Guiding Admissibility Solvers via Co-Admissibility Predictions

Sandra HOFFMANN, Isabelle KUHLMANN, and Matthias THIMM
Artificial Intelligence Group, University of Hagen, Hagen, Germany

Abstract. We investigate the use of graph-based features and machine learning models for predicting *co-admissibility*, i. e., whether multiple arguments occur *jointly* in an admissible set, in abstract argumentation frameworks, with the goal of guiding a complete solver. We introduce the notion of *hard arguments*, whose acceptance cannot be determined deterministically, as a principled learning target, and propose the *i-categoriser*, a query-sensitive gradual semantics feature refined using deterministic admissibility information. Through a systematic evaluation of 21 features across four models on three datasets of increasing difficulty, we show that feature selection has a greater impact on prediction quality than model architecture, and propose a compact four-feature set for solver guidance. Integrating this feature set into the SMART solver achieves near-perfect guidance on two datasets and improves coverage and reduces solving time on ICCMA 2025 benchmarks compared to the degree-feature baseline, with an online refinement variant yielding further gains on individual instances.

Keywords. abstract argumentation, machine learning, heuristic solver

1. Introduction

Argumentation is fundamental to reasoning under uncertainty, as new information can overturn previously justified conclusions. In *abstract* argumentation, introduced by Dung [1], reasoning is modelled using a set of arguments and a binary attack relation. This paradigm has been widely applied in areas such as legal reasoning, decision making, and explainable AI [2–4]. Recent work has explored *Graph Neural Networks* (GNNs) for predicting argument acceptance under admissibility [5–7]. While achieving near-perfect accuracy on synthetic data, these models struggle to generalise to harder benchmark instances such as those used in the ICCMA[1] competition. Moreover, their performance is often driven by simple structural signals such as node degrees, and classical models such as random forests can already achieve strong performance using such features [8].

Learning-based predictions have also been used to guide symbolic solvers. Hoffmann et al. [9] show that GNN-based admissibility predictions can reduce backtracking, but require post-processing before running the solver since admissibility is defined over sets of arguments and individual acceptability predictions do not directly translate to joint admissibility of arguments. To address this, Malmqvist [10] proposes predicting

[1] https://argumentationcompetition.org/

co-admissibility, i. e. whether pairs of arguments can jointly belong to an admissible set, enabling more effective integration into SAT-based solvers.

Craandijk and Bex [11] found attention-based aggregation to outperform fixed-weight alternatives for out-of-domain generalisation; building on this, Cibier and Mailly [12] propose a graph-attention architecture competitive with the strongest ICCMA 2023 systems.

In this work, we systematically study the impact of feature representations and neural architectures for learning on abstract argumentation frameworks. We evaluate 21 graph-based features using a Random Forest classifier and multiple GNN variants, both individually and in pairwise combinations[2], on datasets shown to challenge learning-based methods [8]. From this analysis, we derive a compact feature set with strong predictive performance. We further introduce *hard arguments*, defined as arguments whose acceptance cannot be determined without branching in a complete solver, and therefore constitute the most relevant targets for learning-based guidance. Using the best-performing features and models, we generate co-admissibility predictions for ICCMA 2025 instances and integrate them into the SMART solver [13], evaluating their impact against both the original solver and a gold-standard oracle.

The main contributions of this work are as follows:

- Systematic evaluation of 21 graph-based features across Random Forest and GNN models, including pairwise combinations, leading to a compact high-performing feature set.
- Introduction of *hard arguments* as a principled learning target for solver guidance.
- Refinement of a query-based feature using deterministic admissibility information, yielding the strongest predictive performance.
- Integration of the resulting feature set into the SMART solver and evaluation on ICCMA 2025 benchmarks.

The remainder of the paper is structured as follows. Section 2 introduces background, Section 3 defines hard arguments, and Sections 4–7 present experiments and results. Section 8 concludes.

2. Preliminaries

This section introduces the necessary background on abstract argumentation frameworks, semantics, co-admissibility, the SMART solver, and the machine learning models considered in this work.

2.1. Abstract Argumentation

An *abstract argumentation framework* (AF) [1] is a tuple $F = (\mathsf{Args}, R)$, where Args is a finite set of arguments and $R \subseteq \mathsf{Args} \times \mathsf{Args}$ is a binary attack relation. A set $E \subseteq \mathsf{Args}$ *defends* $a \in \mathsf{Args}$ if every attacker of a is attacked by some element of E.

Definition 1 (Extensions)**.** Let $F = (\mathsf{Args}, R)$ be an argumentation framework. A set $E \subseteq \mathsf{Args}$ is *conflict-free* if there are no $a, b \in E$ with $(a, b) \in R$; *admissible* if conflict-

[2]Code available under: `https://e.feu.de/co-adm`

free and every $a \in E$ is defended by E; *complete* if admissible and containing every argument defended by E; *preferred* if $\subseteq$-maximal complete; and *grounded* if $\subseteq$-minimal complete. We denote the set of all extensions under semantics σ by $\sigma(F)$, with $\mathsf{gr}(F)$, $\mathsf{co}(F)$, $\mathsf{pr}(F)$ referring to the grounded, complete, and preferred extensions respectively. We write $a^- = \{b \in \mathsf{Args} \mid (b,a) \in R\}$ and $a^+ = \{b \in \mathsf{Args} \mid (a,b) \in R\}$, as well as $A^- = \{b \in \mathsf{Args} \mid \exists a \in A \; s.t. \; b \in a^-\}$ and $A^+ = \{b \in \mathsf{Args} \mid \exists a \in A \; s.t. \; b \in a^+\}$.

A key decision problem in abstract argumentation is *credulous acceptability* (DC-σ): given an AF F, a semantics $\sigma \in \{\mathsf{co}, \mathsf{pr}, \mathsf{gr}\}$, and $q \in \mathsf{Args}$, decide whether $q \in E$ for some $E \in \sigma(F)$. Semantics can equivalently be defined via *labellings* [14], total functions $L : \mathsf{Args} \to \{in, out, undec\}$, where a labelling is *complete* iff every argument labelled *in* has all attackers labelled *out*, every argument labelled *out* has at least one attacker labelled *in*, and every argument labelled *undec* has neither all attackers labelled *out* nor an attacker labelled *in*.

Definition 2 (Co-admissibility [10])**.** Two arguments $a, b \in \mathsf{Args}$ are *co-admissible* if there exists an admissible set $S \subseteq \mathsf{Args}$ with $\{a,b\} \subseteq S$.

2.2. The SMART Solver

The SMART solver [13] is a backtracking-based algorithm for DC-pr that follows the labelling-based approach of Nofal et al. [15]. Alongside the standard labels, it uses *blank* for arguments that have not yet received a label, and *must out* and *must undec* to mark arguments that must eventually be set to *out* or *undec*, respectively. These auxiliary labels implement a look-ahead pruning strategy [16], which allows the algorithm to detect at an early stage that a partial labelling cannot be extended to a complete extension, thereby pruning the search space without fully exploring dead-end branches. Given a query q, the solver computes $\mathsf{gr}(F)$, constructs an initial labelling, and propagates it deterministically. When branching is required, it selects an argument predicted to be co-admissible with q, preferring those attacking a *must out* argument, and falling back to the blank argument with the highest total degree [15]. A branch is abandoned early if a *must out* argument has no blank attackers (*hopeless* labelling), and a *terminal* labelling is checked for admissibility to confirm acceptance of q.

2.3. Machine Learning Models

We consider four models, following prior work on learning-based approaches to abstract argumentation: a *Random Forest* (RF) [17], a *Graph Convolutional Network* (GCN) [18], a *Graph Attention Network* (GAT) [19, 20], and an argumentation-specific GAT variant (AFGAT) [12]. The RF operates on a fixed feature vector per argument without exploiting graph structure, serving as a baseline for assessing the benefit of graph-based learning. GCNs aggregate neighbour information via fixed normalised weights determined solely by graph topology, whereas GATs replace these with learned attention coefficients, allowing the model to weight neighbours differentially based on feature content. AFGAT follows the same attention mechanism but adopts a deeper architecture and was specifically designed for abstract argumentation frameworks. Architectural details are given in Section 4.

3. Hard Arguments and Their Impact on Algorithm Guidance

In [8], challenging instances are introduced as frameworks where credulously accepted arguments lie outside the grounded extension. We refine this via *hard arguments*, defined through initial labellings and fixpoint propagation.

Definition 3 (Initial Labelling)**.** Let $F = (\mathsf{Args}, R)$ be an AF and $q \in \mathsf{Args}$ an argument with $(q,q) \notin R$ and $q \notin \mathsf{gr}(F)^+$, i.e. q is not self-attacking and is not attacked by the grounded extension. Let $I = \mathsf{gr}(F) \cup \{q\}$ be the set of arguments initially forced *in*. The *initial labelling* $\ell_0 : \mathsf{Args} \to \{in, out, must\ out, blank\}$ is defined as:

- $\ell_0(a) = in$ if $a \in I$;
- $\ell_0(a) = out$ if $a \in I^+$;
- $\ell_0(a) = must\ out$ if $a \in I^- \setminus I^+$;
- $\ell_0(a) = blank$ otherwise.

Definition 4 (Propagated Initial Labelling)**.** The *propagated initial labelling* ℓ^* is the least fixpoint of the following update rules applied to an initial labelling ℓ_0, where a is *blank*:

- If all attackers of a are labelled *out* or *must out*, set $\ell(a) = in$ and $\ell(b) = out$ for all $b \in a^+$.
- If any attacker of a is labelled *in*, set $\ell(a) = out$.

Note that each rule application strictly decreases the number of *blank* arguments, thus, as Args is finite, the rule application terminates after finitely many steps. Furthermore, ℓ^* is uniquely determined, since no argument $a \in args$ can simultaneously have all attackers labelled *out* or *must out* and at least one attacker labelled *in*.

Definition 5 (Hard Argument)**.** Let $F = (\mathsf{Args}, R)$ be an AF, $q \in \mathsf{Args}$, and let ℓ^* be the propagated initial labelling w.r.t. q. Define $S = \{a \in \mathsf{Args} \mid \ell^*(a) = in\}$. We say that q is *hard* if $q \in S$, S is not admissible, $\exists a \in \mathsf{Args} \setminus S$ with $\ell^*(a) = blank$, and there is no $a \in \mathsf{Args}$ with $\ell^*(a) = must\ out$ such that none of its attackers is labelled *blank*. The *hardness* $h_\sigma(q)$ is the minimum number of branching steps required by any complete solver to decide DC-σ for q in F. We note that $h_{\mathsf{gr}}(q)$ is 0 for each $q \in \mathsf{Args}$ and $h_{\mathsf{co}} = h_{\mathsf{pr}}$.

The following example illustrates that computational effort depends on the query argument, even within the same extension.

Example 1. Consider the AF in Figure 1. For query 5, ℓ^* determines the admissible set $\{1,3,5\}$, so 5 is not hard. For query 1, propagation is inconclusive, making 1 a hard argument with $h_{\mathsf{pr}}(1) = 1$.

4. Experiment Design

The goal of our experiments is to systematically evaluate the predictive performance of different graph-based features and neural architectures for co-admissibility prediction, and to assess whether the best-performing combination can improve the efficiency of the SMART solver on challenging benchmark instances. We structure the evaluation across three datasets of increasing difficulty and scale.

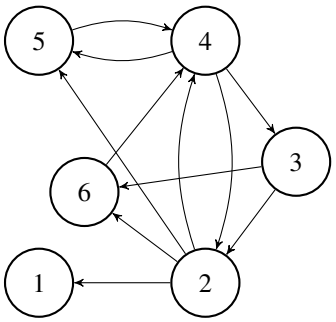

Figure 1. An abstract argumentation framework with an empty grounded extension and a unique preferred extension $\{1,3,5\}$.

4.1. Datasets

For each dataset and selected query argument q, node features are computed over the entire AF, with labels assigning 1 to arguments co-admissible with q and 0 otherwise. The **KWT** dataset consists of 1,000 AFs generated with the KWT graph generator [21][3], yielding 253,227 training, 19,177 validation, and 200,226 test instances after filtering for credulously accepted and hard query arguments. The **ERAF** (Erdős–Rényi Argumentation Frameworks) dataset [22] consists of 2,000 AFs with 100 arguments each, reported as particularly challenging for learning-based methods [8], yielding 155,300 training, 19,200 validation, and 15,800 test instances. For the **ICCMA** dataset, we use the ICCMA 2025 main track benchmark[4], restricting evaluation to the 48 hard and credulously accepted query arguments. Since each query argument induces a feature vector over all arguments in its AF, this yields 340,905 test instances in total. For training we use instances from ICCMA 2017–2023 filtered for hard and credulously accepted query arguments, yielding 1,106,367 training and 23,789 validation instances.

4.2. Models

We implemented four models: a Random Forest (RF) and three GNN architectures. The RF receives a node feature vector per argument and does not perform any graph message passing. All GNN models share a linear output layer and are trained for 100 epochs using the Adam optimiser (learning rate $= 0.01$) with binary cross-entropy loss. All models are implemented using scikit-learn[5] (RF) and PyTorch Geometric[6] (GNNs).

The **RF** uses 100 estimators. The **GCN** uses two convolutional layers (64 and 16 hidden units, ReLU, dropout 0.2). The **GAT** uses three `GATv2Conv` [20] layers with 4 heads, 64 hidden channels per head in intermediate layers, and dropout 0.2. The **AF-GAT** uses three `GATv2Conv` layers with concatenated heads in intermediate layers and averaged heads in the final layer, configured as (5×5, 3×5, 3×1) (heads $\times$ channels per head), ELU activations, attention dropout 0.1, and layer dropout 0.2.

[3] `https://e.feu.de/26r`
[4] `https://argumentationcompetition.org/2025/benchmarks.html`
[5] `https://scikit-learn.org/stable/`
[6] `https://pytorch-geometric.readthedocs.io/en/latest/`

4.3. Graph Features

We consider 21 node-level features, defined in Table 1, in four categories. *Centrality measures* quantify the structural importance of arguments in the attack graph: in- and out-degree capture how many attackers and attackees an argument has, Katz centrality refines this by weighting neighbours by their own importance, betweenness identifies arguments that bridge different parts of the graph, and closeness reflects how near an argument is, on average, to all others. *Structural properties* such as SCC size, number of SCCs, strong connectivity, irreflexivity, and aperiodicity have been linked to acceptability, but so far mainly at the level of an entire framework rather than individual arguments: Vallati et al. [23] found SCC count, average degree, and aperiodicity predictive of how many preferred extensions an AF has, and Doumbouya et al. [24] showed that strongly connected, symmetric, irreflexive AFs guarantee credulous acceptance for every argument. *Gradual semantics scores* [25–27] assign each argument a continuous strength value in $[0,1]$ via fixed-point iteration: an unattacked argument is maximally strong, and an argument is weakened in proportion to the strength of its attackers. This yields a graded measure of acceptability that centrality measures or structural features alone cannot capture. Lastly, *query-context features* relate each argument to the query q, and incorporate information that can be computed deterministically, such as an argument's relation to the grounded extension or the label it is assigned during initial propagation.

Let $F = (\mathsf{Args}, R)$ be an AF, G its induced directed graph, G^T its transpose, $M \in \{0,1\}^{|\mathsf{Args}|\times|\mathsf{Args}|}$ its adjacency matrix with $M_{ab} = 1$ iff $(a,b) \in R$, and $\lambda_{\max}$ the largest eigenvalue of M. We write $d(a,b)$ for the length of the shortest directed path from a to b in G. All features are z-score standardised before model input.

4.4. Evaluation Metrics

Model selection is based on validation *Matthews Correlation Coefficient* (MCC), defined as

$$\mathrm{MCC} = \frac{\mathsf{TP}\cdot\mathsf{TN} - \mathsf{FP}\cdot\mathsf{FN}}{\sqrt{(\mathsf{TP}+\mathsf{FP})(\mathsf{TP}+\mathsf{FN})(\mathsf{TN}+\mathsf{FP})(\mathsf{TN}+\mathsf{FN})}}, \tag{1}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. MCC is particularly suitable for imbalanced classification tasks as it takes all four entries of the confusion matrix into account [29]. The decision threshold is chosen by maximising MCC over the precision-recall curve on the validation set.

Since arguments not resolved by the propagated initial labelling are the only ones on which a complete solver would branch, we additionally report an *adapted MCC* (aMCC), defined as the MCC computed exclusively over arguments whose label remains blank in ℓ^*. This metric more directly reflects the practical utility of a model for solver guidance.

5. Results

This section reports the results of the feature and architecture comparison on the KWT dataset. We first evaluate all 21 features (Section 4.3) individually, then assess pairwise combinations, and finally extend to higher-order feature sets.

Table 1. The 21 node-level features where $a \in \mathsf{Args}$ denotes the argument for which the feature is computed. Gradual semantics features use fixed-point iteration initialised at $s^{(0)}(a) = 1$. Query-specific features depend on the query argument q.

Feature	**Definition**
In-degree	$\deg^{in}(a) = \lvert a^- \rvert$
Out-degree	$\deg^{out}(a) = \lvert a^+ \rvert$
Avg. degree	$(\deg^{in}(a) + \deg^{out}(a))/2$
In-Katz [28]	$katz(a) = \alpha \sum_b M_{ab} katz(b) + 1$, $\alpha = 0.9/\max_a \deg^{out}(a)$, $b \in \mathsf{Args}$ on G
Out-Katz [28]	As in-Katz, computed on G^T
In-closeness	$close(a) = (\lvert \mathsf{Args} \rvert - 1)/\sum_{b \neq a} d(b,a)$, on G
Out-closeness	As in-closeness, computed on G^T
Betweenness	$between(a) = \frac{1}{\lvert \mathsf{Args} \rvert (\lvert \mathsf{Args} \rvert - 1)} \sum_{\substack{s \neq t \\ s,t \in \mathsf{Args}}} \frac{\pi(s,t\lvert a)}{\pi(s,t)}$, where $\pi(s,t)$ is the number of shortest paths from s to t
Shortest path	$d(q,a)$; set to -1 if no path exists
Shortest path parity	1 if $d(q,a) \geq 0$ and even, 0 otherwise
SCC size	Size of the strongly connected component (SCC) containing a
No. of SCCs	Total number of SCCs in G, uniform over all a
Strong connectivity	1 if G is strongly connected, 0 otherwise; uniform over all a
Irreflexivity	1 if G has no self-attacks, 0 otherwise; uniform over all a
Aperiodicity	1 if G is aperiodic, 0 otherwise; uniform over all a
h-cat [25]	$h\text{-}cat(a) = \frac{1}{1 + \sum_{b \in a^-} h\text{-}cat(b)}$
No-sa [27]	As h-cat, with $no - sa(a) = 0$ for self-attacking arguments $(a,a) \in R$
Max-based [26]	$max(a) = \begin{cases} 1 & a^- = \emptyset \\ \frac{1}{1 + \max_{b \in a^-} max(b)} & \text{otherwise} \end{cases}$
Card-based [26]	$card(a) = \begin{cases} 1 & a^- = \emptyset \\ \frac{1}{1 + \lvert a^- \rvert + \frac{1}{\lvert a^- \rvert} \sum_{b \in a^-} card(b)} & \text{otherwise} \end{cases}$
Query-grounded	1 if $a \in \mathsf{gr}(F)$ or $a = q$, else 0
Det. labelling	ℓ^*-labels: 2 (*in*), 0 (*out*), 1 (otherwise)

Results for the single features are shown in Figure 2, displaying both overall MCC and aMCC values. Several features achieve high aMCC values, with the no-sa categoriser attaining the highest overall score (0.908). Across all centrality measures, the in-variant consistently outperforms the out-variant for the neural network models. Several features prove entirely uninformative, with all models assigning every argument the same label regardless of the input. Both the grounded extension and the deterministic labelling features were uninformative for hard arguments: since the dataset was designed to contain credulously accepted arguments under preferred semantics that lie outside the grounded extension, the grounded extension feature carries little discriminative information for the

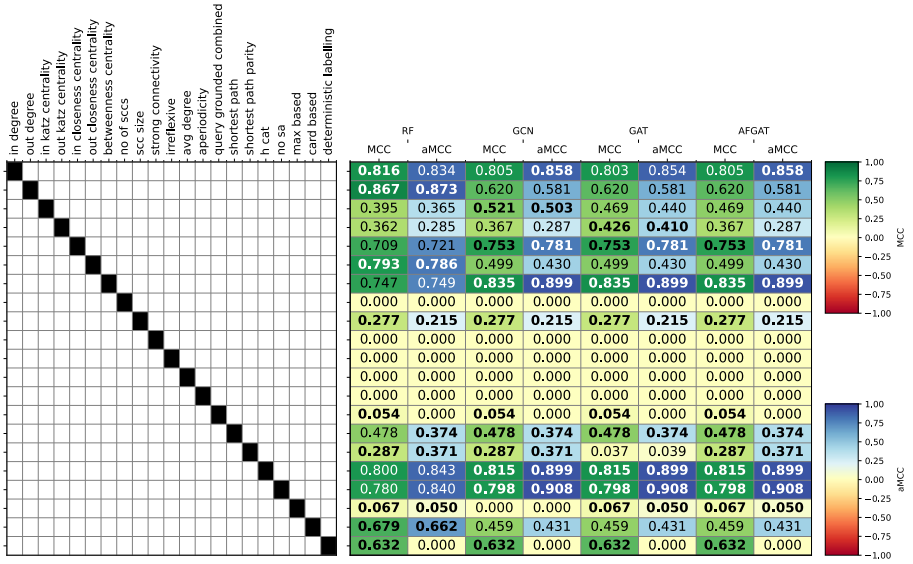

Figure 2. MCC and aMCC(MCC computed on arguments not determined by the initial labelling), per model and feature (KWT dataset).

relevant instances, and all classifiers using the deterministic labelling feature resorted to labelling every argument not assigned *out* as co-admissible.

Each model is further trained on all pairwise feature combinations. The top-10 results are shown in Figure 3. The best-performing combination pairs the no-sa categoriser with the shortest path feature. While shortest path achieves only a mid-range aMCC in isolation, it consistently appears as a strong complementary feature, occurring in four of the top-10 combinations. This can be attributed to its query-sensitivity: whereas centrality and gradual semantics measures are invariant to the query argument, shortest path directly relates each argument to q. While the RF model slightly outperforms the neural models on several feature combinations when considering overall MCC, it consistently falls short on aMCC, suggesting that deeper message-passing models are better suited to predicting co-admissibility for hard arguments.

The top pairwise features (no-sa and shortest path) already achieve near-ceiling aMCC, and higher-order combinations yield no further gain.

6. The i-Categoriser: A Query-Sensitive Refinement of Gradual Semantics

The results above highlight the predictive value of query-relative features. The no-sa categoriser extends the h-categoriser by zeroing out self-attacking arguments before iterating. A natural further refinement is to zero out all arguments labelled *out* or *must out* by ℓ^*, since these cannot belong to any co-admissible set containing q. We call the resulting feature the *i-categoriser*:

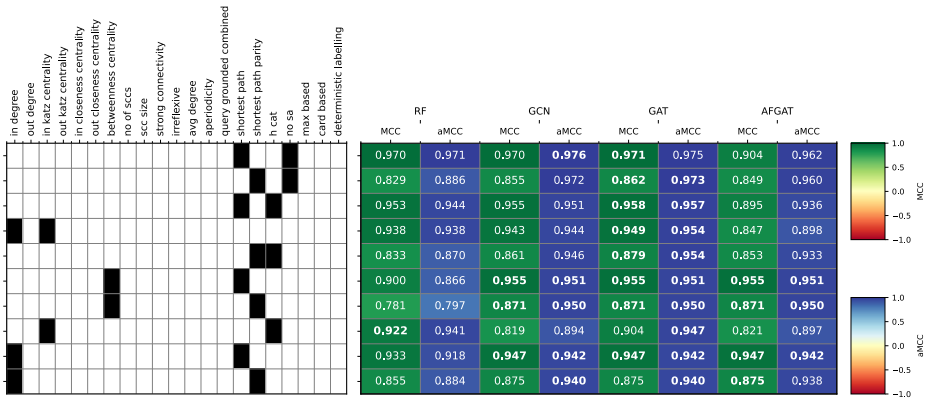

Figure 3. aMCC and MCC per model and 2-feature combination, top 10 (KWT dataset).

Definition 6.

$$i\text{-}cat(F,a) = \begin{cases} 0 & \text{if } \ell^*(a) \in \{out, must\ out\} \\ \dfrac{1}{1+\sum_{b \in a^-} i\text{-}cat(F,b)} & \text{otherwise} \end{cases} \tag{2}$$

where $\ell^*(a)$ is the label assigned by the propagated initial labelling. In the absence of any argument labelled *out* or *must out*, the i-categoriser reduces to the h-categoriser.

Using the i-categoriser alone, all models exceed an MCC of 0.92, while the aMCC values stayed consistent with the no-sa categoriser. The four-feature combination consisting of the i-categoriser, in-degree, shortest path and shortest path parity yields the highest overall MCC of 0.986 and aMCC of 0.977, achieved by the GAT model. The i-categoriser is therefore included in all subsequent experiments.

7. Extended Evaluation and Solver Integration

To validate the findings from Sections 5 and 6, we replicated the single-feature and pairwise feature comparison on the ERAF dataset. The i-categoriser again emerged as the single most informative feature, achieving an aMCC of 0.517 and an overall MCC of 0.784 with the GAT model, substantially outperforming the no-sa categoriser which achieved only an aMCC of 0.280 and an overall MCC of 0.568. Contrary to the KWT results, no pairwise feature combination improved upon the i-categoriser alone. The neural architectures again marginally outperform the RF on aMCC.

Due to the computational cost of feature extraction and training on the large ICCMA graphs, we restricted evaluation to the i-categoriser, in-degree, shortest path, and shortest path parity, individually and in combination, as well as the in- and out-degree combination as a baseline reflecting the feature set most commonly used in prior work. The i-categoriser alone achieved the best single-feature result, with an aMCC of 0.612 and an overall MCC of 0.655 for the AFGAT model. Notably, this is the first dataset

on which AFGAT outperforms GAT, which may indicate that its deeper architecture is better suited to the structural diversity of the ICCMA benchmarks. Adding in-degree, shortest path, and shortest path parity improved performance to an aMCC of 0.626 and an overall MCC of 0.672, while the in- and out-degree baseline reached only 0.434 and 0.460 respectively. Across all three datasets, the four-feature combination consistently and substantially outperformed the degree baseline.

To assess whether these features can guide a complete solver, we trained a GAT model for each dataset on the four-feature combination and integrated its predictions into the SMART solver, serialising it in ONNX format[7] for runtime efficiency. As baselines, we implemented a gold standard, in which the branching heuristic uses the ground-truth co-admissibility relation rather than a learned model, and a model trained only on in- and out-degree features. Ground truth for each blank argument is obtained by checking admissible-set membership with q, so the solver always branches correctly, establishing a lower bound on achievable solving time against which the other approaches are measured. We further implemented a version in which, rather than using the i-categoriser as a static one-time feature, its value is recomputed during search. SMART maintains a stack of arguments that must be labelled *out* in any admissible labelling; when one of these is predicted co-admissible with q, the i-categoriser is rerun after explicitly setting all such arguments to *out* in the initial labelling. Using the initial rather than the current labelling avoids bias from descending a potentially incorrect branch. We refer to the four-feature version as *4F* and the version with online refinement as *4FR*; results are shown in Table 2.

On KWT and ERAF, all solver variants solve every query instance within the 10-minute time limit, and the performance gap relative to the gold standard is attributable primarily to prediction overhead rather than additional search effort.

On the ICCMA dataset, *4F* and *4FR* both solve more instances and reduce mean solving time compared to the degree baseline, with *4FR* holding a slight edge. The impact of online refinement is particularly evident on individual instances: on one example[8], *4FR* recalculated its prediction 30 times and required only 1,019 recursive calls to find an admissible set, compared to 171,348 calls for *4F*, a 140-fold reduction in search effort.

Despite these gains, the gap to the gold standard on ICCMA remains large. We note that the aggregate aMCC is dominated by a small number of large graphs on which the model performs well, masking poor per-instance performance on other graphs. When evaluated per instance, the average aMCC drops to 0.170. This points to the need for more targeted online refinement strategies. One promising direction, which we leave for future work, is incorporating conflict-driven learning analogous to CDCL in SAT solving, which would allow the model to update its predictions dynamically in response to search conflicts rather than solely based on the initial labelling.

To assess the practical overhead introduced by the learning-based guidance, we measured feature extraction and inference time for each query under *4F* that could be solved within the 10-minute limit. Feature extraction accounted for most of this cost, with a mean of 0.517 s per query, reflecting the effort of computing features such as shortest path over large graphs, while ONNX inference itself was comparatively cheap, with a mean of only 4.9 ms. Combined, mean prediction overhead was 0.562 s per query, a small fraction of the mean solving time of 15.975 s for these solved instances (lower than the mean reported in Table 2, which also includes timeouts). This indicates that the gap to

[7] `https://onnx.ai/`

[8] `st_211_11_10_7_31.af`

Table 2. Mean solving time (s) and coverage (%, in parentheses) per solver variant. **Bold**: best excluding gold standard.

Dataset	Gold	4F	4FR	Degree
KWT	0.100 (100)	0.421 (100)	**0.409** (100)	0.441 (100)
ERAF	0.098 (100)	**0.394** (100)	0.403 (100)	0.405 (100)
ICCMA 25	8.827 (100)	167.433 (77)	**167.028** (77)	174.248 (73)

the gold standard on the ICCMA dataset is driven primarily by prediction quality rather than computational overhead.

8. Discussion and Conclusion

The i-categoriser, our proposed query-sensitive refinement incorporating deterministic admissibility information, emerged as the single most informative feature across a systematic evaluation of 21 features and four architectures, consistently dominating performance across datasets and models. Integrating a four-feature set of i-categoriser, in-degree, shortest path, and shortest path parity into the SMART solver improved coverage and reduced mean solving time on ICCMA 2025 benchmarks compared to the degree baseline, with the online refinement variant *4FR* achieving up to a 140-fold reduction in search effort on individual instances.

A key finding is that feature selection has a substantially greater impact on prediction quality than model architecture. Differences among GNN variants were generally small: GAT performed marginally best on the KWT and ERAF datasets, while AFGAT led on ICCMA. The RF consistently fell short on aMCC despite competitive overall MCC, confirming that deeper message-passing models are better suited to hard argument prediction.

Performance on ICCMA instances remains considerably below the gold standard, with average per-instance aMCC of only 0.170. We believe the greatest potential for improvement lies in richer online refinement strategies such as incorporating conflict-driven learning from SAT solving. A complementary direction is to shift the prediction target from co-admissibility status to branching order, more directly aligning the learning objective with the solver's decision process.

Acknowledgments. The research reported here was partially supported by the Deutsche Forschungsgemeinschaft (project "Argumentative Reasoning in Nonsensical Situations", grant 550735820).

References

[1] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence. 1995;77(2):321-57.

[2] Prakken H, Wyner A, Bench-Capon T, Atkinson K. A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. Journal of Logic and Computation. 2015;25(5):1141-66.

[3] Müller J, Hunter A. An Argumentation-Based Approach for Decision Making. In: 2012 IEEE 24th International Conference on Tools with Artificial Intelligence. vol. 1; 2012. p. 564-71.

[4] Vassiliades A, Bassiliades N, Patkos T. Argumentation and explainable artificial intelligence: a survey. The Knowledge Engineering Review. 2021;36:e5.

[5] Craandijk D, Bex F. Deep learning for abstract argumentation semantics. arXiv preprint arXiv:200707629. 2020.

[6] Kuhlmann I, Thimm M. Using graph convolutional networks for approximate reasoning with abstract argumentation frameworks: A feasibility study. In: International Conference on Scalable Uncertainty Management. Springer; 2019. p. 24-37.

[7] Malmqvist L, Yuan T, Nightingale P, Manandhar S. Determining the Acceptability of Abstract Arguments with Graph Convolutional Networks. In: SAFA@ COMMA; 2020. p. 47-56.

[8] Kuhlmann I, Wujek T, Thimm M. On the Impact of Data Selection when Applying Machine Learning in Abstract Argumentation. In: COMMA; 2022. p. 224-35.

[9] Hoffmann S, Kuhlmann I, Thimm M. Enhancing abstract argumentation solvers with machine learning-guided heuristics: A feasibility study. In: Conference on Advances in Robust Argumentation Machines. Springer; 2024. p. 185-201.

[10] Malmqvist L. Approximate solutions to abstract argumentation problems using graph neural networks [PhD thesis]. University of York; 2022.

[11] Craandijk D, Bex F. Effects of Graph Neural Network Aggregation Functions on Generalizability for Solving Abstract Argumentation Semantics. In: CEUR Workshop Proceedings. vol. 3757. CEUR WS; 2024. p. 83-9.

[12] Cibier P, Mailly JG. Graph Convolutional Networks and Graph Attention Networks for Approximating Arguments Acceptability–Technical Report. arXiv preprint arXiv:240418672. 2024.

[13] Hoffmann S, Kuhlmann I, Thimm M. Smart v1.0. In: Solver and Benchmark Descriptions of the Sixth International Competition on Computational Models of Argumentation (ICCMA'25); 2025. p. 37-8.

[14] Caminada MW, Gabbay DM. A logical account of formal argumentation. Studia Logica. 2009;93:109-45.

[15] Nofal S, Atkinson K, Dunne PE. Looking-ahead in backtracking algorithms for abstract argumentation. International Journal of Approximate Reasoning. 2016;78:265-82.

[16] Nofal S, Atkinson K, Dunne PE, Hababeh IO. A New Labelling Algorithm for Generating Preferred Extensions of Abstract Argumentation Frameworks. In: ICEIS (1); 2019. p. 340-8.

[17] Breiman L. Random forests. Machine learning. 2001;45(1):5-32.

[18] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907. 2016.

[19] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y, et al. Graph attention networks. In: International conference on learning representations. vol. 6. Ithaca; 2018. .

[20] Brody S, Alon U, Yahav E. How attentive are graph attention networks? arXiv preprint arXiv:210514491. 2021.

[21] Kuhlmann I, Thimm M. A Discussion of Challenges in Benchmark Generation for Abstract Argumentation. In: Arg&App@ KR; 2023. p. 78-84.

[22] Erdős P, Rényi A. On random graphs I. Publ math debrecen. 1959;6(290-297):18.

[23] Vallati M, Cerutti F, Giacomin M. Predictive Models and Abstract Argumentation: The Case of High-Complexity Semantics. The Knowledge Engineering Review. 2019 Jan;34:e6.

[24] Doumbouya MB, Kamsu-Foguem B, Kenfack H. Argumentation and graph properties. Information Processing & Management. 2016;52(2):319-25.

[25] Besnard P, Hunter A. A logic-based theory of deductive arguments. Artificial Intelligence. 2001;128(1-2):203-35.

[26] Amgoud L, Ben-Naim J, Doder D, Vesic S. Acceptability semantics for weighted argumentation frameworks. In: Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017). International Joint Conferences on Artifical Intelligence (IJCAI); 2017. p. 56-62.

[27] Beuselinck V, Delobelle J, Vesic S. A principle-based account of self-attacking arguments in gradual semantics. Journal of Logic and Computation. 2023;33(2):230-56.

[28] Katz L. A new status index derived from sociometric analysis. Psychometrika. 1953;18(1):39-43.

[29] Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics. 2020;21(1):6.