methods

The comparison covers three complementary approaches to graph information flow. Neighborhood and hop-based models include GCN, GAT, H2GCN, and HopGNN (Kipf & Welling, 2017; Veličković et al., 2018; Zhu et al., 2020; Chen et al., 2023a). Spectral and higher-order propagation includes ChebyNet, GPRGNN, BernNet, JacobiConv, and HiGCN (Defferrard et al., 2016; Chien et al., 2021; He et al., 2021; Wang & Zhang, 2022; Huang et al., 2024). GPRGNN+JDR additionally aligns graph and feature spectral spaces through joint rewiring and feature denoising (Linkerhägner et al., 2025). The attention-based group includes Transformer, GraphGPS, NodeFormer, SGFormer, NAGphormer, and Specformer (Vaswani et al., 2017; Rampášek et al., 2022; Wu et al., 2022; 2023; Chen et al., 2023b; Bo et al., 2023). These families provide complementary tests of QBAF's central design: graph information acts through explicit argument roles and contributes to accumulated class claims.

For Questions, Specformer uses a truncated spectrum.

B.4 Ablation panel and matched identities

The 48 ablation runs comprise Cora, Photo, Texas, and Actor, three factors, two splits, and two initializations. Every ablation uses its dataset's selected configuration and is paired with the same main-experiment split and initialization. For each factor, Table 2 reports the mean of the four paired accuracy differences.

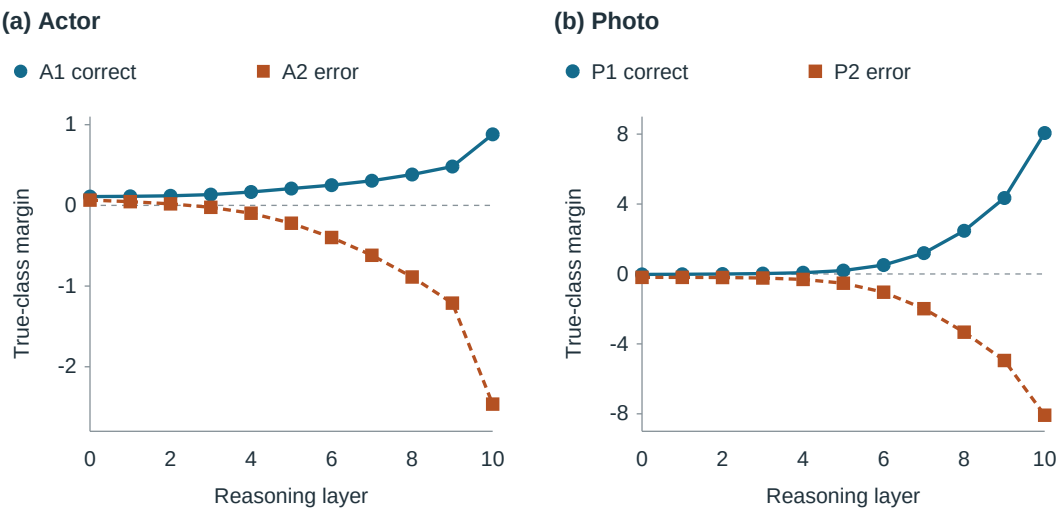


Figure 2: **Recorded claim trajectories explain how decisions develop.** Each marker is the actual margin $q_{iy}^{(\ell)} - q_{ia}^{(\ell)}$ at layer ℓ, with the true class y and the same fixed rival a throughout a case. Solid circles indicate correctly classified cases; dashed squares indicate errors. Actor and Photo use their own vertical scales. P1 crosses its fixed rival between layers 2 and 3, whereas A2 crosses in the opposite direction at the same transition. The traces expose both successful revisions and the accumulation of evidence favoring an incorrect claim.

B.5 Decision reconstruction and relational interventions

The four decision cases are Actor nodes 2 and 4 and Photo nodes 13 and 153. They retain their original feature indices, learned coefficients, class labels, and fixed rivals. Role probabilities act on signed hidden-state payloads; final-score contributions are the recorded claim increments, and edge intervention effects are obtained by downstream recomputation. For intervention, the selected incoming edge has the largest absolute final-layer signed hidden-message difference between the true class and its fixed rival. This selection score is distinct from the final class-score contribution in Eq. 12. Edge neutralization applies to all its layers, heads, and classes.

Figure 2 resolves the endpoint comparison into the actual reasoning steps. For P1, the initial margin is -0.0250, becomes positive at layer 3, and reaches 8.0630. For A2, the initial margin is $+0.0655$, becomes negative at layer 3, and reaches -2.4630. These trajectories connect a prediction to the sequence of claim revisions that produced it. Signed payloads, role coefficients, and learned layer coefficients jointly determine each increment, as specified by Eq. 12.

Table 5 gives the endpoints and reconstruction errors of all four cases. Replay reconstructs graph-state updates and final class scores with zero observed absolute error, yielding the same fixed-rival margins. Reconstructing the initial margins from feature, predicate, and bias terms has maximum absolute error 1.49×10^{-8}. The computational explanation connects raw evidence, intermediate revisions, and final decisions.

For the three cases with eligible non-self incoming edges, controls use 16 fixed-seed draws at the same intervention budget and direction. Source-degree matching uses the nearest eligible degree bucket: its distance is two for A1, zero for A2, and four for P1. The eligible pools contain three, one, and one distinct edges, respectively. The same draws are used for all three intervention types. Photo node 153 has no non-self incoming edge, so its graph-edge intervention is inapplicable. Table 6 reports every eligible case and its actual control coverage.

The distinction between local evidence and total intervention effect is particularly visible in P1. Neutralizing its selected role reduces the true-class margin by 2.2822, while deleting the edge reduces it by 0.6809. Retaining that edge while removing the target's other non-self incoming edges increases the margin by 0.1582. These are three different modifications to the same reasoning pro-

Table 5: **From feature-grounded evidence to reconstructed decisions.** Cases are the four validation nodes in Figure 1. The tuple $(y, \hat{y}, a)$ gives the true class, predicted class, and fixed rival. Margins compare y against a. Replay error is the maximum absolute reconstruction error over graph-state updates and final class scores. Initial-margin reconstruction uses feature, predicate, and bias terms.

Case	Node	$(y, \hat{y}, a)$	Initial margin	Final margin	Replay error	Initial error
A1	Actor 2	$(3, 3, 4)$	$+0.108942$	$+0.880378$	0	1.86×10^{-9}
A2	Actor 4	$(1, 4, 4)$	$+0.065518$	-2.462960	0	3.73×10^{-9}
P1	Photo 13	$(1, 1, 3)$	-0.025045	$+8.062979$	0	1.86×10^{-9}
P2	Photo 153	$(1, 6, 6)$	-0.194120	-8.080023	0	1.49×10^{-8}

cess: neutralization changes role action at fixed normalization, while deletion and retention also change graph normalization. The model exposes and executes each operation separately.

Table 6: **Relational action and graph structure produce distinct decision effects.** Entries are changes in the true-class margin after fixed-parameter recomputation. Selected edges are $508 \to 2$ (A1), $5260 \to 4$ (A2), and $1160 \to 13$ (P1). Each random value averages 16 pre-fixed equal-budget draws from the eligible control pool. Neutralization changes the selected edge's roles; deletion removes it; retention keeps it while removing the target's other non-self incoming edges. Deletion and retention recompute graph normalization. Positive values increase the true-class margin.

	Neutralize		Delete		Retain	
Case	Selected	Random	Selected	Random	Selected	Random
A1	-0.11677	$+0.01652$	-0.09665	$+0.04286$	$+1.49033$	$+0.92295$
A2	$+0.28081$	$+0.19961$	$+0.16071$	$+0.06801$	-0.47774	-0.00379
P1	-2.28221	-1.27468	-0.68091	$+0.15821$	$+0.15821$	-0.68091

Case	Non-self incoming edges	Control-pool edges	Fixed draws	Source-degree bucket distance
A1	12	3	16	2
A2	6	1	16	0
P1	2	1	16	4
P2	0	0	0	—

B.6 TIMING PROTOCOL

GPU inference measurements use two warm-up passes and five synchronized measurements with the model and graph resident on the device. They exclude loading, graph construction, and host-to-device transfer. Peak allocated memory is 431.0 MB for Photo and 286.5 MB for Actor. CPU replay uses one warm-up and five measurements with two threads, includes trace deserialization and hash verification, and invokes no model forward pass. Intervention timings include loading, graph construction, transfer, and output copying; file writing is measured separately. The timings in Section 5.4 preserve these operation-specific scopes.

Table 7 expands the measurements to ordinary execution, explanation capture, replay, and all three interventions. CPU forward and capture each use one warm-up and three timed executions. Tensor and trace-JSON serialization are each measured once per dataset. Each intervention type uses one warm-up followed by one measurement per distinct request, with two Photo requests and six Actor requests. The corresponding JSON-write medians use six and eighteen measurements across the three operation types. These measurements identify the cost of obtaining an explanation during inference and the cost of reusing it afterward.

Table 7: **Measured costs of prediction, explanation, and intervention.** Times are seconds. Repeated operations report their measured median. CPU operations use two threads; GPU operations use an RTX 5090D. Forward and capture timings use resident models and graphs. Replay includes trace loading and verification. Intervention APIs include loading, graph construction, transfer, synchronized forward computation, and output copying. Serialization is measured separately.

Device	Operation	Photo	Actor
CPU	Ordinary forward	39.984236	21.522404
	Forward with explanation capture	39.914527	22.388377
	Saved-trace replay	0.044266	0.026092
	Trace tensor serialization and hash	0.119227	0.088797
	Trace JSON write and flush	0.001374	0.000902
GPU	Ordinary forward	0.079455	0.032038
	Neutralization API	0.127549	0.059478
	Deletion API	0.098190	0.051286
	Retention API	0.098602	0.053218
Host	Intervention JSON write and flush	0.000945	0.000890

C Extended Related Work

From argument acceptability to learned reasoning. Computational argumentation studies how interacting reasons determine the acceptability of a claim. Abstract argumentation formalizes attacks between arguments (Dung, 1995); bipolar frameworks distinguish support from attack (Cayrol & Lagasquie-Schiex, 2005); and gradual semantics analyze numerical strengths and the principles governing their response to these relations (Baroni et al., 2018). Together, these frameworks make arguments and their supporting or attacking relations explicit objects of reasoning. We bring this perspective to node classification, where a node's attributes and graph context provide evidence concerning several competing class claims.

Neuro-argumentative learning connects these foundations to prediction through different choices of arguments and interaction structures. Potyka (2021) connects multilayer perceptrons with quantitative argumentation and gradient-based parameter learning. Gradual AA-CBR and Deep Arguing jointly learn neural representations and case-based argumentative interactions (Gould & Toni, 2025; 2026), while CAM organizes tabular prediction through concepts and learnable quantitative argumentation layers (Chi et al., 2026). A complementary line trains neural architectures, including GCNs and GATs, to approximate strengths produced by QuAD semantics (Al Anaissy et al., 2025). We develop a predictor whose arguments are feature-derived, class-indexed vectors at the nodes of an observed graph. Their interactions depend on the current source and target states at each layer, connecting local argument composition, graph propagation, and cumulative class decisions within one explicit reasoning process.

Adaptive graph propagation. Graph learning adapts both the sources and the form of propagated information. Neighborhood aggregation ranges from degree-normalized graph convolution (Kipf & Welling, 2017) to endpoint-dependent attention (Veličković et al., 2018; Brody et al., 2022). Heterophily motivates greater control over how neighborhood information enters a representation: H2GCN separates ego and neighbor representations and combines information across neighborhoods and layers (Zhu et al., 2020), while FAGCN and GPR-GNN learn signed edge gates and hop coefficients, respectively (Bo et al., 2021; Chien et al., 2021). Joint denoising and rewiring aligns the leading spectral spaces of graph and feature matrices before downstream prediction (Linkerhägner et al., 2025). Across these approaches, the central design choice is how information contributes to a node's representation. We make that contribution an explicit argument interaction, assigning support, neutral, and attack coefficients to each edge, layer, head, and class. These coefficients normalize across roles, and each message preserves its argument coordinate. This exposes how a neighbor's evidence acts on a particular class claim throughout propagation.

Explaining graph predictions and argumentative contributions. Graph explanations address both the evidence associated with a prediction and the computation producing it. GNNExplainer optimizes graph and feature masks for individual predictions (Ying et al., 2019); PGExplainer learns a shared generator of structural explanations (Luo et al., 2020); and SubgraphX searches explanatory subgraphs using Monte Carlo tree search and approximate Shapley scoring (Yuan et al., 2021). Intrinsic approaches such as GSAT jointly learn stochastic explanatory subgraphs and the predictor through an information-bottleneck objective (Miao et al., 2022). Argumentative explanations provide a complementary representation: SpArX compresses a trained multilayer perceptron and translates the resulting sparse computation into quantitative argumentation (Ayoobi et al., 2023). Kampik et al. (2024) compare contribution functions through principles governing an argument's influence on a topic argument. We make argumentative operations the node predictor itself. Feature-derived arguments, role-conditioned messages, and class-score revisions form one computational trace: replay reconstructs its output, and fixed-parameter interventions recompute how that output changes with its relations. Prediction and explanation thus share the same learned evidence and reasoning operations.