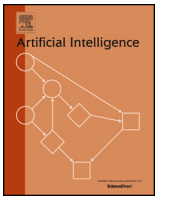
Artificial Intelligence 336 (2024) 104209

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Approximating problems in abstract argumentation with graph convolutional networks

Lars Malmqvist *, Tangming Yuan, Peter Nightingale

University of York, Department of Computer Science, Deramore Lane, Heslington, York, YO10 5GH, UK

ARTICLE INFO

Dataset link: https://github.com/lmlearning/AFGraphLib

ABSTRACT

In this article, we present a novel approximation approach for abstract argumentation using a customized Graph Convolutional Network (GCN) architecture and a tailored training method. Our approach demonstrates promising results in approximating abstract argumentation tasks across various semantics, setting a new state of the art for performance on certain tasks. We provide a detailed analysis of approximation and runtime performance and propose a new scheme for evaluation. By advancing the state of the art for approximating the acceptability status of abstract arguments, we make theoretical and empirical advances in understanding the limits and opportunities for approximation in this field. Our approach shows potential for creating both general purpose and task-specific approximators and offers insights into the performance differences across benchmarks and semantics.

1. Introduction

The field of argumentation encompasses the study of how arguments are constructed, analyzed, evaluated, and used in communication to persuade or convince others of a particular viewpoint or claim [1]. Abstract argumentation is a specific area within the broader field of argumentation that focuses on the formal representation and analysis of arguments in a structured manner. It deals with the abstract structure of arguments, independent of specific content or context, and aims to provide a framework for reasoning about the acceptability and relationships between arguments [2]. It has been used in numerous fields to reason about the acceptability of argumentative structures and holds great promise as a formal approach to certain reasoning tasks in Artificial Intelligence (AI). Examples include legal argumentation [3], where abstract argumentation can help model reasoning under disagreement, the structure of argument systems, and aid the development of dispute tactics. Applications have also been found in Intelligence Analysis [4] to determine which items of intelligence are internally consistent, in multi-agent systems as a part of communication protocols [5], in misinformation detection in tweets [6], and even in maritime safety [7] to determine consistency of sensor readings.

Acceptability in abstract argumentation concerns whether a given argument belongs to specific positions within an argumentation framework. An admissible position represents a subset of arguments that are internally consistent and defend themselves against all attacking arguments. For example, when investigating an academic misconduct case, one possible position could involve a subset of evidence supporting the case of collusion, while another position might involve a subset of evidence for the case of false authorship. When evaluating arguments, the goal is typically to find a position that includes as much available evidence as possible. This position is known as the preferred extension and is one of the most crucial solutions in abstract argumentation systems. When an argumentation

* Corresponding author.
E-mail addresses: lama@thetechcollective.eu, lars.malmqvist@york.ac.uk (L. Malmqvist).

https://doi.org/10.1016/j.artint.2024.104209
Received 29 August 2023; Received in revised form 15 August 2024; Accepted 19 August 2024

Available online 29 August 2024

system has multiple preferred extensions, one obvious means to determine the strength of an argument is to count how many preferred extensions it appears in; the more it appears, the stronger it is [8]. Various notions of acceptability have been developed based on preferred extensions. For example, an argument is sceptically accepted if it is a member of every preferred extension, and credulously accepted if it is a member of at least one preferred extension [2].

The decision problems associated with acceptability are theoretically difficult (e.g. NP-hard or Π_2^P-complete), except for acceptance under the grounded semantics, another notion of acceptance which represents the most sceptical stance possible and can be computed polynomially [9]. The lack of a general polynomial time solution has led to an interest in approximate approaches for runtime critical applications.

There have been several previous works that have adopted approximate approaches to abstract argumentation [10,11]. While they have had some success, neither of these approaches has considered a large and varied set of argumentation frameworks across different semantics. Scalability, however, is important for applications with large argumentation frameworks (AFs), such as social media.

In this article, we develop a new approximation approach based on a customised version of the Graph Convolutional Network (GCN) architecture developed by Kipf and Welling [12], AFGCN, combined with a training approach tailored to abstract argumentation. We then conduct a substantial number of experiments to test the capabilities and limitations of this approach. This solver approach was tested at the 2021 International Competition on Computational Models of Argument (ICCMA), where it won 4 out of 6 categories in the approximate track. AFGCN has been successfully applied to a large AF extracted from Twitter, which comprised 392 posts and 9088 comments, for fake news detection [13]. It was shown that the combined use of language sources i.e., tweets together with the (labelled) graph significantly outperforms using language sources alone [14].

In this article, we make the following contributions:

- We present systematic results for approximating abstract argumentation tasks across all the current ICCMA semantics.
- We set a new state of the art for performance on two previously studied abstract argumentation tasks: credulous and sceptical acceptability under the preferred semantics (referred to as DC-PR and DS-PR respectively) [10,11].
- We propose an improved Graph Convolutional Network architecture and runtime implementation for this purpose.
- We demonstrate two different ways for grounded reasoning to be combined with a neural network model with the aim of improving the accuracy of approximation.
- We provide a detailed analysis of approximation performance across the 11 benchmark sets that formed part of ICCMA 2019.
- We provide a detailed analysis of runtime performance of our GCN approximation approach.
- We present a new scheme for evaluation for the approximation of abstract argumentation tasks that better reflect performance when classifying different types of argumentation frameworks.

By making these contributions, we advance the state-of-art for approximating the acceptability of abstract arguments and make theoretical and empirical advances in understanding the limits and opportunities for approximation in this field. Because of the large number of experiments and evaluations conducted in the course of this research, only the most important results are covered in the main text. The appendix covers the remainder.

2. Abstract argumentation

Abstract argumentation is a way of formalising the representation of conflicting claims [15] using an intentionally minimalist approach. In abstract argumentation, (abstract) arguments are composed into argumentation frameworks that contain only the arguments themselves and the relationships of conflict between them.

The origin of abstract argumentation is in a seminal paper by Dung [2] that presented the general theory of argumentation frameworks and related them to a number of other logical formalisms. We will go through some of the key definitions from this paper in order to cover the necessary background for the subsequent discussions of the rules of interpretation that can be applied to argumentation frameworks and the approaches one can take to solving them. Definitions here are given using an extension based approach, but an equivalent labelling based approach is equally common in the literature [16].

2.1. Definitions

Definition 2.1 *(Argumentation framework).* An argumentation framework is a tuple, $F = \langle args, atts \rangle$ in which $args$ is a finite set of arguments and $atts \subseteq args \times args$ defines a relation of attack.

To say that a attacks b is hence the same as saying that $(a, b) \in atts$. If $S \subseteq args$ and $a \in args$ we can extend this nomenclature by saying that a attacks S iff there exists $b \in S$ such that $(a, b) \in atts$. In a parallel manner we can say that S attacks a iff there exists $b \in S$ such that $(b, a) \in atts$.

We can also define a similar notion of defence.

Definition 2.2 *(Defence).* Given an argumentation framework $F = \langle args, atts \rangle$, an argument $a \in args$ is defended by a set $S \subseteq args$ if, for each $b \in args$ such that $(b, a) \in atts$, there exists a $c \in S$ such that $(c, b) \in atts$.

Definition 2.3 *(Attacking and attacked).* Given an argumentation framework $F = \langle args, atts \rangle$, the set of all attacking arguments of a subset of the arguments $S \subseteq args$ can be written as $S^- = \{b \mid \exists a \in S : (b, a) \in atts\}$. The set of all attacked arguments can be written as $S^+ = \{b \mid \exists a \in S : (a, b) \in atts\}$.

This notation is convenient in that it also lets us define notions of range and negative range.

Definition 2.4 *(Range and negative range).* Given an argumentation framework $F = \langle args, atts \rangle$ and set $S \subseteq args$, the range of S can be defined as $S \cup S^+$. The negative range of S can be defined as $S \cup S^-$.

The range is thus the union of a set of arguments S and those arguments attacked by S, whereas the negative range is a set S and all arguments that attack S.

Definition 2.5 *(Characteristic function).* Given an argumentation framework F $= \langle args, atts \rangle$, the characteristic function $\chi : 2^{args} \rightarrow 2^{args}$ of F is defined as $\chi_F(S) = \{a \in args \mid a$ is defended by $S\}$ for each $S \subseteq args$.

The characteristic function returns the set of arguments defended by a given subset of the arguments of an argumentation framework. The last basic concepts required before we can move on from this section are those of conflict-freeness, admissibility, and acceptability.

Extensions are subsets of arguments that are considered to be collectively acceptable. Extensions are evaluated based on semantics that define rules for which sets of arguments can be accepted together. We provide the relevant definitions of semantics in Section 2.2 below.

Definition 2.6 *(Conflict-freeness).* Given an argumentation framework $F = \langle args, atts \rangle$, a given subset $S \subseteq args$ is said to be conflict-free iff there does not exist $(A, B) \in atts$ *with* A, B $\in S$.

The notion of conflict-freeness implies that there are no internal conflicts within an extension and is a building block of all semantics used in this paper.

Definition 2.7 *(Admissibility).* A subset $S \subseteq args$ of an argumentation framework $F = \langle args, atts \rangle$ is admissible iff it is conflict-free and $S \subseteq \chi_F(S)$.

The definition above states that an admissible set is a set that is conflict-free and defends itself from all attacks as in Definition 2.2. Finally we define acceptability of individual arguments with respect to a given semantics.

Definition 2.8 *(Credulous and Sceptical Acceptability).* Given an argumentation framework $F = \langle args, atts \rangle$, an argument $A \in args$ is *credulously acceptable* with respect to a given semantics iff there exists a set $ext \subseteq args$ where $A \in ext$ and ext is an extension under the given semantics. A is *sceptically acceptable* iff $A \in ext$ for *all* sets $ext \subseteq args$ where ext is an extension under the given semantics.

2.2. Argumentation semantics

The semantics of an argumentation framework define the rules under which a set of arguments can be accepted. That is to say which sets of arguments contained in an argumentation framework can be said to constitute acceptable solutions under the given semantics. The original paper by Dung defined the four "classic" semantics: grounded, complete, preferred, and stable [2].

However, a number of additional semantics have been proposed. In the following, we will cover the four "classic" semantics and three additional semantics: ideal [17], stage [18], and semi-stable [19] that all feature prominently in the literature. This will be done based on the definitions covered above, but we will also attempt to give a more "qualitative" account of what each semantics encapsulates and how they differ in the kinds of solutions they allow. The derivations of each semantics is kept to a minimal, but complete level. More elaborate derivations exist in the papers cited above.

2.2.1. Complete semantics

The most basic semantics for most purposes are complete semantics. Many other common semantics are special cases of complete semantics. Complete semantics can be defined as a fix point of a conflict-free subset of the arguments of an argumentation framework. That is to say for a subset S of an argumentation framework, F, the subset forms a complete extension iff $\chi_F(S) = S$ and S is conflict-free. That means in practical terms that a complete extension is an extension that defends itself and also contains all the elements defended by the extension. Qualitatively, one can think of a complete extension as a reasonable or at least defensible position given the evidence.

Definition 2.9 *(Complete Semantics).* An extension $S \subseteq args$ is a complete extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is admissible and S contains all the arguments it defends, i.e., $\forall a \in args$, if $\forall b \in args$ such that $(b, a) \in atts, \exists c \in S$ such that $(c, b) \in atts$, then $a \in S$.

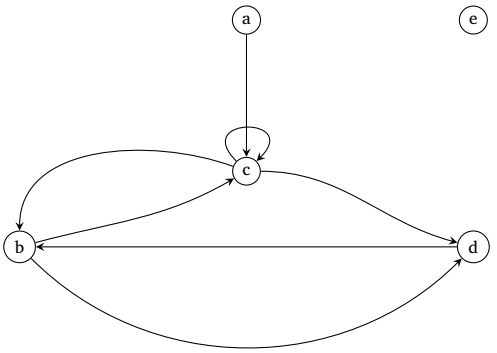


Fig. 1. Example to distinguish grounded, complete, and preferred semantics.

Fig. 2. Example of ideal semantics.

2.2.2. Grounded semantics

The grounded extension is the subset-minimal complete extension. That is to say that if S is a complete extension then it is also grounded iff there does not exist another complete extension, $C \mid C \subset S$. Qualitatively, one can think of the grounded extension as the most sceptical position one can take vis-à-vis the evidence.

Definition 2.10 *(Grounded Semantics).* An extension $S \subseteq args$ is the grounded extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is the minimal (with respect to set inclusion) complete extension of F.

2.2.3. Preferred semantics

A preferred extension in contrast is a subset-maximal complete extension. That is to say that if S is a complete extension then it is also preferred iff there does not exist another complete extension, $C \mid S \subset C$. A preferred extension can be thought of as a position that tries to incorporate as much as possible of the available evidence in formulating a defensible position.

Definition 2.11 *(Preferred Semantics).* An extension $S \subseteq args$ is a preferred extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is a maximal (with respect to set inclusion) complete extension of F.

Consider Fig. 1. This framework has a grounded extension $\{a, e\}$, three complete extensions $\{\{a, e\}, \{a, e, b\}, \{a, e, d\}\}$, and two preferred extensions $\{\{a, e, b\}, \{a, e, d\}\}$.

2.2.4. Ideal semantics

In many cases, the scepticism of the grounded extension proves too severe for practical applications and a slightly less severe form of scepticism is called for. This is provided by the ideal semantics, which can be defined as the largest admissible subset of the arguments of an argumentation framework in which all the elements are members of every preferred extension. This is still a sceptical position, but can vary from the grounded extension.

Definition 2.12 *(Ideal Semantics).* An extension $S \subseteq args$ is the ideal extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is the maximal (with respect to set inclusion) admissible set that is contained in every preferred extension of F.

Ideal semantics can be less sceptical than grounded semantics. Consider the argumentation framework in Fig. 2. This framework has an ideal extension $\{a\}$ and an empty grounded extension.

2.2.5. Stable semantics

While the stable extension can also be shown to be a complete extension it is not usually defined as such. Instead, a stable extension, which may or may not exist for a given argumentation framework, is defined as a conflict-free extension whose range is equal to the total set of arguments in the argumentation framework. That is to say, the stable extension takes the "if you're not with us you're against us" approach by ensuring that every argument is either a member of the extension or attacked by the extension.

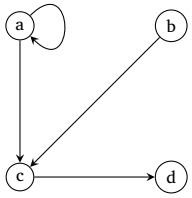


Fig. 3. Example of stage and stable semantics.

Table 1
Definitions of abstract argumentation semantics, adapted from [16].

Semantics	Definition
Complete	An extension S is complete iff it is admissible and it includes all arguments that it defends.
Grounded	An extension S is grounded iff it is complete and subset minimal.
Preferred	An extension S is preferred iff it is complete and subset maximal.
Stable	An extension S is stable iff it is conflict-free and $S \cup S^+$ contains all arguments in the argumentation framework.
Semi-Stable	An extension S is semi-stable if it is complete and $S \cup S^+$ is subset maximal.
Stage	An extension S is stage iff it is conflict-free and $S \cup S^+$ is subset maximal.
Ideal	An extension S is ideal iff it is an admissible subset of all preferred extensions and is subset maximal.

Definition 2.13 *(Stable Semantics).* An extension $S \subseteq args$ is a stable extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is conflict-free and S attacks all arguments not in S, i.e., $\forall a \in args \setminus S, \exists b \in S$ such that $(b, a) \in atts$.

2.2.6. Semi-stable and stage semantics

The completeness of the stable extension's binary division of the argumentation framework is a desirable feature in some applications, where undecidability is an issue. However, the stable extension does not exist in all argumentation frameworks. Therefore, two alternatives have been proposed that try to maximise the range of extensions, but can be shown to exist for all argumentation frameworks.

The first of these, stage semantics, can be defined formally as the conflict-free set that maximises range. That is to say S is a stage extension [20] iff S is conflict-free and there does not exist a conflict-free extension C such that $S^+ \subset C^+$. The semi-stable semantics is defined similarly only it starts from an admissible set rather than a conflict-free one [19].

Definition 2.14 *(Semi-Stable Semantics).* An extension $S \subseteq args$ is a semi-stable extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is a complete extension of F and $S \cup S^+$ is maximal (with respect to set inclusion) among all complete extensions of F, where $S^+ = \{a \in args \mid \exists b \in S$ such that $(b, a) \in atts\}$.

Definition 2.15 *(Stage Semantics).* An extension $S \subseteq args$ is a stage extension of an argumentation framework $F = \langle args, atts \rangle$ iff S is a conflict-free set of F and $S \cup S^+$ is maximal (with respect to set inclusion) among all conflict-free sets of AF, where $S^+ = \{a \in args \mid \exists b \in S$ such that $(b, a) \in atts\}$.

Consider the argumentation framework in Fig. 3. This framework has one semi-stable extensions $\{b, d\}$, which is also a stage extension. It does not have a stable extension.

The difference between the evaluation rules leads to significant differences in the computational approaches that are taken towards them. While many solvers deploy a similar starting point for generating solutions across semantics, there are substantial differences in how they are computed leading to high variability in performance. The complexity of reasoning tasks equally vary substantially by semantics [21,22] and it is not always the case that similar semantics have similar computational complexity, which we will discuss later in this article.

2.2.7. Summary of semantics

These seven semantics can be summarized as per Table 1 [16]. Some more examples are shown in Fig. 4. These are all the semantics we shall concern ourselves with in this article and are also the ones used for recent ICCMA competitions (2017, 2019, 2021). Note that the abbreviations shown in Table 2 are used extensively in this article.

2.3. Basic deep neural network (DNN) concepts

There are a variety of DNN specific terms that relate to the architecture and training of DNNs, which are relevant to the understanding of the material development in this article. Below we define these key terms.

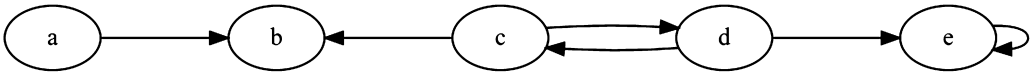


Fig. 4. This framework has the following extensions by semantics. Grounded and ideal = $\{\{a\}\}$, complete = $\{\{a\},\{a,c\},\{a,d\}\}$, stable and semi-stable = $\{\{a,d\}\}$, preferred = $\{\{a,c\},\{a,d\}\}$.

Table 2
Abbreviations of problems and semantics used extensively in this article.

Abbreviation	Description
DC-CO	The credulous acceptability problem under complete semantics.
DC-PR	The credulous acceptability problem under preferred semantics.
DC-ST	The credulous acceptability problem under stable semantics.
DC-SST	The credulous acceptability problem under semi-stable semantics.
DC-STG	The credulous acceptability problem under stage semantics.
DS-CO	The sceptical acceptability problem under grounded semantics.
DS-PR	The sceptical acceptability problem under preferred semantics.
DS-ST	The sceptical acceptability problem under stable semantics.
DS-SST	The sceptical acceptability problem under semi-stable semantics.
DS-STG	The sceptical acceptability problem under stage semantics.
DS-ID	The sceptical acceptability problem under ideal semantics.

ReLU The Rectified Linear Unit (ReLU) is a nonlinear function that is used in neural networks to introduce nonlinearity into the network. It is defined as $f(x) = max(0, x)$. ReLU has been shown to improve the convergence of neural networks and can help prevent overfitting [23].

Dropout Dropout is a method to prevent overfitting of neural network models. Nonoutput units in the network are dropped (i.e. the unit and all its incoming and outgoing connections are temporarily removed) at random during training [23]. The practical implementation is via dropout layers that may be added after any nonoutput layer of a DNN.

Repeating blocks A repeating block is a sequence of layers that is repeated as a block multiple times in the DNN architecture. For example, a repeating block could be a computation layer followed by a dropout layer [23].

Residual connection A residual connection is a type of skip connection in which the input to a layer is directly connected to the output of that layer, without passing through any intermediate computation. This type of connection can be useful in preventing the vanishing gradient problem and can improve the training speed of deep neural networks [23].

Loss function A loss function quantifies the error between predicted values and actual values of training examples. It is minimised during training [23].

Learning rate The learning rate is a parameter of the DNN training process that controls how much the weights of the network are updated in response to the error gradient [23].

ADAM ADAM (Adaptive Moment Estimation) is a method for training neural networks that is based on gradient descent. The main difference between ADAM and other methods is that it uses a different learning rate for each parameter, which is adapted based on the parameter's gradient. This allows the learning rate to be automatically adjusted as the training progresses, which can lead to faster and more efficient training [23].

Graph embeddings A graph (or node) embedding is a mathematical mapping of the vertices of a graph into a Euclidean space, such that the distances between the vertices in the graph correspond in some way to the distances between the points in the Euclidean space. It can be thought of as a representation of the graph structure that is amenable to mathematical analysis.

Graph embeddings are used in a range of graph analysis applications including node classification, link prediction, clustering, and visualisation either directly or as additional input features to machine learning algorithms. There are several approaches to generating graph embeddings that are suited to different use cases. Goyal and Ferrara define three main approaches [24].

Feature vector A feature vector is a numerical representation of an object's characteristics. In the context of graph neural networks, a feature vector is associated with each node in the graph and contains information about the node's properties or features. The feature vector serves as input to the neural network, allowing it to learn and exploit patterns in the node features for tasks such as node classification or graph representation learning.

2.4. Convolutional graph neural networks

Convolutional graph neural networks (CGNNs) draw on the popularity and success of traditional Convolutional Neural Networks (CNNs), in particular in computer vision. Convolution is a mathematical operation that combines two functions to produce a third function expressing how the shape of one is modified by the other. In the context of CNNs, convolution is typically applied to two-dimensional grid-structured data, such as images, where it involves sliding a learnable filter over the input to extract local features and create a feature map. A filter in this context is a linear function followed by an activation function (such as ReLU), where the input of the linear function is a contiguous part (often a sub-square) of the grid-structured data.

In graph neural networks, convolution is adapted to operate on graph-structured data. The goal is to learn a function that aggregates data from a node's neighbourhood to compute a new representation for the node. This is achieved by defining a learnable filter that is applied to the node features and the features of its neighbours, weighted by the edges connecting them.

There are multiple definitions of convolution on graphs. One method, used in models like GCN [12] and ChebNet [25], is to define convolution based on the graph Laplacian matrix, which encodes the graph structure and node connectivity. These models learn a filter that is applied to the graph Laplacian to aggregate information from a node's local neighbourhood.

Another approach, used in models like GraphSAGE [26] and GAT [27], stays closer to the conventional CNN by defining convolution based on a node's spatial relationships. In these models, the convolution operation involves aggregating information from a node's immediate neighbours, similar to how a filter slides over a local patch in a grid-structured CNN.

The conventional CNN definition refers to the application of learnable filters to grid-structured data, such as images, where the filters are convolved with local patches of the input to extract features. The spatial-based graph convolution approaches, like GraphSAGE and GAT, adapt this idea to graph-structured data by defining convolution based on a node's local neighbourhood, analogous to the local patches in a grid-structured CNN.

The most relevant model for our work is the Graph Convolutional Network (GCN) by Kipf and Welling [12]. This model learns a function on a graph given a set of node features and a representation of graph structure.

In several areas, particularly computer vision, transformer based architectures [28] have become dominant in recent years. There has also been research into graph transformers [29] that has achieved a level of success. Why then have we not considered this architecture in our work?

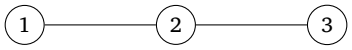
First, the success of the transformer architecture is much less marked in graph applications than it is for some other areas (e.g. computer vision). Second, the flexibility of the basic GCN architecture means that it is easily adaptable to special cases such as ours. This easy adaptability is no doubt why it is still commonly used in practical applications, despite the availability of other architectures. Also, the basic GCN model works well in cases where graph structure rather than node features is the predominant carrier of information, which is why we prefer it to other models within the same family such as Graph Attention Networks (GAT) [27] or GraphSage [30].

2.4.1. Definition of graph convolutional networks

We cover the key definitions of the GCN architecture [12] in this section.

Definition 2.16 *(Adjacency Matrix).* An adjacency matrix is a square matrix used to represent a finite graph. The matrix has a row and column for every vertex in the graph, and the entry in row i and column j is 1 iff there is an edge from vertex i to vertex j, and 0 otherwise.

Consider an undirected graph with 3 nodes and 2 edges:



The adjacency matrix A for this graph would be:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Definition 2.17 *(Graph Convolutional Network).* A model that learns a function f on a graph $G(V, E)$, using inputs X, a matrix representation of node features, and A, the adjacency matrix of G.

Each layer in a GCN can be configured to produce an arbitrary number of output features, which are learned representations of the input graph. These output features are vectors associated with each node in the graph, and they encode information about the node's local neighbourhood and its role within the overall graph structure.

Definition 2.18 *(Layer-wise propagation).* Each layer in a GCN can be written as a non-linear function $H^{(l+1)} = f(H^{(l)}, A)$, where $H^{(0)} = X$ and the output of the final layer is the output of the GCN [12].

Each layer of the GCN follows a propagation rule that maps an input representation to an output representation following a given rule. The propagation rule used by GCN makes use of the following elements:

- **Diagonal node degree matrix** (**D**): A diagonal matrix where each entry on the main diagonal corresponds to the degree of a node in the graph. The degree of a node is the number of edges incident to it.
- **Weight matrix** (**W**): A learnable matrix of weights that determines the importance of each feature in the input feature vector during the transformation. Individual weight w_{ij} corresponds to the j-th feature of node i.
- **Activation function** (σ): A non-linear function (e.g. ReLU or sigmoid) applied element-wise to the output of the linear transformation.

Definition 2.19 *(GCN propagation rule).* $f(\mathbf{H}^{(l)}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)})$

- $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix
- $\hat{\mathbf{D}}$ = Diagonal node degree matrix of $\hat{\mathbf{A}}$
- $\mathbf{W}^{(l)}$ = The weight matrix for layer l
- σ = A non-linear activation function applied element-wise [12]

This propagation rule uses two key tricks to improve on a naïve update rule that would simply multiply the adjacency matrix with the weights and layer-wise representations. First, self-loops are created by adding the identity matrix to the adjacency matrix. This ensures that a node's own information can propagate to itself. Without this, the node would receive only information from its neighbours. Second, the adjacency matrix is normalized to avoid changing the scale of the feature vectors. This helps ensure numerical stability. Overall, the GCN model provides a remarkably simple, flexible model that has proven effective in many practical applications.

As an example, suppose each node in the graph from the previous example has a 2-dimensional feature vector:

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

A single GCN layer with a 2×2 weight matrix W and ReLU activation function would transform these features as follows, producing output $H^{(1)}$ comprising a new set of node features learned by the GCN layer:

$$H^{(1)} = \text{ReLU}(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}XW^{(0)})$$

GCN methods have been applied to abstract argumentation with some success [10,31], however neither of these approaches have considered a large and varied set of argumentation frameworks across different semantics. We shall consider these approaches below. Graph Neural Networks have also been used more successfully in the related fields of Automated Theorem Proving [32] and the Graph Colouring Problem [33].

3. Related work

Computing the acceptability of arguments in an abstract argumentation framework (under various semantics) has been studied extensively, however almost all work in this area has been on exact solution methods. The 2015 survey of Charwat et al. [34] includes reductions to boolean satisfiability (SAT), constraint programming (CSP), answer-set programming (ASP) and other formalisms, as well as methods that operate directly on the abstract argumentation framework. All methods surveyed by Charwat et al. are exact. To our knowledge, only two approximation methods have been proposed. The first uses the grounded extension as an approximation for other semantics. The second involves training a graph neural network to predict acceptability. In this section we relate both these prior approaches to our proposed approach.

Cerutti et al. [35] found that the grounded extension (which is unique and can be computed in polynomial time) is a close approximation of acceptability for common benchmark argumentation frameworks, with several semantics and with both credulous and sceptical acceptability tasks. The idea of using the grounded extension as an approximation of other semantics is implemented in the solver HARPER++ [36], an entrant in the ICCMA 2021 competition. We have extensively compared the accuracy of grounded reasoning to that of our GCN models (in Section 5). Also, we have experimented with two ways of combining the grounded extension with a GCN model, both described in Section 4.2.2. Grounded reasoning is perfectly accurate for one task (DS-CO), and the *Grounded* benchmark set, but for other tasks/benchmarks we typically find that our GCN-based models are more accurate.

Kuhlmann and Thimm [10] trained a GCN to approximate the set of acceptable arguments with the preferred semantics, and with credulous acceptability. They demonstrated that the GCN can be far faster than a SAT-based sound and complete solver, with the SAT-based solver taking on average 17,000 times longer per argument to classify all arguments in a set of benchmark argumentation frameworks. However, accuracy of the GCN model was somewhat limited, with overall accuracy of approximately 63% when tested on benchmark graphs from the ICCMA competition. Kuhlmann and Thimm identified the balance between acceptable and non-acceptable arguments in the training data (where the majority are non-acceptable) as a particular problem in training a GCN. We have compared

Kuhlmann and Thimm's results (with and without balancing of the training data) to our results in Section 5.3, and found that we are able to substantially improve accuracy.

Craandijk and Bex [31] developed the argumentation graph neural network (AGNN), a deep learning architecture based on message passing. The model is trained with 32 message passing steps, but may be executed with any number of steps since the same function is applied at each step. The training process minimises the loss at every step (as opposed to only the final step) with the goal of learning a convergent message passing process. Experimental results show that a high degree of accuracy can be achieved on random argumentation frameworks with 25, 50, 100, or 200 arguments. However, convergence slows for the larger frameworks, and for frameworks of size 100 or 200 the model has not fully converged after 1,000 message passing steps. Our goal is approximate solving of argumentation frameworks at scale, hence we train and evaluate our models using a set of frameworks from the ICCMA competition with up to 10,000 arguments. We have compared the AGNN model (trained exactly as described by Craandijk and Bex) to our models experimentally in Section 5.3, and we found that our models are substantially more accurate on our test set of competition benchmarks.

Approximation methods based on graph neural networks have also been applied to other problems in abstract argumentation. Craandijk and Bex [37] studied the problem of *enforcement*: determining a set of attack relations to add or delete to enforce the acceptability of a set of arguments while minimising the number of changes to the argumentation framework. They developed the enforcement graph neural network (EGNN), a deep learning architecture similar to the AGNN. The enforcement problem is cast as a Markov decision process (where actions modify the argumentation framework), and the EGNN is trained using reinforcement learning to predict the reward of each potential action. The method could provide inspiration for future work on reinforcement learning for acceptability problems.

Klein et al. [38] addressed an algorithm selection task in abstract argumentation using graph neural networks. Given an argumentation framework and a query argument, the task was to predict the most efficient one of three exact argumentation solvers for sceptical acceptance under the preferred semantics. They applied 4 classical machine learning methods and 3 types of graph neural network in a supervised setting. They found that GraphSage [26] provided the highest accuracy of 0.71 while the Graph Isomorphism Network (GIN) [39] was slightly better in terms of total solver runtime. However, the results do not seem conclusive, with 6 of the 7 models performing similarly.

Finally, Kuhlmann et al. [40] studied the impact of training and test dataset selection on graph neural networks for predicting acceptability of arguments (sceptical acceptance under the preferred semantics in this case). They studied the AGNN [31] and FM2 [10] models and found that AGNN in particular is capable of predicting acceptability on challenging (even adversarial) instances with a degree of accuracy if a suitable training set is used. The training set used with AGNN by Craandijk and Bex [31] (consisting of random frameworks with $5-25$ arguments) was found to be insufficiently diverse to produce good performance on ICCMA competition frameworks, a finding that is consistent with our evaluation of AGNN in Section 5.3. In general, Kuhlmann et al.'s findings support our use of a diverse training set sampled from the ICCMA competition frameworks.

4. AFGCN: a GCN-based approximate solver for abstract argumentation

In this section we present our new approximate solver architecture, AFGCN. The solver architecture can be trained for each of the seven semantics described above, and for credulous or sceptical acceptability. Given an argumentation framework, the trained solver predicts whether each argument in the framework is acceptable under the chosen semantics.

4.1. Neural network architecture

The architecture used in this article, which we refer to by the moniker AFGCN, builds on the seminal approach introduced by Kipf and Welling [12], but extends it in a number of areas. In the original formulation, the GCN consisted of an input layer, two hidden layers with ReLU (Rectified Linear Unit, i.e. $f(x) = \max(0, x)$) nonlinearities inserted in between, and ending with an output layer. Node embeddings were generated using a propagation rule following a first-order approximation of spectral graph convolutions. We follow the same basic pattern, but add a number of features to allow for greater depth and to tailor the approach to abstract argumentation graphs that do not intrinsically have node-level features.

The core GCN architecture has been extended using several techniques to improve its performance and generalization ability. One such extension is a randomized training regime that involves two main components: (1) randomly shuffling the order of the argumentation frameworks used for training at each epoch, and (2) randomly selecting a subset of arguments within each argumentation framework to be used for training. This randomized selection of arguments is performed continuously throughout the training process. By introducing this randomness in the training data, the model is encouraged to learn more robust and generalisable representations of the argumentation frameworks, rather than overfitting to specific frameworks or arguments.

The core components of AFGCN architecture are:

1. Randomly generated input features combined with input features generated from the grounded extension of the argumentation framework.
2. An input layer receiving these inputs along with the normalized adjacency matrix.
3. 4 repeating blocks of a GCN layer [12] and a Dropout layer [41].
4. Residual connections feeding the original features and the normalised adjacency matrix as additional input to each block [42].
5. A Sigmoid output layer generating an estimate for the acceptability of each argument on a continuous [0..1] scale.

The model was trained using Adam [43] with Binary Cross-Entropy as the loss function. The learning rate was set to $1e^{-3}$ for two hours and then dropped to $1e^{-6}$ for an additional six hours of training. These rates were identified by manual inspection of the training process. Details on the training regime are described in subsequent sections. All hyperparameters were manually tuned.

4.1.1. Deep residual connections

The original formulation of Graph Convolutional Networks suffers from major performance degradation with an increase of depth beyond a certain limit. Kipf and Wellings' [12] original GCN, for instance, used only two layers in the model. In practice, as the depth of the GCN increases beyond this limit the model stops responding to training data and instead converges to a fix-point. This problem is known as the suspended animation problem [42] and the limit as the suspended animation limit.

Several approaches have been applied to overcome this limit and allow greater depth in GCN architectures. Among the most fruitful approaches have been those that adapt the notion of residual connections to the GCN context [42] by feeding in the graph structure and node features across layers in a variety of ways.

In this article, we follow a similar approach by adapting the graph-raw residual defined by Zhang and Meng [42]. They define the residual term as the multiplication of the normalised adjacency matrix and the raw input features. This residual term is fed as input to each layer in the model, which achieves the aim of extending the suspended animation limit.

The only difference in our approach is that the normalised adjacency matrix and raw input features are fed to each layer separately rather than as a unit, largely for reasons related to the implementation approach.

Definition 4.1 *(Deep residuals).* By deep residuals, we mean layer-wise terms, R, that are added to the hidden state at each layer according to the following equation.

$$R(H^{(l-1)}, X; G) = \hat{A}X$$

4.2. Input features

The input features can be divided into standard input features and the extended features generated by grounded reasoning.

4.2.1. Standard input features

The first important input feature is the adjacency matrix of the argumentation graph. This is generated dynamically from the input file and preprocessed in accordance with the normalization in Definition 2.19. That is to say the identity matrix is added to the adjacency matrix.

To preserve the directionality of the attack relationships, only incoming links are included in the adjacency matrix. These are normalised using the function $\hat{A}\hat{D}^{-1}$. The normalisation is done dynamically on initialization of the training function.

In addition there are 64 random features initialized using Xavier initialization [44], which help provide numerical stability in the initial training. Xavier initialization, proposed by Glorot and Bengio, is a widely used initialization scheme for neural networks that aims to keep the scale of the gradients roughly the same across all layers. This is achieved by setting the initial weights of the network to random values drawn from a specific distribution, which helps prevent the gradients from vanishing or exploding during training.

4.2.2. Grounded reasoning as an input feature

We incorporate grounded reasoning into AFGCN using input features that correspond to the binary status of whether an argument is included in the grounded extension or not. The grounded extension is a unique, universally accepted set of arguments that can be computed in polynomial time using a grounded solver. A grounded solver is an algorithm that iteratively identifies and labels the arguments in an argumentation framework that are not attacked by any other argument, and then removes those arguments and their associated attacks from the framework. This process continues until no more arguments can be labelled, and the resulting set of labelled arguments constitutes the grounded extension.

While the grounded extension is a specific type of extension, it can provide valuable information for approximating other semantics as well. This is because the grounded extension represents a set of uncontroversial arguments that are likely to be included in many other extensions. By using the grounded extension as an input feature, the GCN can learn to prioritize these arguments and potentially improve its accuracy in predicting extensions under different semantics.

To incorporate grounded reasoning, we use a binary vector representation of the grounded extension, where each element corresponds to an argument in the argumentation framework. An element is set to 1 if the corresponding argument is included in the grounded extension, and 0 otherwise. This binary vector is then used as an additional input feature alongside the adjacency matrix and other input features in the GCN.

We explore two ways of incorporating grounded reasoning into the AFGCN model:

1. Using the grounded extension as an input feature during both training and inference.
2. Using the grounded extension only during inference, where the grounded solver's output is directly used for arguments that are part of the grounded extension, and the GCN is used to predict the status of the remaining arguments.

These approaches allow us to investigate the effectiveness of incorporating grounded reasoning into the GCN model and its impact on approximating different semantics.

Fig. 5. Overall process for data processing and training of the AFGCN solver.

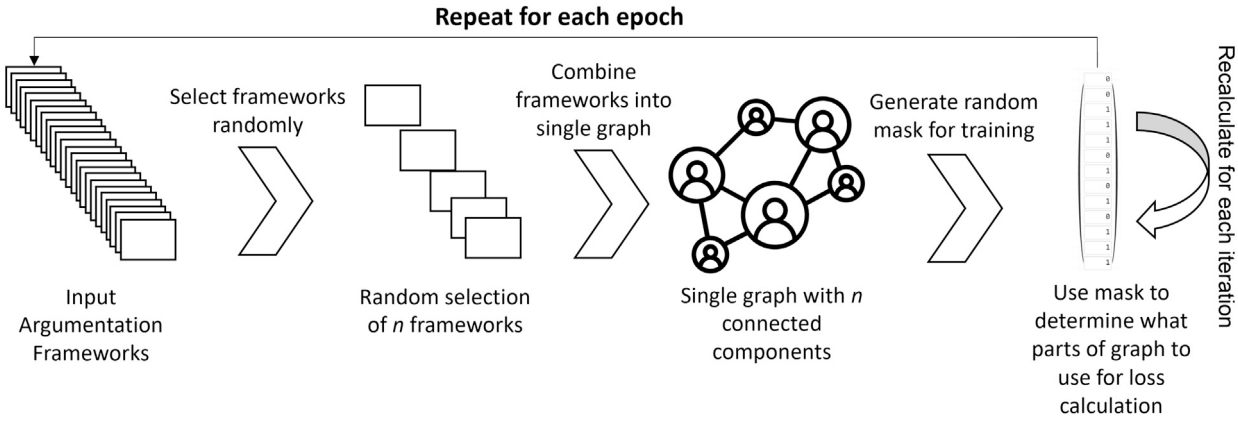


Fig. 6. Overview of the process for generating random training batches.

4.3. Training regime

The overall process for generating data and training the model for use in the solver can be seen in Fig. 5. In the following sections, we will cover the relevant details.

4.3.1. Randomised training batches

Real-world abstract argumentation frameworks tend to have a skewed distribution between acceptable and non-acceptable arguments both for credulous and sceptical acceptance. In particular, there tends to be a large preponderance of non-acceptable arguments. In the argumentation frameworks used for the experiments in this article, the percentage of non-acceptable arguments ranges from 69.5% to 99.95% across semantics and reasoning tasks with the main differentiator being credulous vs. sceptical acceptability.

Unbalanced training data affects GCN training as the neural net will by default learn to predict a negative outcome even in cases where it is incorrect. This problem was also noted by Kuhlmann and Thimm [10], who generated balanced training data to attempt to address it, but this approach does not seem to have generalised well.

To overcome these limitations, we have devised a randomised training scheme that generates random training batches at the start of each training iteration. These are generated by sampling the total dataset and selecting n argumentation frameworks at random with uniform probability. See Fig. 6 for an overview of batch generation.

The overall training scheme feeds multiple argumentation frameworks to the neural network as a single graph that forms the union of the vertices and edges of all the component graphs. That enables effective batch processing of multiple argumentation frameworks that can be treated as a single graph for learning purposes. This processing happens dynamically in the training function.

When performing inference, the output layer of AFGCN produces a prediction of the acceptability of every argument represented in the input graph. For training, we generate a mask for each training iteration, essentially asking the neural net to fill in the blanks. This mask is a binary vector with a length equal to the node count of all the graphs in the training batch combined. The value in the binary vector indicates whether the prediction in the given spot is included in the loss calculation used for network learning.

Randomised masking is performed to encourage the network to learn to generalise based on structural properties of the graphs. By continuously randomising both the set of AFs and the set of arguments in the mask, the ability of the neural network to generalise to unseen graphs improves.

4.3.2. Dynamic balancing and outlier exclusion

Two additional measures were taken to address the problems related to imbalanced training data and poor generalisation performance. First, the training mask was developed to have the option of dynamically balancing the training mask to adjust the balance acceptable and non-acceptable arguments.

That is accomplished by programmatically adjusting the mask during training so the target vector contains similar amounts of acceptable and non-acceptable arguments. The algorithm to implement this simply replaces a given number of arguments of one class with arguments of another class in order to achieve the balance.

This has the intention to avoid the skew caused by unbalanced training data, but also has the unfortunate side effect of reducing the amount of data used for training. This mode is therefore not used in all experiments described in the results section.

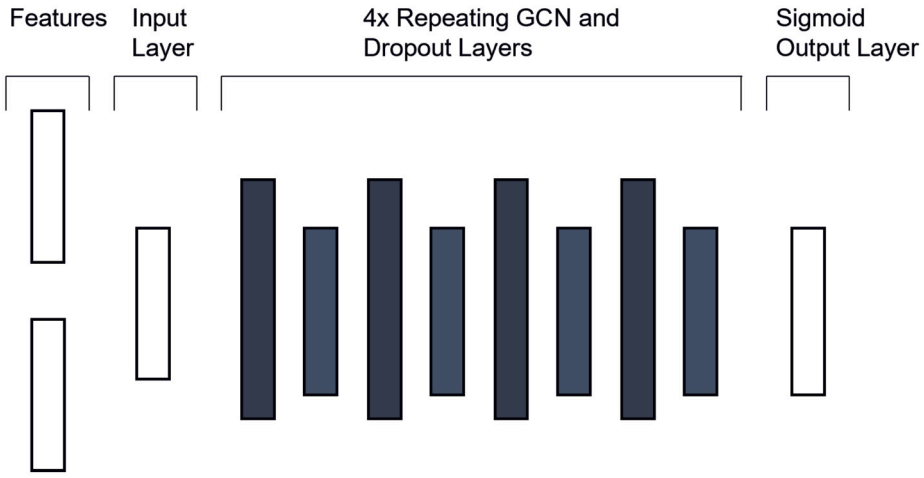


Fig. 7. Network architecture for the solver used in the results section for AFGCN. Overall training is done using the ADAM optimizer with Binary Cross-Entropy loss as the loss function. Training was done using a set of 792 argumentation frameworks from past ICCMA competitions, using cross-validation and a holdout set of 99 frameworks for testing. One model was trained for each problem in the ICCMA competition.

A second enhancement was added to handle extremely skewed argumentation frameworks. Some frameworks have no or almost no acceptable arguments and when included tend to skew the training disproportionately. These frameworks have been excluded from the training set using a z-score test with a threshold of 3.5 [45].

4.4. Runtime implementation

The model chosen for the final solver runtime is a 4-layer model, not including the input and output layers, with 128 hidden features per GCN layer. The solver has been built using the Python programming language, utilising the Pytorch framework for training and modelling [46], the Deep Graph Library for graph representation [47], and Numpy [48] for numerical computation.

At runtime the solver is called using a shell script wrapper that conforms to the specifications of ICCMA 2021. This shell script calls a Python script that loads the relevant parameters into the GCN model based on the semantics in question. It then pre-computes the grounded extension using a Numpy-based grounded solver and passes this information along with random input features to the GCN model for inference.

The output of the inference step is then passed to a threshold function, which applies a threshold for acceptance that is adapted to the size of the argumentation framework and the semantics under consideration. The solver approximates the acceptability status of all arguments in the argumentation framework in parallel during the inference step, using a single step of the GCN, but to conform with the ICCMA 2021 solver specification it only outputs the predicted status for the particular argument under consideration.

5. Experimental results

5.1. AFGCN results

We will start by presenting the experimental setup for the article and then review the results first by semantics and then in a cross-cutting way. An overview of the AFGCN network architecture can be found in Fig. 7.

5.1.1. Dataset and experimental setup

We train our models on a dataset of 792 graphs taken from past ICCMA competitions. The ICCMA competitions provide a comprehensive set of benchmark problems that provide good comparability to historical results and therefore provides a good source for dataset creation. The graphs range in size from 2 to 100,000 arguments. The test set consists of 99 graphs constructed to contain a fairly even split of graphs between the benchmarks present at the ICCMA 19 competition. Table 3 contains a description of the benchmarks under consideration.

Table 4 contains the characteristics of the test set relative to the benchmarks defined above. Max, mean, and min refers to the number of arguments.

This division into benchmarks allows us to evaluate the relative performance of the solver on different graph structures. The proportion of arguments within the grounded extension exhibits substantial variation across benchmarks, ranging from 0% for AFGen, Erdős–Rényi, and Watts-Strogatz to 57.71% for Barabasi-Albert. This heterogeneity provides valuable insight into the potential efficacy of grounded reasoning across diverse graph structures. For each benchmark, we computed the percentage of arguments in the grounded extension, providing insight into the potential impact of grounded reasoning on different graph types.

Table 3
Description of benchmarks used for the AFGCN solver evaluation.

Benchmark	Description
ABA2AF	Assumption-Based Argumentation translated to abstract argumentation frameworks
AFGen	Based on a general model for producing random digraphs with differing properties
Barabasi-Albert	Barabasi-Albert graphs, randomly generated
Erdős–Rényi	Erdős-Rényi graphs, randomly generated
Grounded	Randomly generated argumentation frameworks containing only a large grounded extension
Logic Based Argumentation	Argumentation graphs based on knowledge bases
Planning2AF	Planning problems transformed to abstract argumentation problems
Stable	Graphs generated to have a high number of stable extensions
Traffic	Traffic networks converted to abstract argumentation frameworks
Watts-Strogatz	Watts-Strogatz graphs, randomly generated
admbuster	admbuster graphs, based on Caminada and Podlaszewski [49], designed to foil certain types of solvers

Table 4
Characteristics of the test set used for AFGCN evaluation.

Benchmark	Number	Max	Mean	Min	% Grounded
ABA2AF	10	848.0	611.7	443.0	1.44%
AFGen	10	320.0	189.6	100.0	0.00%
Barabasi-Albert	10	201.0	111.0	21.0	57.71%
Erdős–Rényi	3	102.0	101.7	101.0	0.00%
Grounded	10	8020.0	3942.7	1697.0	11.32%
LBA	10	103.0	58.0	6.0	0.90%
Planning2AF	10	1992.0	627.4	86.0	18.44%
Stable	10	767.0	562.7	265.0	1.73%
Traffic	10	35.0	21.1	7.0	8.06%
Watts-Strogatz	6	300.0	266.7	200.0	0.00%
admbuster	10	10000.0	7000.0	4000.0	50.00%

We trained three different models for comparison using a variety of features described in the previous sections. In addition, we also used a deterministic grounded solver as a fourth option. For convenience, we also refer to this as a model in our results presentation, although strictly speaking it does not rely on any machine learning components.

The four models are characterised below, the abbreviation after each model denotes how it is referred to in the results tables:

- **GR-ONLY (GR)**. This model uses only the deterministic grounded solver.
- **GCN-NO-GR (NO-GR)**. This model uses a 4-layer GCN model using the randomised training regime, input feature initialisation, thresholding, and residual connections, but no grounded features.
- **GCN-WITH-GR (W/GR)**. This model uses everything discussed under the GCN-NO-GR model, but also takes the grounded extension as an input feature both during training and inference.
- **HYBRID-GCN-GR (HYBR)**. This model uses everything included in the GCN-WITH-GR model. However, it does not train on elements of the grounded extension. Instead, it incorporates a grounded solver during the inference stage and always trusts a positive answer from that solver. For negative cases, it applies the neural network for inference as in the previously discussed model.

We evaluate our models in four different ways for all semantics under consideration. This allows us to see whether there are any systematic differences in approximation performance across semantics. Due to space considerations only evaluations for the preferred, complete, and stable semantics are included in the main text, excepting cross-cutting analyses. Detailed results for other semantics can be found in the appendix.

The evaluation settings are summarised in the following list:

- **Equally weighted**. The equally weighted setting weighs each argumentation framework equally regardless of its size. This is equivalent to the score one would expect when picking a single argument to classify from each of a number of different argumentation frameworks as for instance in the ICCMA competitions.
- **Complete balanced**. This setting classifies all arguments across all argumentation frameworks and gives weight according to the size of the framework. So performance on a 1,000 argument framework is weighted 10 times as highly as on a 100 argument one. This setting is included as it is the one which has been used in previous work on approximating argumentation frameworks [11,10].
- **Reduced balanced**. This setting is equivalent to complete balanced, but excludes the benchmarks Grounded and admbuster. As can be seen from Table 4, these two benchmarks dominate in terms of size and also share the characteristic of being fully solvable using only grounded reasoning. Therefore, including these on an equal basis makes the results hard to interpret. The reduced balanced setting corrects for this problem.

- **By benchmark**. This setting compares performance across the benchmarks targeted for evaluation.

In the appendix, we report a number of different evaluation metrics for the sake of completeness. But in the evaluation, we rely mostly on accuracy, which has been the key metric in past research, and Matthews Correlation Coefficient (MCC), which gives the best view of an estimator's overall performance taking into consideration all classes and class imbalances. Note that in this context, "positive" refers to an argument being predicted as acceptable, while "negative" refers to an argument being predicted as not acceptable.

The metrics are defined as follows:

$$
\begin{aligned}
&TP = True\ Positive\\
&FP = False\ Positive\\
&TN = True\ Negative\\
&FN = False\ Negative\\
&Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\\
&Precision = \frac{TP}{TP+FP}\\
&Recall = \frac{TP}{TP+FN}\\
&F1 = \frac{2\times Precision\times Recall}{Precision+Recall} = \frac{2\times TP}{2\times TP+FP+FN}\\
&MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}
\end{aligned}
$$

The models were tuned using the hyperparameters for the original AFGCN model submitted to ICCMA 21. As the purpose here has been to evaluate systematically, we have not sought to wring out every last little bit of performance from the various models. We tuned the configurable thresholds on the complete training set, assigning an optimal set of thresholds based on this data.

In the following sections, we will note considerable variability across the different semantics that will be evaluated as part of this article. In the following sections, we will systematically go through the results for each of the included semantics before proceeding to the additional cross-cutting analyses.

5.1.2. Results for credulous acceptance

Preferred semantics We will first consider the results of running the experiment on credulous acceptability. We refer readers to results tables that can be found in the appendix for detailed presentation of the metrics. We provide an overview by including summary diagrams that show the headline results and facilitates digesting the main data points.

Fig. 8 shows the results of running our model against the test set for DC-PR. Overall the best performing model under equal weighting is the one combining GCN and grounded reasoning in a hybrid mode, although the difference in performance to the GCN-WITH-GR model, incorporating grounded reasoning only through input features is minimal. For positive accuracy and precision the grounded reasoner achieves a perfect score, which is unsurprising considering that all preferred extensions are also complete extensions and the grounded extension is a subset of any complete extension.

It is perhaps surprising that the HYBRID-GCN-GR model does not achieve a higher boost in positive accuracy by applying grounded reasoning. While the configurable threshold helps performance here, it may not be sufficient to compensate for the tendency towards false positives exhibited by all the neural network models. However, all of the performance boost seen relative to the GCN-WITHOUT-GR model that does not incorporate grounded reasoning does come from an increase in positive accuracy as the simplified GCN model actually performs slightly better on negative accuracy.

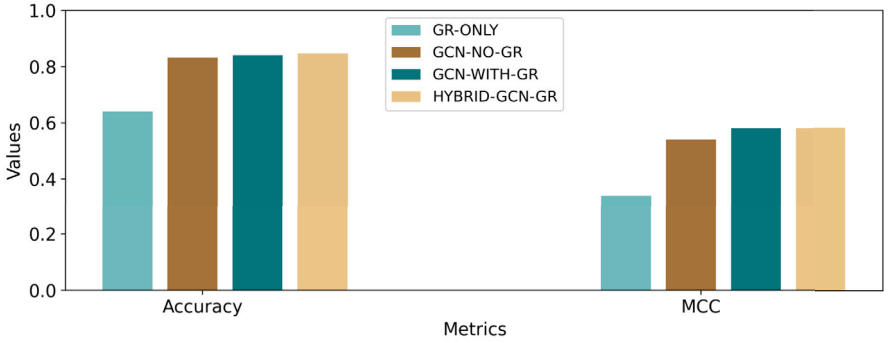
The F1 and MCC scores both indicate that all the GCN models are strong positive predictors for credulous acceptability under the preferred semantics. The GR-ONLY model exhibits only a moderate positive relationship as a predictor in comparison.

Moving on to the results for the complete balanced setting, we can see that all models have very strong performance, which is attributable to the dominance of grounded reasoning in this evaluation setting. Unsurprisingly, the best performing model both in terms of accuracy and MCC is HYBRID-GCN-GR.

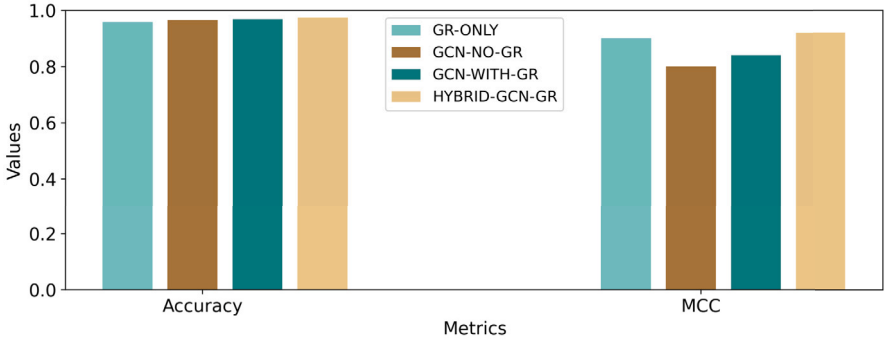
On the other hand in the reduced balanced setting, excluding the two large grounded-focused benchmarks, we see marginally better performance from the GC-WITH-GR model, largely attributable to this model having slightly better performing thresholds for targeting benchmarks that do not rely exclusively on grounded reasoning. It is also worth noting that there is poor recall performance of the GCN-NO-GR model in this evaluation setting, which indicates that the threshold configuration has not been enough on its own to achieve a low rate of false negatives.

If we turn to the benchmark results in Table 5, there is a marked difference in performance between benchmark types. All models achieve comparable results on the ABA2AF framework, but the model without grounded features achieved a much lower MCC indicating a weaker balance in the results. For the AFGen benchmark, no model is strong enough to have real predictive power according to MCC, although the grounded GCN variants perform slightly better. This is, however, unlikely to be genuinely significant given the fact that AFGen graphs have an empty grounded extension.

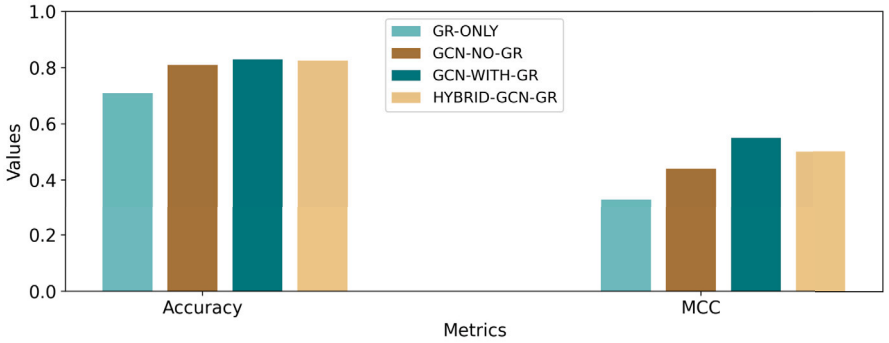
The three GCN models have equivalent correlation scores for Barabasi-Albert (BA) graphs, but the GCN-NO-GR model has highest accuracy. The substantial proportion (57.71%) of arguments within the grounded extension for Barabasi-Albert graphs complicates the minimal performance improvement observed when incorporating grounded reasoning. This suggests that the GCN architecture may already be implicitly capturing much of the information provided by the grounded extension.



(a) Equal Weighting



(b) Complete Balanced



(c) Reduced Balanced

Fig. 8. AFGCN Results for the DC-PR decision problem.

ER graphs exhibit identical accuracy for all three models that apply grounded reasoning, outperforming the GCN-NO-GR model. This is an interesting phenomenon worthy of a separate analysis, but outside the scope of the current article. As expected the GR-ONLY model performs best on the Grounded benchmark with the GCN-NO-GR model performing the worst. This also shows the hybrid approach to work better as a way of incorporating grounded reasoning than just features on tasks that are slanted heavily towards this mode.

All the GCN models achieve perfect scores on the Logic Based Argumentation benchmark, outperforming the grounded reasoner. The fairly well-structured information derived from knowledge base information would seem to be a good fit for approximation with GCNs.

On the Planning2AF benchmark, the hybrid model achieves slightly higher accuracy than and equal MCC to the GCN-WITH-GR model, outperforming the other two, again showing the value of combining GCNs with grounded reasoning. The hybrid model also outperforms slightly on the stable benchmark, although none of the models do particularly well.

Traffic network data is another area, where grounded reasoning does not seem to play a part and in fact seems harmful, given the much stronger performance of the GCN-NO-GR model. The relatively low proportion (8.06%) of arguments in the grounded

Table 5
Overview of AFGCN approximation results for DC-PR ordered by benchmark. Abbreviations refer to models defined above (GR:Grounded-Only, HYBR:Hybrid-GCN-GR, NO-GR:GCN-NO-GR, W/GR:GCN-WITH-GR). Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	98.53%	99.07%	98.97%	98.96%	0.53	0.73	0.71	0.70
AFGen	58.23%	55.36%	54.60%	58.16%	0.094	0.13	-0.10	0.11
Barabasi-Albert	91.55%	86.52%	49.93%	90.70%	0.72	0.71	0.31	0.73
Erdős–Rényi	63.03%	96.04%	96.04%	96.04%	0.32	0.67	0.67	0.67
Grounded	97.94%	98.48%	100.00%	98.98%	0.69	0.77	1.0	0.94
LBA	100.00%	100.00%	0.79%	100.00%	1.0	1.0	-0.40	1.0
Planning2AF	65.18%	74.91%	59.66%	76.45%	0.34	0.52	0.40	0.52
Stable	65.59%	67.24%	62.53%	68.03%	0.27	0.30	0.20	0.32
Traffic	81.40%	75.67%	32.95%	73.19%	0.51	0.31	0.029	0.27
Watts-Strogatz	75.97%	75.92%	75.25%	75.25%	0.10	0.16	0.0	0.0
admbuster	99.75%	99.57%	100.00%	100.00%	0.99	0.99	1.0	1.0

Table 6
Overview of AFGCN approximation results for DC-CO ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	98.74%	99.10%	99.03%	99.21%	0.64	0.77	0.77	0.82
AFGen	54.66%	65.93%	54.60%	62.43%	-0.099	0.068	-0.10	0.058
Barabasi-Albert	93.88%	95.61%	49.93%	95.91%	0.78	0.84	0.31	0.85
Erdős–Rényi	91.75%	88.80%	96.04%	81.90%	0.75	0.061	0.67	0.10
Grounded	97.95%	98.51%	100.00%	99.78%	0.69	0.79	1.0	0.98
LBA	67.76%	95.73%	0.00%	96.11%	0.30	0.30	-1.0	0.30
Planning2AF	72.48%	83.09%	59.66%	77.34%	0.36	0.64	0.40	0.54
Stable	67.49%	71.36%	62.04%	69.30%	0.34	0.43	0.19	0.40
Traffic	81.04%	89.79%	32.95%	85.59%	0.27	0.72	0.029	0.64
Watts-Strogatz	73.42%	76.64%	75.25%	78.14%	0.13	0.27	0.0	0.21
admbuster	99.38%	99.67%	100.00%	100.00%	0.99	0.99	1.0	1.0

extension for Traffic benchmarks may account for the limited impact of grounded reasoning on model performance in this domain. Watts-Strogatz graphs as with AFGen previously seems basically unapproximable using either a GCN or grounded reasoning approach. Finally, we can note that none of the models are fooled by the admbuster benchmark.

Complete semantics The results for the complete semantics have many similarities with those for the preferred semantics, which is unsurprising as all preferred extensions are also complete extensions. There are, however, a number of salient differences that we shall point out as we go through.

Starting with the equally weighted results in Fig. 9, we see the GCN-WITH-GR model edge ahead of the HYBRID-GCN-GR model on accuracy, while maintaining equal MCC. As expected for credulous reasoning the grounded reasoner does relatively poorly.

This changes, however, when we get to the complete balanced setting, dominated by the large Grounded benchmarks. As DS-CO is equal to the grounded extension, we would expect the GR-ONLY model to have near perfect performance in this case. Here it is matched by the HYBRID-GCN-GR model that even slightly outperforms it by having a better class balance between positive and negative cases.

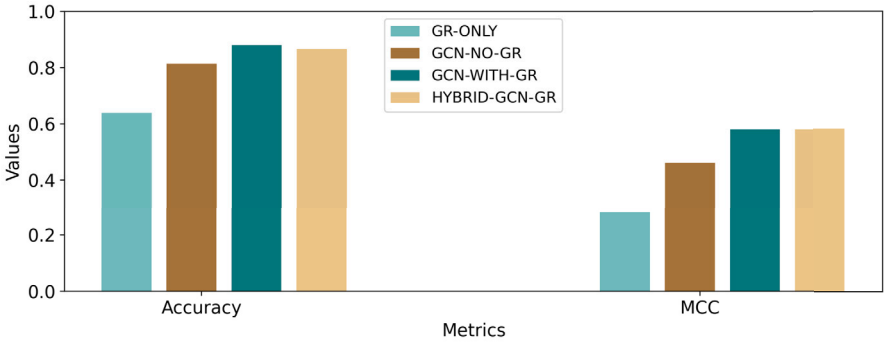
Removing these large grounded-focused frameworks, the overall pattern seen in the equally weighted setting reasserts itself.

From Table 6 we can see that ABA2AF, AFGen, Grounded, Planning2AF, Stable and admbuster benchmarks have effectively the same behaviour seen in the preferred case albeit with some variation in performance. However, the other five benchmarks do not follow the same pattern.

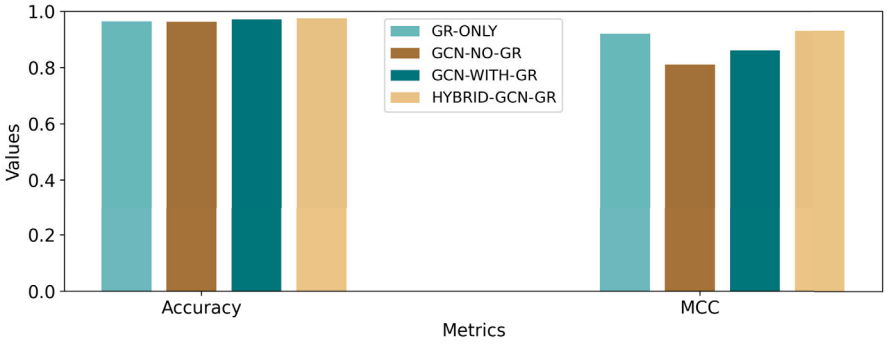
Barabasi-Albert (57.71% grounded), ER (0% grounded), and Traffic (8.06% grounded) benchmarks show improved performance with grounded reasoning under complete semantics, despite their varying grounded extension proportions. This suggests that the impact of grounded reasoning differs across semantics, even for the same graph structures.

The LBA benchmark that was perfectly predictable by the GCN models under preferred semantics is only partially predictable under complete semantics and interestingly has a perfect negative correlation with grounded reasoning, which is explained by the grounded extension being empty and all arguments being credulously accepted under complete semantics.[1] Watts-Strogatz graphs are slightly more predictable by a GCN under complete than preferred semantics, although still only weakly so.

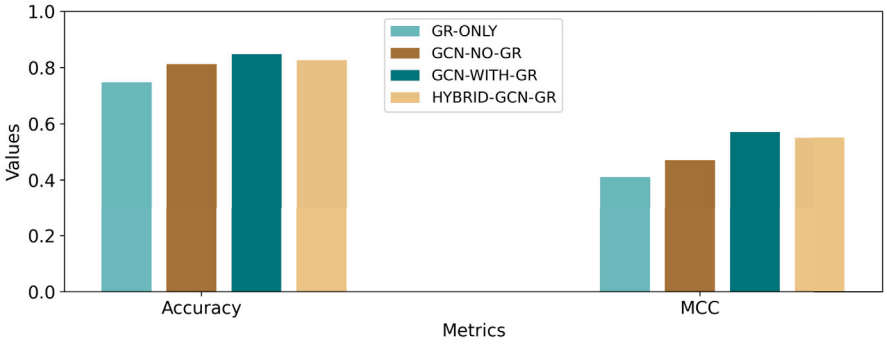
[1] We thank a reviewer for making this observation.



(a) Equal Weighting



(b) Complete Balanced



(c) Reduced Balanced

Fig. 9. AFGCN Results for the DC-CO decision problem.

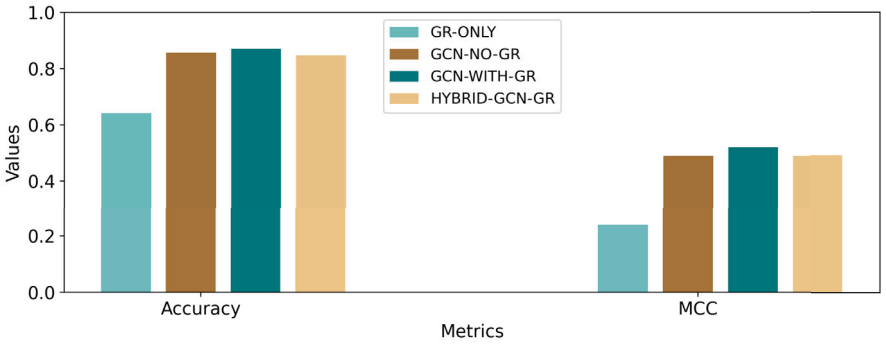
Stable semantics The results for stable semantics have much in common with the ones we have just seen for complete semantics, more so than it shares with preferred semantics, which is somewhat strange, considering that every stable extension is a preferred extension.

In the equally weighted setting, shown in Fig. 10.(a) again the GCN-WITH-GR model is the overall best performing model. Interestingly, the HYBRID-GCN-GR model and the GCN-NO-GR model have near identical performance, which might indicate that there is less space for improving performance with grounded reasoning under stable semantics.

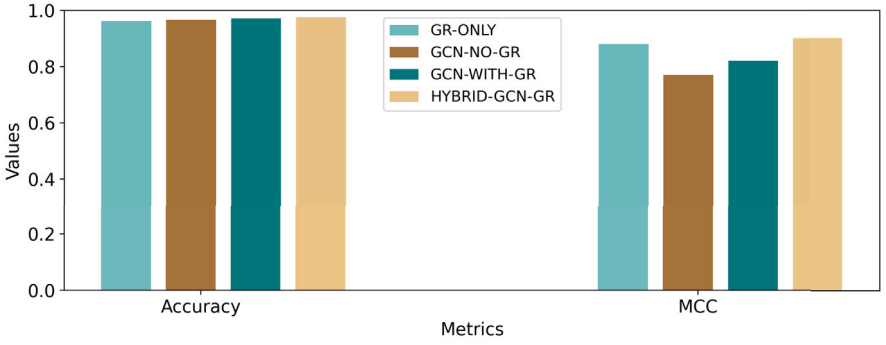
In the complete balanced setting, we see the same pattern as under complete semantics, where the HYBRID-GCN-GR model performs the best, due to the dominance of the large grounded frameworks.

However, when excluding these frameworks the trend reverses again. This is consistent with the other findings we have seen so far.

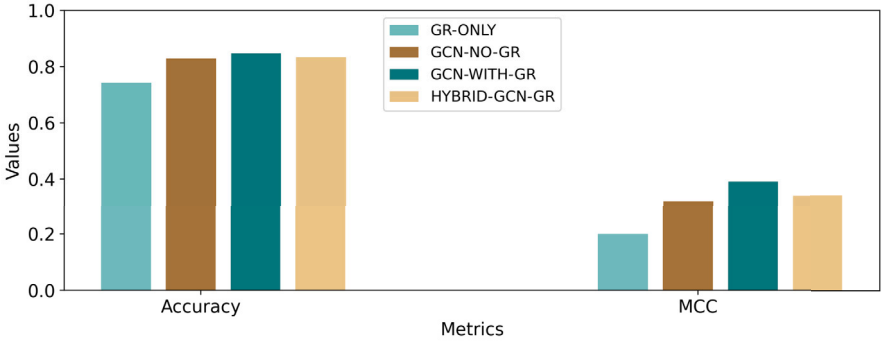
Again, it is in the benchmark specific performance, shown in Table 7, we find the most interesting variation due to the semantics. For AFGen, Barabasi-Albert, Watts-Strogatz, Grounded, admbuster, and Stable benchmarks, the pattern is the same as we saw under complete semantics. One might have expected that a model trained on stable extension would perform better on the Stable benchmark,



(a) Equal Weighting



(b) Complete Balanced



(c) Reduced Balanced

Fig. 10. AFGCN Results for the DC-ST decision problem.

but this is not reflected in the data. This would seem to indicate that the GCN has not learned any semantics specific representations for these semantics. The approximation of ER graphs is for some reason easier with the GCN-WITH-GR model under stable semantics than under complete semantics where this model completely failed. The GCN-NO-GR model is still the best performing model here, which was true for complete, but not preferred semantics. Approximation of LBA frameworks is somewhere in the middle between those of complete and preferred semantics. Planning2AF problems have slightly better results under stable semantics across the board than under complete or preferred semantics, excepting a slight drop for the GCN-WITH-GR model relative to complete semantics, the same is true for the Traffic benchmark.

Summary for credulous acceptability The results for credulous acceptability show that the best performing model under equal weighting is the HYBRID-GCN-GR model, which combines GCN and grounded reasoning in a hybrid mode. This model outperforms the GCN-WITH-GR model, which incorporates grounded reasoning only through input features, but the difference in performance is minimal. For positive accuracy and precision, the grounded reasoner achieves a perfect score, as expected.

Table 7
Overview of AFGCN approximation results for DC-ST ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	98.97%	98.90%	98.53%	98.57%	0.049	0.12	0.070	0.085
AFGen	54.53%	54.83%	54.60%	59.70%	-0.087	0.0016	-0.10	0.045
Barabasi-Albert	94.24%	92.81%	49.93%	82.25%	0.78	0.77	0.31	0.61
Erdős–Rényi	95.38%	93.73%	96.04%	80.93%	0.78	0.73	0.67	0.12
Grounded	97.75%	98.50%	100.00%	99.97%	0.64	0.77	1.0	1.0
LBA	97.05%	97.81%	0.35%	97.81%	0.70	0.50	-0.70	0.50
Planning2AF	80.76%	84.55%	62.06%	84.73%	0.58	0.67	0.42	0.68
Stable	68.35%	72.36%	64.49%	66.55%	0.32	0.43	0.23	0.27
Traffic	81.49%	88.58%	31.52%	80.18%	0.58	0.56	-0.071	0.47
Watts-Strogatz	76.78%	76.14%	75.25%	73.56%	0.15	0.15	0.0	0.23
admbuster	99.79%	99.73%	100.00%	100.00%	1.0	0.99	1.0	1.0

In the complete balanced setting, all models have very strong performance, with the HYBRID-GCN-GR model being the best performing model both in terms of accuracy and MCC. In the reduced balanced setting, the GC-WITH-GR model performs marginally better, mainly due to its slightly better thresholds for targeting benchmarks that do not rely exclusively on grounded reasoning.

When looking at the results by benchmark type, there is a marked difference in performance. All models achieve comparable results on the ABA2AF framework, but the model without grounded features achieves a much lower MCC. For the AFGen benchmark, no model is strong enough to have real predictive power according to MCC, although the grounded GCN variants perform slightly better. The three GCN models have equivalent correlation scores for BA graphs, but the GCN-NO-GR model has the highest accuracy, indicating that grounded reasoning is not an important factor for this graph type.

Overall, the MCC scores indicate that all the GCN models are strong positive predictors for credulous acceptability under the preferred and complete semantics. The GR-ONLY model exhibits only a moderate positive relationship as a predictor in comparison.

The results for credulous acceptability show that incorporating grounded reasoning into the model can improve its performance, especially in cases where the problem relies heavily on grounded reasoning. The hybrid approach seems to be more effective in incorporating grounded reasoning than just using features, and the choice of evaluation setting can have a significant impact on the model's performance.

5.1.3. Results for sceptical acceptance

Preferred semantics Now we turn attention to sceptical acceptance under the preferred semantics.

We start again with the equally weighted setting (refer to Fig. 11). On an equally weighted basis results are slightly better overall than for credulous acceptance. The GR-ONLY model is the second best performing on an MCC basis, which is on expectation. The best performing model is the GCN-WITH-GR model, which is somewhat surprising given the better performance we saw from the hybrid model on the Grounded benchmark for credulous acceptance. However, the GCN-WITH-GR model would seem to have better ability to generalise sceptical acceptance across benchmarks leading to the overall higher score.

Looking instead at the complete balanced setting, the GR-ONLY model is the overall winner followed by GCN-WITH-GR both in terms of accuracy and MCC, largely reflecting its superior performance on the Grounded benchmark. The hybrid model does particularly poorly in this evaluation, which reflects an overoptimism in the configured thresholds leading to low precision.

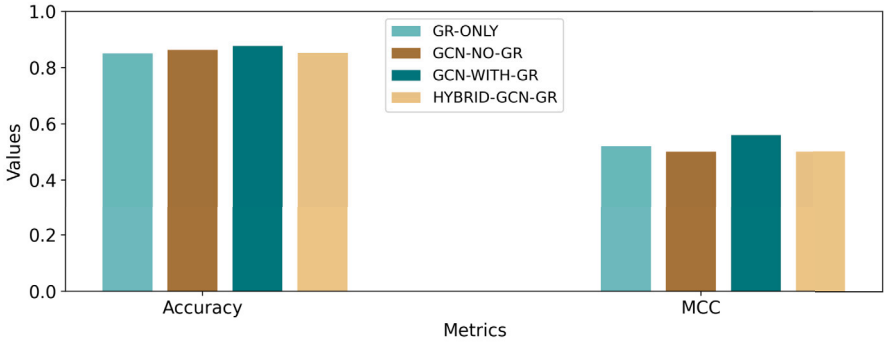
Excluding the Grounded and admbuster benchmarks, the GCN-WITH-GR model edges ahead of pure grounded reasoning in both accuracy and MCC, reflecting this model's better ability to generalise across benchmarks. The GCN-NO-GR model performs the worst under this setting, although the HYBRID-GCN-GR model is still underperforming due to low precision.

Considering the benchmark evaluation in Table 8, the ABA2AF benchmark shows a common phenomenon when dealing with sceptical acceptance, which is very high accuracy, but significantly lower MCC, which is due to a large imbalance in favour of negative cases in the data. This is demonstrated perfectly by the AFGen benchmark, where 93.96% accuracy is revealed to have no predictive power by the MCC score. Watts-Strogatz graphs reveal similar behaviour, but less strongly. This is consistent with neither of these models having any arguments in their grounded extensions.

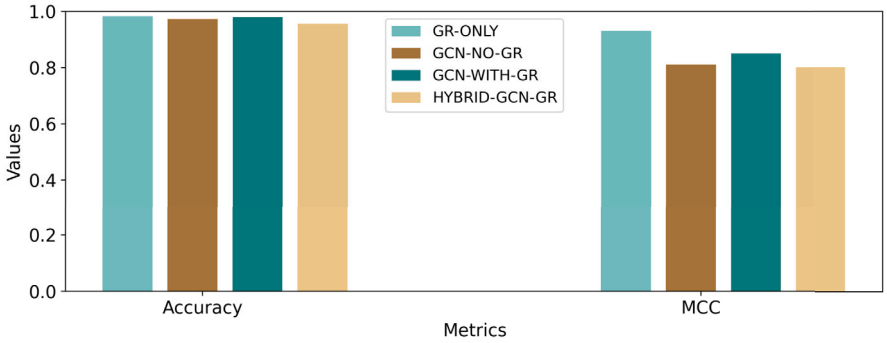
For Barabasi-Albert graphs and ER graphs all models perform effectively on par, excepting a small dip for the hybrid model on BA graphs. Grounded graphs are solved perfectly by the grounded model, but less well by the various GCN model. On the other hand the grounded model does not have any predictive power for LBA graphs, which is consistent with low proportion of the grounded extension while the scores for the GCN models are much lower than in the credulous setting.

Planning2AF graph performance is on par between the GR-ONLY and the GCN-WITH-GR models, suggesting that in this case the GCN simply applies grounded reasoning via the approximation. For the Stable benchmark, the GCN-WITH-GR model has best performance. In contrast, it has worst performance on the Traffic benchmark, while other models are approximately on par. Neither of these phenomena are readily explainable. Admbuster graphs are once again solved near-perfectly in the sceptical setting.

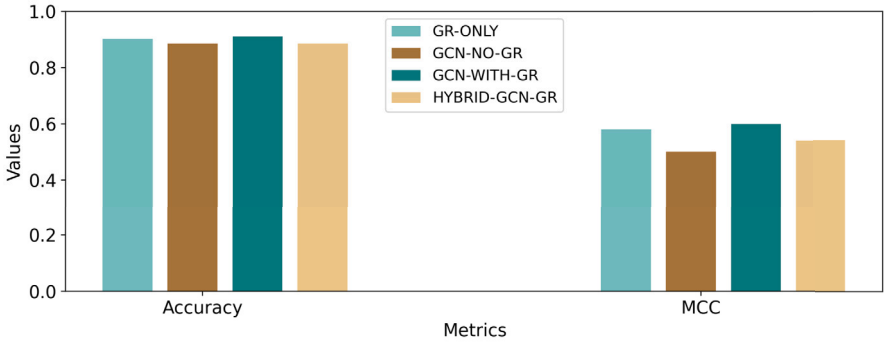
Complete semantics The sceptical setting for the complete semantics is equal to the grounded semantics and computable in polynomial time. One would therefore never in practice need to approximate this task. However, for the sake of completeness, we will still run through the results.



(a) Equal Weigthing



(b) Complete Balanced



(c) Reduced Balanced

Fig. 11. AFGCN Results for the DS-PR decision problem.

The equally weighted setting shown in Fig. 12, shows perfect performance for the GR-ONLY model with near-perfect performance for the HYBRID-GCN-GR model, indicating that it is using a .99 threshold for all size bands and therefore almost always use a grounded reasoner to answer. The other GCN models do not reach the same level of performance, indicating that they have not learnt pure grounded reasoning.

The picture is identical for the complete balanced setting, although the underperformance of the GCN models is less marked.

The reduced balanced setting is somewhere in between the two other evaluation settings, but shows the same overall pattern.

The benchmark level view in Table 9 shows us where the difficulties are in approximation. The hardest to approximate frameworks in this setting are the Planning2AF and Traffic benchmarks, accounting for most of the reduced performance in the GCN models. It would be worth a separate investigation to see why these are hard to approximate.

Stable semantics Moving on to stable semantics, we see a drop in performance for the GR-ONLY model, relative to the other semantics we have examined when considering the equally weighted performance in Fig. 13. In contrast, the three GCN models are within the

Table 8
Overview of AFGCN approximation results for DS-PR ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	99.21%	99.55%	99.52%	99.52%	0.68	0.81	0.84	0.84
AFGen	93.96%	93.96%	93.96%	93.96%	0.0	0.0	0.0	0.0
Barabasi-Albert	84.22%	84.98%	82.86%	81.08%	0.69	0.71	0.71	0.61
Erdős–Rényi	96.04%	96.04%	96.04%	96.04%	0.67	0.67	0.67	0.67
Grounded	98.04%	98.64%	100.00%	92.37%	0.73	0.79	1.0	0.68
LBA	68.72%	81.40%	50.50%	68.92%	0.40	0.62	0.045	0.43
Planning2AF	81.76%	88.29%	88.20%	84.99%	0.56	0.72	0.72	0.65
Stable	79.27%	81.36%	79.12%	75.68%	0.29	0.40	0.27	0.21
Traffic	70.77%	61.35%	69.26%	68.50%	0.36	0.23	0.36	0.34
Watts-Strogatz	82.08%	82.64%	82.03%	81.69%	0.020	0.15	0.0	0.011
admbuster	99.73%	99.92%	100.00%	100.00%	0.99	1.0	1.0	1.0

Table 9
Overview of AFGCN approximation results for DS-CO ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	99.76%	99.83%	100.00%	100.00%	0.84	0.85	1.0	1.0
AFGen	100.00%	100.00%	100.00%	100.00%	1.0	1.0	1.0	1.0
Barabasi-Albert	97.29%	97.25%	100.00%	99.76%	0.95	0.94	1.0	1.0
Erdős–Rényi	100.00%	100.00%	100.00%	100.00%	1.0	1.0	1.0	1.0
Grounded	98.49%	98.28%	100.00%	99.99%	0.78	0.75	1.0	1.0
LBA	99.97%	100.00%	100.00%	100.00%	0.90	1.0	1.0	1.0
Planning2AF	90.75%	92.07%	100.00%	99.86%	0.71	0.75	1.0	1.0
Stable	99.93%	100.00%	100.00%	100.00%	0.97	1.0	1.0	1.0
Traffic	84.20%	88.38%	100.00%	96.79%	0.20	0.53	1.0	0.94
Watts-Strogatz	100.00%	100.00%	100.00%	100.00%	1.0	1.0	1.0	1.0
admbuster	99.94%	99.83%	100.00%	100.00%	1.0	1.0	1.0	1.0

Table 10
Overview of AFGCN approximation results for DS-ST ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

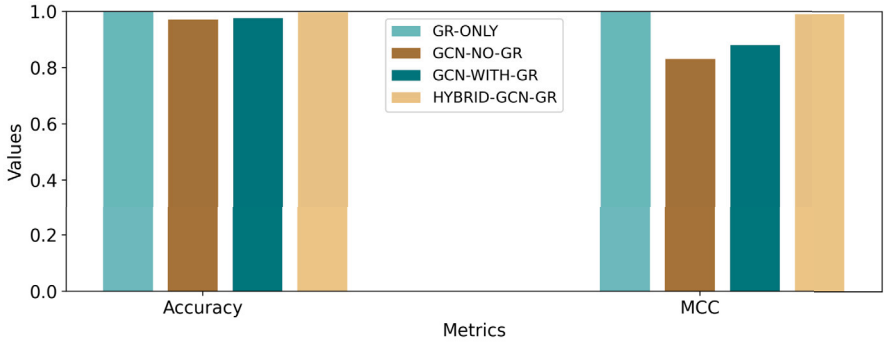
Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	99.46%	99.38%	98.80%	98.80%	0.035	0.050	0.050	0.050
AFGen	94.04%	93.37%	94.04%	94.04%	0.0	0.0061	0.0	0.0
Barabasi-Albert	83.11%	81.41%	68.60%	84.45%	0.62	0.58	0.53	0.66
Erdős–Rényi	96.04%	92.75%	96.04%	96.04%	0.67	0.41	0.67	0.67
Grounded	98.11%	97.73%	100.00%	99.98%	0.71	0.65	1.0	1.0
LBA	98.12%	99.38%	10.25%	98.75%	0.80	0.90	0.0055	0.80
Planning2AF	86.51%	83.26%	78.43%	84.29%	0.71	0.64	0.57	0.67
Stable	82.08%	81.80%	81.56%	81.55%	0.30	0.29	0.27	0.27
Traffic	58.53%	61.25%	63.56%	55.72%	0.21	0.16	0.24	0.10
Watts-Strogatz	81.92%	82.00%	81.92%	81.92%	0.0	0.031	0.0	0.0
admbuster	99.74%	99.52%	100.00%	100.00%	0.99	0.99	1.0	1.0

same performance envelope, once again reinforcing the view that grounded reasoning does not add as much to an approximation attempt under stable semantics as it does under other semantics we have seen.

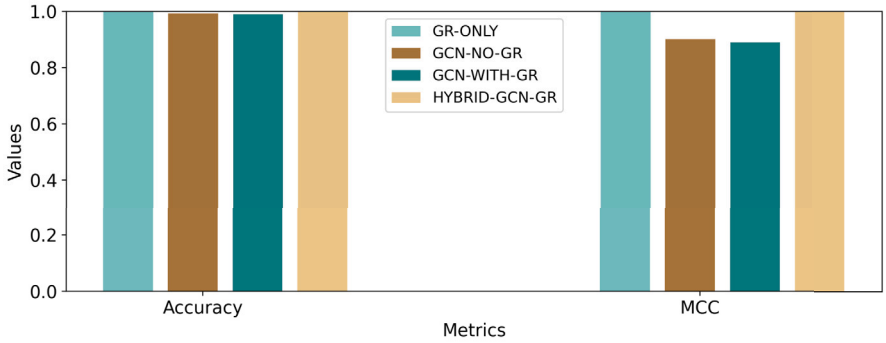
The complete balanced setting, as with past cases, accentuates the performance of the pure grounded elements in the GR-ONLY and HYBRID-GCN-GR models. In contrast, the GCN-WITH-GR model becomes the worst performing in this setting, because it has learned to generalise more across benchmarks at the cost of underperforming on the Grounded one.

In the reduced setting, the trend again reverses and the three GCN-based models are once again within the same performance envelope. The GCN-NO-GR model is marginally ahead as it was for the equally weighted setting, but not enough to be noteworthy.

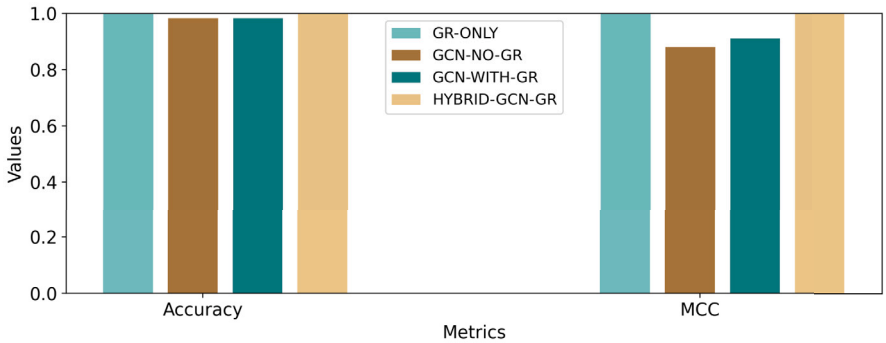
The benchmark specific analysis for the stable semantics can be seen in Table 10. Comparing to the results for sceptical acceptance under the preferred semantics, we find rough equivalence of results for AFGen, ER, Grounded, Planning2AF, Stable, Watt-Strogatz, and admbuster. As was the case with credulous acceptance, the ABA2AF frameworks are not approximable under stable semantics. There are notable performance drops for Barabasi-Albert and Traffic benchmarks and a large increase in performance for LBA frameworks.



(a) Equal Weighting



(b) Complete Balanced



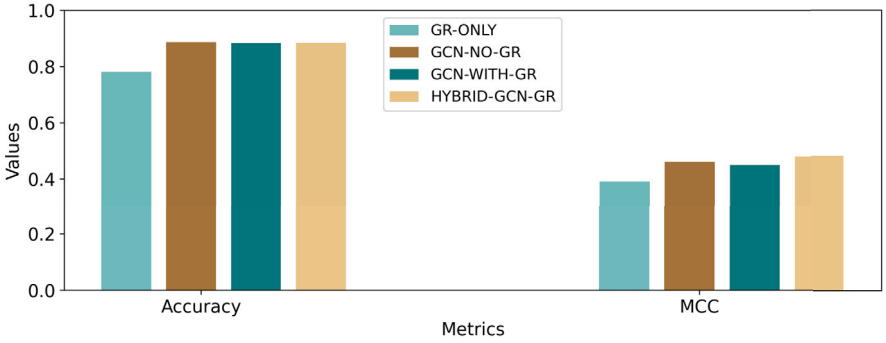
(c) Reduced Balanced

Fig. 12. AFGCN Results for the DS-CO decision problem.

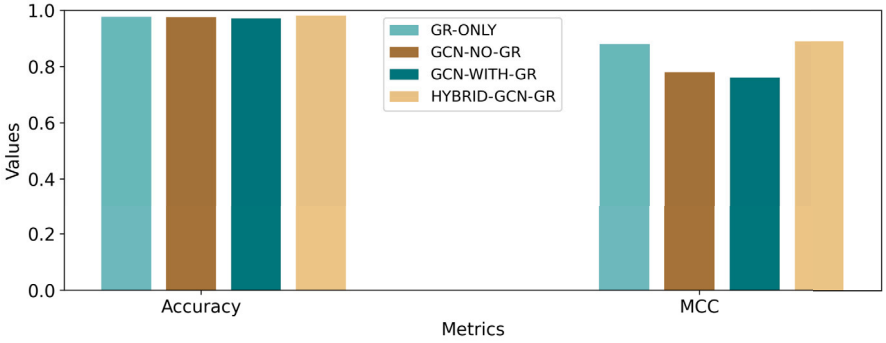
Summary of sceptical results The results for sceptical acceptance show some interesting patterns. For the preferred and complete semantics, the best performing model is generally the GCN-WITH-GR model, which combines GCN with grounded reasoning features. This model is able to generalize well across benchmarks and show strong performance overall. However, it is worth noting that the HYBRID-GCN-GR model, which incorporates grounded reasoning in a hybrid mode, also performs well, particularly in the complete balanced setting where it is dominated by the large Grounded benchmarks.

For stable semantics, the performance of the GR-ONLY model drops, indicating that grounded reasoning is less relevant for this semantics. In this case, the three GCN models are within the same performance envelope, with the GCN-NO-GR model being marginally ahead in some settings. This suggests that grounded reasoning does not add as much to an approximation attempt under stable semantics compared to other semantics.

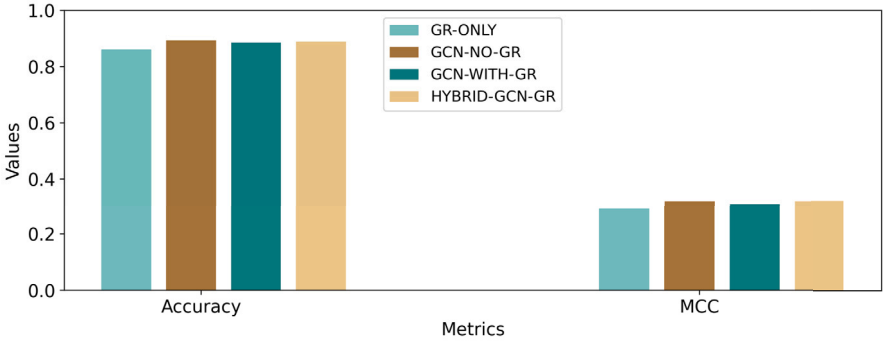
Overall, the results demonstrate that combining GCN with grounded reasoning can lead to improved performance in approximating argumentation semantics, particularly for preferred and complete semantics.



(a) Equal Weighting



(b) Complete Balanced



(c) Reduced Balanced

Fig. 13. AFGCN Results for the DS-ST decision problem.

5.1.4. Cross-cutting results

In this section, we will look at the results across the various parameters that we have used for our analysis. This includes analyses across semantics, benchmarks, and sizes. We do this in order to show any general results that are not specific to individual semantics. We will start by presenting the cross-cutting analysis based on semantics.

By semantics Table 11 shows accuracy and MCC results aggregated for each semantics. It is difficult to clearly identify significant differences in the ease with which semantics can be approximated using these models. Excluding the DS-CO task, which can be calculated in polynomial time, the spread between approximation performance in both accuracy and MCC terms is low. Overall, considering both factors, semi-stable semantics would seem to be the easiest to approximate and stable semantics the hardest. But the difference is not large enough to make a substantial point.

Identifying a best performing model is also difficult. The GR-ONLY model wins the DS-CO task, but for the other tasks, it is very close between the HYBRID-GCN-GR model and the GCN-WITH-GR model. For DC-CO and DC-PR, they are close enough in performance to be indistinguishable. For the DC-SST, DS-ID, DS-SST, and DS-ST tasks, the HYBRID-GCN-GR models outperform the

Table 11
Overview of AFGCN approximation results compared across semantics using equally weighted setting. Accuracy to 2 decimal places, MCC to 2 significant figures.

Semantics	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
DC-PR	83.10%	83.96%	63.98%	84.69%	0.54	0.58	0.34	0.58
DC-CO	81.31%	88.02%	63.86%	86.58%	0.46	0.58	0.28	0.58
DC-ST	85.62%	87.06%	64.19%	84.66%	0.49	0.52	0.24	0.49
DC-SST	77.04%	86.00%	64.41%	86.64%	0.42	0.55	0.32	0.60
DC-STG	84.05%	86.84%	61.45%	85.76%	0.46	0.57	0.23	0.54
DS-PR	86.24%	87.66%	84.99%	85.14%	0.50	0.56	0.52	0.50
DS-CO	97.00%	97.54%	100.00%	99.64%	0.83	0.88	1.0	0.99
DS-ST	88.65%	88.29%	78.10%	88.44%	0.46	0.45	0.39	0.48
DS-SST	86.75%	86.94%	85.51%	86.63%	0.53	0.51	0.52	0.55
DS-STG	87.48%	88.81%	85.90%	87.86%	0.48	0.55	0.48	0.52
DS-ID	86.16%	87.28%	85.33%	87.44%	0.52	0.53	0.52	0.57

Table 12
Overview of AFGCN approximation results compared across benchmarks for all semantics using equally weighted setting. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	98.81%	99.07%	98.96%	99.04%	0.49	0.59	0.63	0.64
AFGen	75.63%	76.51%	74.89%	77.20%	0.083	0.13	0.050	0.12
Barabasi-Albert	89.24%	89.69%	67.60%	89.32%	0.74	0.75	0.53	0.75
Erdős–Rényi	89.90%	94.79%	89.64%	90.87%	0.60	0.57	0.50	0.52
Grounded	97.93%	98.27%	100.00%	99.06%	0.68	0.74	1.0	0.95
LBA	79.86%	93.14%	27.56%	88.94%	0.53	0.65	-0.24	0.53
Planning2AF	80.66%	84.04%	74.83%	85.00%	0.57	0.64	0.58	0.68
Stable	75.77%	77.77%	73.58%	76.28%	0.37	0.43	0.31	0.38
Traffic	75.64%	76.74%	52.32%	76.64%	0.37	0.43	0.22	0.45
Watts-Strogatz	80.86%	81.53%	80.41%	80.87%	0.19	0.26	0.10	0.19
admbuster	99.63%	99.72%	100.00%	100.00%	0.99	0.99	1.0	1.0

GCN-WITH-GR model. For DC-ST, DC-STG, DS-PR, and DS-STG tasks it is the other way around. That means that we cannot give a clear answer to whether it is better to incorporate grounded reasoning only using features fed to the neural network or whether there is a benefit hybridizing GCN models with a grounded solver.

We can, however, note that incorporating grounded reasoning into a GCN model using either mechanism results in increased performance relative to a model that does not. The performance boost is modest, but consistent across semantics, considering both variants that include grounded reasoning.

By benchmark The benchmark specific results, shown in Table 12 are more varied than was the case for the analysis based on semantics. We can start with the admbuster benchmark, which is designed to foil certain types of solvers and note that it does not manage to do so for any of the models under consideration here.

The Grounded benchmark is a major factor in the evaluation, due to the large size of the frameworks and their focus on grounded reasoning. Here we see that the GR-ONLY model has its expected perfect performance, followed closely by the HYBRID-GCN-GR model for the simple reason that it will default to grounded reasoning in the majority of cases. The difference in performance to the GCN-WITH-GR model indicates that these two models have substantially different ways of incorporating grounded reasoning.

Four other benchmarks have a large component of grounded reasoning: ABA2AF, ER, Planning2AF, and Barabasi-Albert. Interestingly, for both ER and Barabasi-Albert graphs, the better performance for the GCN-models comes from the GCN part of the equation, despite the importance of grounded reasoning that can be seen from the performance of the GR-ONLY model. The reverse seems to be the case for ABA2AF. For Planning2AF there is a small boost from combining both GCN and grounded reasoning.

ER and LBA graphs both varied substantially in the performance seen across the semantics. In aggregate, they end up being fairly approximable, which is obviously misleading given the specific results we have seen.

Stable and Traffic benchmarks have similar and rather middling performance in aggregate, which is the result of the small but significant variations that were seen across semantics for these benchmarks.

AFGen and Watts-Strogatz graphs are some of the most consistent benchmarks in the set, given that they are weakly approximable or unapproximable by these models across semantics.

The heterogeneity in grounded extension proportions across benchmarks underscores the significance of incorporating grounded reasoning. Benchmarks with higher proportions of grounded arguments, such as Barabasi-Albert (57.71%), provide a robust signal for the model.

Table 13
Ablation study results for the AFGCN model on the credulous (DC-PR) and sceptical (DS-PR) acceptability problems under the preferred semantics.

Model	Semantics	Accuracy (%)	MCC
4-Layer AFGCN	DC-PR	95.1	0.610
5-Layer AFGCN	DC-PR	94.9	0.601
6-Layer AFGCN	DC-PR	93.2	0.398
4-Layer AFGCN (no training optimization)	DC-PR	92.2	0.327
4-Layer AFGCN	DS-PR	97.5	0.720
5-Layer AFGCN	DS-PR	97.4	0.704
6-Layer AFGCN	DS-PR	97.4	0.704
4-Layer AFGCN (no training optimization)	DS-PR	94.9	0.291

However, it is noteworthy that even benchmarks with null grounded extensions (e.g., AFGen, Erdős–Rényi) benefit in some cases from models having incorporated grounded reasoning, suggesting that this process contributes valuable structural information beyond mere argument inclusion.

We also in a number of cases examined the correlation between the proportion of arguments in the grounded extension and the performance improvement gained by incorporating grounded reasoning. This analysis revealed that while higher proportions generally led to greater improvements, the relationship was not strictly linear, suggesting that other structural properties of the graphs also play a significant role.

5.2. Ablation studies

To investigate the impact of various architectural choices and training strategies on the performance of the AFGCN model, a series of ablation studies were conducted. These studies aimed to examine the effects of network depth, class balancing, and training optimization techniques on the model's accuracy and Matthews Correlation Coefficient (MCC) when solving the credulous and sceptical acceptability problems under the preferred semantics. The results of these ablation studies are presented in Table 13.

5.2.1. Effects of depth

The impact of network depth on the AFGCN model's performance was examined by training models with 4, 5, and 6 layers. For the credulous acceptability problem under the preferred semantics (DC-PR), the 4-layer AFGCN achieved the highest accuracy (95.1%) and MCC (0.61), followed closely by the 5-layer model. The 6-layer model exhibited a notable decrease in performance, with an accuracy of 93.2% and an MCC of 0.398.

For the sceptical acceptability problem under the preferred semantics (DS-PR), the 4-layer AFGCN also outperformed the deeper models, achieving an accuracy of 97.5% and an MCC of 0.72. The 5-layer and 6-layer models both had slightly lower accuracy (97.4%) and MCC (0.704) scores.

These results suggest that increasing the depth of the AFGCN model beyond 4 layers does not necessarily lead to improved performance and may even be detrimental in some cases. This finding is consistent with the challenges faced by deep GCNs, such as the over-smoothing problem [50,51] and vanishing gradient issue [52].

5.2.2. Effects of training optimization

The impact of training optimization techniques on the AFGCN model's performance was assessed by comparing the 4-layer AFGCN model with and without the training optimization strategies.

For both the DC-PR and DS-PR problems, the model trained with optimization techniques significantly outperformed the model trained without them. In the DC-PR setting, the optimized model achieved an accuracy of 95.1% and an MCC of 0.61, while the non-optimized model had an accuracy of 92.2% and an MCC of 0.327. Similarly, for the DS-PR problem, the optimized model attained an accuracy of 97.5% and an MCC of 0.72, compared to the non-optimized model's accuracy of 94.9% and MCC of 0.291.

These results highlight the importance of employing appropriate training optimization strategies, such as the randomized training regime, to improve the AFGCN model's performance on the acceptability problems.

5.2.3. Summary of ablation studies

The ablation studies presented in this section provide insights into the factors influencing the performance of the AFGCN model on the credulous and sceptical acceptability problems under the preferred semantics. Key findings include:

- Increasing network depth beyond 4 layers does not consistently improve performance and may lead to a decline in accuracy and MCC scores.
- Employing appropriate training optimization strategies, such as the randomized training regime, is important for achieving strong performance on the acceptability problems.

Table 14
Overview of AFGCN runtime results, key statistics.

	Runtime with GR (ms)	Runtime without GR (ms)
min	6.83	6.12
25%	12.44	10.55
50%	28.96	20.72
75%	810.58	242.72
max	21563.85	4922.45

Table 15
PYGLAF runtime to determine a single argument. All times are in milliseconds, to 6 significant figures.

Group	Mean (ms)	Median (ms)	Min (ms)	Max (ms)
DC-CO	123490	51294.7	122.480	594880
DC-PR	123646	50806.5	137.729	595089
DC-SST	190652	64992.1	137.916	600172
DC-ST	147624	68084.8	151.857	588594
DC-STG	470557	599532	135.897	601137
DS-CO	92184.7	46223.1	108.567	570255
DS-ID	490946	595811	170.028	601034
DS-PR	239657	104572	114.881	600682
DS-SST	191973	66349.2	145.543	600861
DS-ST	192526	93378.4	137.009	589832
DS-STG	460623	599432	152.218	601131

These results emphasize the importance of carefully designing the architecture and training process for AFGCN models to effectively solve the credulous and sceptical acceptability problems in argumentation frameworks. Further research should focus on refining these techniques and exploring additional strategies to enhance the model's performance and robustness.

5.2.4. Runtime performance

Runtime performance is one of the major reasons one might consider using an approximate approach to solving abstract argumentation problems. Here we consider the runtime in a variety of contexts focusing on a comparison that either includes or excludes the time it takes to compute the grounded extension for an argumentation framework.

The runtime statistics shown in Table 14 gives the results breakdown for classifying an entire argumentation framework with all its arguments. This table shows that while most frameworks can be fully classified in less than a second including the overheads needed to initialise the model, this can increase substantially for the worst case. It also shows that the cost of computing the grounded extension increases disproportionately with scale. This is as expected as the algorithm to compute the grounded extension has polynomial runtime in the number of arguments.

5.2.5. Runtime comparison with exact methods

In this analysis, we compare the runtime performance of PYGLAF [53], an exact method that won the preferred track at ICCMA 2021, and AFGCN. The results presented are derived from the actual data obtained during the ICCMA 2021 competition, evaluated on the relevant argumentation semantics.

Table 15 shows the runtime performance of the PYGLAF method for single argument evaluation across various argumentation semantics groups. The results are capped at 600 seconds as per the competition rules. This implies that if the actual runtimes exceeded 600 seconds, the real comparison would be even more striking.

Tables 16 and 17 present the runtime performance of the AFGCN method in two different scenarios: 'all arguments' and 'time per argument'. The 'all arguments' method is the right comparison mode when the answer is required for a single argument, while the 'time per argument' method is more appropriate when the answer is needed across all arguments in an argumentation framework. It should be noted that the 'per argument' method is not directly comparable to the 'single argument' case for PYGLAF since enumeration of extensions would be used rather than running the solver for each argument. However, the comparison can still provide an indicative sense of the speedup when answers are needed across a framework.

Focusing on the mean runtimes, we observe that AFGCN consistently outperforms PYGLAF in terms of speed. In the 'all arguments' case, AFGCN is up to 122.8 times faster than PYGLAF when comparing the mean runtimes, despite classifying all arguments in the framework rather than just one (e.g., DS-ID group: 3988.83 ms for AFGCN vs. 490946 ms for PYGLAF). The lowest speedup in this case can be seen in the DS-CO group, where AFGCN is 2.86 times faster than PYGLAF (e.g., 32225.2 ms for AFGCN vs. 92184.7 ms for PYGLAF).

The speedup is even more pronounced when comparing the 'time per argument' case, with AFGCN being theoretically up to 149,116 times faster than PYGLAF (e.g., DS-ID group: 3.29238 ms for AFGCN vs. 490,946 ms for PYGLAF) if one were to run the solver once for each argument, although as noted this wouldn't be the practical implementation.

Other metrics, such as median, minimum, and maximum runtimes, also support the observation that AFGCN is significantly faster than PYGLAF.

Table 16
AFGCN runtime for all arguments in a framework. The columns show the time taken to classify all arguments in an argumentation framework. All times are in milliseconds (ms).

Group	Mean	Median	Min	Max
DC-CO	29014.5	18456.3	976.1	63070.5
DC-PR	29294.5	19194.8	1012.3	62154.1
DC-SST	27876.7	17168.7	1039.0	65651.4
DC-ST	28333.2	18064.9	972.0	67038.8
DC-STG	8952.2	7669.2	980.9	36944.3
DS-CO	32225.2	31585.8	995.3	62023.9
DS-ID	3988.8	3260.6	982.8	15123.2
DS-PR	10107.5	8449.7	965.2	33380.9
DS-SST	9120.6	7917.3	1024.8	42996.7
DS-ST	21555.2	15009.3	1015.2	62530.0
DS-STG	8938.6	7894.1	975.2	39858.8

Table 17
AFGCN runtime per argument. All times are in milliseconds (ms), to 6 significant figures.

Group	Mean	Median	Min	Max
DC-CO	0.511720	0.325508	0.0172153	1.11236
DC-PR	0.517241	0.338915	0.0178741	1.09743
DC-SST	0.493667	0.304039	0.0183997	1.16261
DC-ST	0.521155	0.332283	0.0178783	1.23310
DC-STG	0.992253	0.850040	0.108718	4.09485
DS-CO	0.450129	0.441197	0.0139025	0.866363
DS-ID	3.29238	2.69133	0.811174	12.4827
DS-PR	0.922482	0.771186	0.0880943	3.04659
DS-SST	1.01557	0.881593	0.114106	4.78765
DS-ST	0.689990	0.480453	0.0324956	2.00161
DS-STG	0.971628	0.858082	0.106003	4.33264

The runtime analysis comparison highlights the superior speed of the AFGCN method compared to the exact PYGLAF method in the context of abstract argumentation. This finding is particularly relevant when evaluating multiple arguments in argumentation frameworks or when time constraints are crucial.

5.3. Comparison to related work

The two most important pieces of related work are Kuhlmann and Thimm [10], and Craandijk and Bex [31]. Kuhlmann and Thimm trained a graph convolutional neural network (named FM2) to predict credulous acceptability with respect to the preferred semantics (DC-PR). They evaluated two versions of their model using a sample of 45 graphs from the ICCMA 2017 competition. The better version of their model obtained a total accuracy of 63%. For comparison, AFGCN was evaluated with 99 graphs also drawn from the ICCMA 2017 competition, and achieved a total accuracy of 97% (Complete Balanced setting) for both the GCN-WITH-GR and HYBRID-GCN-GR versions of AFGCN. While this is not an exact comparison, it indicates that there is a significant performance difference. Kuhlmann et al. [40] found that the AGNN model defined by Craandijk and Bex [31] achieved higher accuracy and MCC than FM2 for every combination of training and test set used in their experiments, so our main focus is on comparing AFGCN to AGNN.

Craandijk and Bex defined the argumentation graph neural network (AGNN), as described in Section 3. They report very high accuracy for random argumentation frameworks with up to 200 arguments. In this section we compare the accuracy of the AGNN model (when trained exactly as described by Craandijk and Bex, and using their software distribution[2]) to the AFGCN with three semantics and with both credulous and sceptical acceptance tasks. The number of message passing layers of the AGNN model was fixed to 32 in this experiment, and similarly the AFGCN models both have a fixed number of layers.

In this experiment, both the AGNN and AFGCN are evaluated using our test set of 99 argumentation frameworks sampled from the ICCMA 2019 competition (as described in Table 4). We use MCC as used by Craandijk and Bex, and each argument is given equal importance regardless of the size of the framework (i.e. the Complete Balanced setting). A training set of 1 million random argumentation frameworks was generated with sizes ranging from 5 to 25 arguments, along with a validation set of 1000 frameworks of size 25. For each task, the AGNN was trained for 200 epochs. The trained model with the highest MCC on the validation set was retained in each case. Training for each task took approximately 50 hours on an NVIDIA Quadro P6000 GPU.

[2] Available from https://github.com/DennisCraandijk/DL-abstract-argumentation.

Table 18
MCC of AGNN on the validation set, and on the test set of 99 competition argumentation frameworks. For comparison the rightmost columns are the MCC on the test set for two versions of the AFGCN model.

Semantics	AGNN MCC		AFGCN MCC	
	Validation	Test	GCN-WITH-GR	HYBRID-GCN-GR
DC-PR	1.00	0.41	0.85	**0.93**
DC-CO	1.00	0.36	0.86	**0.93**
DC-ST	1.00	0.36	0.82	**0.90**
DS-PR	1.00	0.28	**0.85**	0.80
DS-CO	1.00	0.23	0.89	**1.00**
DS-ST	1.00	-0.20	0.76	**0.89**

Table 18 contains the MCC results for AGNN, and for two versions of the AFGCN. Training of the AGNN was successful, as shown by the extremely high MCC achieved for the validation set. However, AFGCN achieves substantially higher MCC values for every task on the test set. It is worth noting that most of the graphs in our test set are substantially larger than those in the validation set, and also drawn from a variety of different distributions, most of which are not represented in the AGNN training set. Given the shift in both size and distribution of graphs, it is not surprising that the AGNN model performs less well than the AFGCN models.

6. Summary

This article has presented systematic results from applying deep learning based approximation approaches to key problems in abstract argumentation. First, we can note that, in general, argumentation frameworks adhering to a variety of schemes can be approximated moderately well to very well by an approach that combines grounded reasoning with graph neural networks. This is true of both credulous and sceptical acceptance and across semantics.

There are cases that prove unapproximable or very hard to approximate and require further analysis such as the unapproximability of ER graphs under some but not all semantics and the general low approximability of Watts-Strogatz graphs.

However, the one benchmark that is generally unapproximable, AFGen, is likely to be for the reason that it is a random graph model with very little structure in its generating function. This means it does not contain enough regularity for a neural network to learn anything.

It's worth noting that ER and Watts-Strogatz are also random graph models with different generating functions, so in general we can suggest that approximation for random graph models is problematic with our chosen approaches. However, differences in how the random graph model is generated do seem to matter in terms of learnability.

We can also conclude that while a GCN-based approach on its own is a good approximator, it is a better approximator when combined with grounded reasoning, although we cannot definitively conclude which is the best way to combine grounded reasoning with GCN-based approaches on the basis of these results.

A grounded reasoner is a good, but not perfect approximator for sceptical acceptance across semantics and in general the improvement made by adding a GCN model is small for sceptical acceptance. On the basis of these results, one might be tempted to conclude that it is not worth bothering with additional approximation approaches for sceptical acceptance unless one is dealing with problems of a scale where a marginal improvement is worth a substantial investment. The problem with this position, however, is that the grounded reasoner is fixed. It will never provide a better approximation than it already does, which is still no better than 80% accuracy on an equally weighted basis in most cases. While the current approach only improves marginally on this by adding a GCN, it at least shows it is possible to improve on this baseline. Further research may lead to greater improvements still.

Perhaps the most promising line of enquiry coming out of this research can be found by considering the considerable difference in performance found across benchmarks and semantics. For some benchmarks in some semantics, such as LBA frameworks under preferred semantics, the approximation performance is near perfect. This begs the question, whether the goal of creating a general purpose approximator for abstract argumentation is actually a foolish one and whether the more profitable approach might be to create task specific approximators depending on the problem at hand.

Referring back to our contributions, we have confirmed that it is possible to create a high-performing approximation approach for abstract argumentation using GNNs. We developed a unique training approach, using a modified GCN architecture that works effectively in this context. We also discussed in detail the effects of bringing in the grounded extension as a starting point and demonstrated that it can provide a boost in approximation performance in many cases. Finally, we systematically evaluated differences in performance across semantics, and benchmarks.

In our future work, we aim to address the limitations and scope for improvement in the accuracy of our current models, particularly for certain poorly performing benchmark types. We believe there are several avenues to achieve better performance, including the development of argumentation-specific graph embeddings, leveraging more data, utilizing targeted data, and employing data augmentation techniques. Additionally, we plan to explore the potential of incorporating advanced architectural elements such as Deep Reinforcement Learning. Although our initial experiments with this approach did not generalize well [54], we believe that further research could yield promising results.

Furthermore, we intend to extend our approach beyond the basic formalism of abstract argumentation, as it is not inherently restricted to this specific formalism. We plan to adapt our models to accommodate other similar formalisms, such as bipolar argu-

mentation [55], assumption-based argumentation [56], and probabilistic argumentation [57]. By extending our approach to these formalisms, we aim to make our solver more versatile and valuable for future research in the field of argumentation.

CRediT authorship contribution statement

Lars Malmqvist: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tangming Yuan:** Writing – review & editing, Validation, Supervision, Methodology. **Peter Nightingale:** Writing – review & editing, Supervision, Software, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is published on GitHub: https://github.com/lmlearning/AFGraphLib.

Acknowledgements

Peter Nightingale is supported by UK EPSRC grant EP/W001977/1.

Appendix A. Additional results tables

See Tables A.19–A.51.

Table A.19
Results DC-PR - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	83.10%	85.36%	77.89%	86.40%	63.06%	0.64	0.54
GCN-WITH-GR	83.96%	86.12%	76.87%	87.52%	69.19%	0.70	0.58
GR-ONLY	63.98%	100.00%	59.69%	100.00%	37.93%	0.43	0.34
HYBRID-GCN-GR	84.69%	88.27%	76.61%	89.93%	68.89%	0.69	0.58

Table A.20
Results DC-PR - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	96.45%	93.08%	96.64%	93.15%	76.11%	0.81	0.80
GCN-WITH-GR	96.82%	95.22%	96.97%	95.27%	80.70%	0.85	0.84
GR-ONLY	95.84%	100.00%	95.47%	100.00%	89.70%	0.91	0.90
HYBRID-GCN-GR	97.39%	96.59%	97.38%	96.65%	93.48%	0.93	0.92

Table A.21
Results DC-PR - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	80.90%	81.63%	80.05%	82.89%	48.52%	0.51	0.44
GCN-WITH-GR	82.90%	81.88%	82.22%	83.21%	61.18%	0.64	0.55
GR-ONLY	70.89%	100.00%	68.31%	100.00%	27.86%	0.36	0.33
HYBRID-GCN-GR	82.52%	82.39%	80.58%	84.81%	54.37%	0.57	0.50

Table A.22
Results DS-PR - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	86.24%	84.53%	86.99%	87.81%	51.82%	0.53	0.50
GCN-WITH-GR	87.66%	84.78%	87.89%	87.55%	57.91%	0.61	0.56
GR-ONLY	84.99%	100.00%	83.31%	100.00%	46.79%	0.51	0.52
HYBRID-GCN-GR	85.14%	76.19%	86.62%	81.00%	57.82%	0.55	0.50

Table A.23
Results DS-PR - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	97.21%	91.44%	97.61%	91.66%	77.09%	0.81	0.81
GCN-WITH-GR	97.88%	94.48%	98.03%	94.59%	80.41%	0.85	0.85
GR-ONLY	98.27%	100.00%	98.13%	100.00%	91.20%	0.92	0.93
HYBRID-GCN-GR	95.58%	77.77%	98.35%	78.33%	92.31%	0.79	0.80

Table A.24
Results DS-PR - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	88.57%	82.43%	90.39%	85.06%	48.17%	0.50	0.50
GCN-WITH-GR	90.97%	84.49%	91.48%	86.22%	57.06%	0.62	0.60
GR-ONLY	90.16%	100.00%	89.33%	100.00%	49.88%	0.55	0.58
HYBRID-GCN-GR	88.54%	78.85%	90.59%	81.87%	56.22%	0.57	0.54

Table A.25
Results DC-CO - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	81.31%	88.16%	69.93%	89.71%	63.11%	0.64	0.46
GCN-WITH-GR	88.02%	84.34%	78.52%	85.13%	75.34%	0.75	0.58
GR-ONLY	63.86%	100.00%	59.65%	100.00%	38.99%	0.43	0.28
HYBRID-GCN-GR	86.58%	86.46%	76.61%	87.42%	77.64%	0.75	0.58

Table A.26
Results DC-CO - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	96.31%	93.21%	96.50%	93.33%	77.62%	0.82	0.81
GCN-WITH-GR	97.15%	92.22%	97.25%	92.28%	85.44%	0.88	0.86
GR-ONLY	96.37%	100.00%	96.01%	100.00%	90.75%	0.92	0.92
HYBRID-GCN-GR	97.47%	97.53%	97.34%	97.56%	94.79%	0.95	0.93

Table A.27
Results DC-CO - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	81.19%	83.34%	80.86%	85.35%	58.25%	0.58	0.47
GCN-WITH-GR	84.75%	82.57%	84.15%	83.60%	65.61%	0.68	0.57
GR-ONLY	74.75%	100.00%	72.21%	100.00%	35.63%	0.42	0.41
HYBRID-GCN-GR	82.56%	84.78%	81.44%	85.98%	63.73%	0.65	0.55

Table A.28
Results DS-CO - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	97.00%	83.37%	98.52%	88.91%	86.84%	0.78	0.83
GCN-WITH-GR	97.54%	86.74%	99.10%	90.76%	91.58%	0.84	0.88
GR-ONLY	100.00%	100.00%	100.00%	100.00%	100.00%	1.0	1.0
HYBRID-GCN-GR	99.64%	98.59%	100.00%	99.05%	100.00%	0.99	0.99

Table A.29
Results DS-CO - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	99.15%	93.16%	99.52%	93.40%	87.92%	0.86	0.90
GCN-WITH-GR	99.00%	93.66%	99.30%	93.91%	85.88%	0.85	0.89
GR-ONLY	100.00%	100.00%	100.00%	100.00%	100.00%	0.96	1.0
HYBRID-GCN-GR	99.98%	99.92%	100.00%	99.93%	100.00%	0.96	1.0

Table A.30
Results DS-CO - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	98.21%	88.93%	99.41%	91.23%	90.42%	0.68	0.88
GCN-WITH-GR	98.17%	92.44%	99.33%	94.12%	91.35%	0.69	0.91
GR-ONLY	100.00%	100.00%	100.00%	100.00%	100.00%	0.78	1.0
HYBRID-GCN-GR	99.90%	99.57%	100.00%	99.67%	100.00%	0.77	1.0

Table A.31
Results DC-ST - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	85.62%	78.92%	79.72%	81.26%	67.60%	0.60	0.49
GCN-WITH-GR	87.06%	80.23%	77.62%	82.23%	72.18%	0.65	0.52
GR-ONLY	64.19%	87.14%	60.22%	90.91%	41.22%	0.36	0.24
HYBRID-GCN-GR	84.66%	77.66%	74.08%	78.79%	73.35%	0.65	0.49

Table A.32
Results DC-ST - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	96.65%	92.44%	96.54%	92.55%	75.15%	0.77	0.77
GCN-WITH-GR	97.13%	93.15%	97.17%	93.24%	81.93%	0.83	0.82
GR-ONLY	96.22%	96.45%	95.86%	96.55%	91.20%	0.89	0.88
HYBRID-GCN-GR	97.55%	94.19%	97.34%	94.24%	94.86%	0.92	0.90

Table A.33
Results DC-ST - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	82.84%	58.21%	81.40%	62.25%	62.66%	0.40	0.32
GCN-WITH-GR	84.67%	59.43%	84.74%	62.96%	70.69%	0.47	0.39
GR-ONLY	74.14%	70.61%	71.67%	76.38%	39.77%	0.22	0.20
HYBRID-GCN-GR	83.23%	58.85%	81.51%	61.29%	64.83%	0.43	0.34

Table A.34
Results DS-ST - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	88.65%	75.31%	90.12%	80.05%	65.01%	0.54	0.46
GCN-WITH-GR	88.29%	69.62%	88.18%	73.61%	60.95%	0.52	0.45
GR-ONLY	78.10%	85.29%	75.87%	89.90%	45.16%	0.39	0.39
HYBRID-GCN-GR	88.44%	78.28%	88.18%	82.45%	64.18%	0.55	0.48

Table A.35
Results DS-ST - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	97.53%	90.35%	97.65%	90.63%	77.64%	0.78	0.78
GCN-WITH-GR	97.14%	86.00%	97.37%	86.31%	76.10%	0.76	0.76
GR-ONLY	97.67%	95.91%	97.48%	96.04%	90.78%	0.88	0.88
HYBRID-GCN-GR	98.14%	94.98%	98.17%	95.12%	92.07%	0.89	0.89

Table A.36
Results DS-ST - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	89.29%	59.92%	89.69%	66.67%	54.63%	0.34	0.32
GCN-WITH-GR	88.50%	55.40%	89.30%	61.24%	54.86%	0.34	0.31
GR-ONLY	86.05%	70.69%	84.89%	76.29%	44.78%	0.27	0.29
HYBRID-GCN-GR	88.86%	65.38%	89.00%	71.22%	52.48%	0.32	0.32

Table A.37
Results DC-SST - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	77.04%	85.15%	72.33%	87.25%	51.36%	0.53	0.42
GCN-WITH-GR	86.00%	88.80%	75.46%	89.93%	65.67%	0.68	0.55
GR-ONLY	64.41%	100.00%	60.12%	100.00%	40.03%	0.44	0.32
HYBRID-GCN-GR	86.64%	90.26%	76.09%	91.54%	72.72%	0.74	0.60

Table A.38
Results DC-SST - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	95.16%	94.27%	95.23%	95.00%	63.51%	0.66	0.66
GCN-WITH-GR	96.87%	93.75%	96.82%	93.84%	77.91%	0.83	0.82
GR-ONLY	96.24%	100.00%	95.83%	100.00%	90.59%	0.92	0.91
HYBRID-GCN-GR	97.75%	97.38%	97.52%	97.42%	94.81%	0.95	0.94

Table A.39
Results DC-SST - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	79.64%	79.10%	81.51%	81.07%	46.17%	0.48	0.41
GCN-WITH-GR	84.77%	84.53%	84.15%	85.89%	56.67%	0.60	0.54
GR-ONLY	76.09%	100.00%	73.48%	100.00%	40.14%	0.46	0.45
HYBRID-GCN-GR	85.84%	84.82%	84.12%	86.32%	66.96%	0.68	0.61

Table A.40
Results DS-SST - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	86.75%	78.94%	90.20%	82.34%	59.76%	0.59	0.53
GCN-WITH-GR	86.94%	81.95%	88.05%	85.23%	52.29%	0.55	0.51
GR-ONLY	85.51%	100.00%	83.80%	100.00%	47.43%	0.51	0.52
HYBRID-GCN-GR	86.63%	90.36%	85.76%	92.41%	53.47%	0.57	0.55

Table A.41
Results DS-SST - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	97.70%	91.17%	98.12%	91.33%	80.95%	0.84	0.83
GCN-WITH-GR	97.32%	92.59%	97.50%	92.73%	74.91%	0.80	0.80
GR-ONLY	98.33%	100.00%	98.18%	100.00%	91.27%	0.92	0.93
HYBRID-GCN-GR	98.39%	98.55%	98.35%	98.58%	92.09%	0.93	0.93

Table A.42
Results DS-SST - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	89.78%	79.43%	91.55%	81.55%	51.25%	0.54	0.53
GCN-WITH-GR	88.43%	77.39%	89.71%	79.93%	43.61%	0.49	0.47
GR-ONLY	90.02%	100.00%	89.16%	100.00%	47.98%	0.53	0.56
HYBRID-GCN-GR	90.37%	90.47%	90.17%	91.54%	52.84%	0.58	0.58

Table A.43
Results DC-STG - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	84.05%	87.42%	68.89%	88.05%	67.34%	0.69	0.46
GCN-WITH-GR	86.84%	88.13%	74.87%	88.85%	73.22%	0.75	0.57
GR-ONLY	61.45%	100.00%	57.08%	100.00%	32.89%	0.39	0.23
HYBRID-GCN-GR	85.76%	89.01%	72.98%	90.01%	74.32%	0.74	0.54

Table A.44
Results DC-STG - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	96.82%	95.02%	96.70%	95.05%	77.67%	0.83	0.82
GCN-WITH-GR	97.00%	94.95%	97.19%	95.00%	81.78%	0.86	0.84
GR-ONLY	96.16%	100.00%	95.76%	100.00%	88.91%	0.90	0.90
HYBRID-GCN-GR	97.54%	96.89%	97.74%	96.94%	94.15%	0.94	0.92

Table A.45
Results DC-STG - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	83.00%	82.97%	81.29%	83.78%	53.11%	0.57	0.46
GCN-WITH-GR	84.08%	82.17%	84.26%	83.29%	64.14%	0.66	0.55
GR-ONLY	73.35%	100.00%	70.59%	100.00%	23.07%	0.34	0.32
HYBRID-GCN-GR	83.20%	82.40%	83.76%	84.36%	59.44%	0.61	0.50

Table A.46
Results DS-STG - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	87.48%	83.28%	87.94%	86.99%	49.03%	0.52	0.48
GCN-WITH-GR	88.81%	85.70%	89.34%	88.01%	55.31%	0.59	0.55
GR-ONLY	85.90%	100.00%	84.24%	100.00%	42.60%	0.48	0.48
HYBRID-GCN-GR	87.86%	92.84%	87.65%	94.58%	50.12%	0.54	0.52

Table A.47
Results DS-STG - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	97.62%	93.31%	98.02%	93.50%	78.48%	0.82	0.82
GCN-WITH-GR	98.00%	95.32%	98.26%	95.40%	81.09%	0.85	0.85
GR-ONLY	98.39%	100.00%	98.24%	100.00%	89.98%	0.91	0.92
HYBRID-GCN-GR	98.42%	99.84%	98.30%	99.84%	90.12%	0.92	0.92

Table A.48
Results DS-STG - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	89.30%	79.63%	91.17%	83.10%	42.10%	0.46	0.44
GCN-WITH-GR	90.50%	81.88%	91.88%	83.86%	51.58%	0.56	0.54
GR-ONLY	90.36%	100.00%	89.49%	100.00%	40.04%	0.48	0.51
HYBRID-GCN-GR	90.54%	98.96%	89.84%	99.14%	40.84%	0.49	0.51

Table A.49
Results DS-ID - equal weighting.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	86.16%	79.43%	87.16%	82.54%	56.30%	0.58	0.52
GCN-WITH-GR	87.28%	81.77%	87.27%	84.35%	54.77%	0.58	0.53
GR-ONLY	85.33%	100.00%	83.57%	100.00%	46.70%	0.51	0.52
HYBRID-GCN-GR	87.44%	87.98%	87.03%	89.93%	57.87%	0.60	0.57

Table A.50
Results DS-ID - complete balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	97.39%	89.95%	97.94%	90.14%	79.71%	0.82	0.82
GCN-WITH-GR	97.59%	94.42%	97.64%	94.52%	76.93%	0.82	0.82
GR-ONLY	98.09%	100.00%	97.92%	100.00%	90.03%	0.91	0.92
HYBRID-GCN-GR	98.31%	98.54%	98.22%	98.57%	91.30%	0.92	0.92

Table A.51
Results DS-ID - reduced balanced.

Type	Accuracy	Acc (yes)	Acc (no)	Precision	Recall	F1	MCC
GCN-NO-GR	88.91%	76.57%	90.72%	79.14%	51.67%	0.55	0.52
GCN-WITH-GR	89.64%	80.90%	89.87%	82.69%	49.02%	0.53	0.53
GR-ONLY	89.30%	100.00%	88.34%	100.00%	44.02%	0.50	0.53
HYBRID-GCN-GR	90.51%	91.09%	90.00%	92.14%	51.18%	0.56	0.57

Appendix B. Analyses of additional semantics

B.1. Additional credulous results

Semi-stable semantics We would intuitively expect the results for semi-stable semantics to most closely resemble those of the stable and preferred semantics. That intuition is not entirely borne out in practice, as the results for these semantics are quite distinctive.

Looking first at the equally weighted setting in Fig. B.14, we note a much reduced performance for the GCN-NO-GR model. It would seem that semi-stable semantics presents a harder approximation problem for a GCN than some of the others we have considered. The overall best performing model is the HYBRID-GCN-GR model, not surprising considering the previous observation.

The complete balanced setting, does not change the picture as much as it has in some other semantics. The GR-ONLY model increases performance as expected, but the ordering among the GCN-based models remains constant.

The reduced balanced setting reverts the picture to one fairly close to the equally weighted one. Overall, the HYBRID-GCN-GR model is the clear winner in terms of performance for credulous acceptance under semi-stable semantics.

On the benchmark side, shown in Table B.52 we can note a similar pattern to stable semantics for AFGen, Barabasi-Albert, Planning2AF, Traffic, Watts-Strogatz, and admbuster graphs. ABA2AF is approximable under semi-stable semantics as it is under preferred semantics. ER graphs prove overall somewhat easier to approximate under semi-stable semantics that we've seen previously, whereas Stable and Traffic benchmarks have some reduced performance. The performance on the LBA benchmark is a bit lower than for stable semantics, but not as bad as for complete semantics. Finally, we can note that the reason the GCN-NO-GR model does poorly under these semantics is mainly attributable to a bad performance on the Grounded benchmark.

As can be seen in Table B.53, the general pattern for size related performance holds for the GR-ONLY, HYBRID-GCN-GR, and GCN-WITH-GR models under semi-stable semantics. However, it breaks for the GCN-NO-GR model as the performance at the low end is much worse that has been seen for other semantics. This demonstrates that while much of the bad performance of this model under semi-stable semantics is attributable to inferior grounded reasoning that is not the whole story.

Stage semantics Stage semantics are the only semantics not based on admissible sets of the ones considered in this article. One might therefore expect a significantly different results than for the other semantics based on the different way extensions are created. However, we don't see any such radical departure from the patterns we have seen, although as in other cases, we see interesting variation in benchmark specific performance.

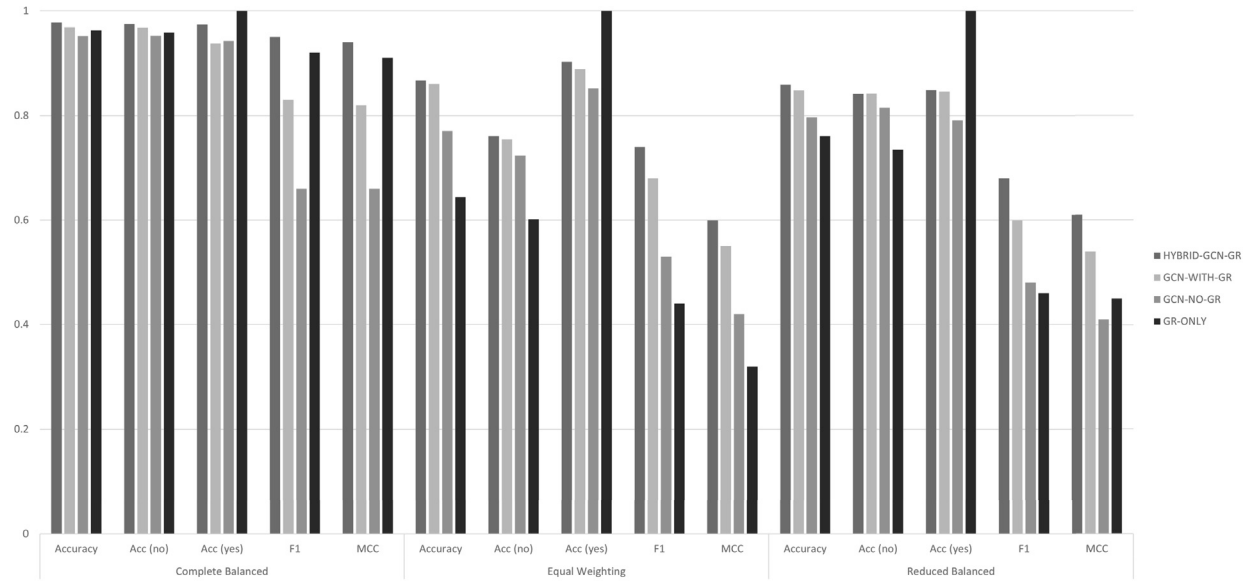


Fig. B.14. Overview of AFGCN approximation results for DC-SST.

Table B.52
Overview of AFGCN approximation results for DC-SST ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	98.55%	98.91%	99.12%	99.27%	0.53	0.65	0.81	0.83
AFGen	54.60%	55.73%	54.60%	54.60%	-0.10	0.021	-0.10	-0.10
Barabasi-Albert	88.27%	92.21%	49.93%	90.56%	0.71	0.75	0.31	0.72
Erdős–Rényi	93.07%	94.72%	96.04%	96.04%	0.74	0.71	0.67	0.67
Grounded	96.84%	97.99%	100.00%	99.87%	0.38	0.69	1.0	0.99
LBA	29.69%	97.40%	1.48%	96.82%	0.20	0.50	-0.70	0.50
Planning2AF	74.36%	82.69%	62.06%	87.74%	0.46	0.62	0.42	0.75
Stable	64.67%	70.95%	63.55%	70.90%	0.20	0.38	0.20	0.37
Traffic	83.63%	80.65%	32.95%	82.36%	0.46	0.48	0.029	0.51
Watts-Strogatz	75.86%	77.81%	75.25%	78.00%	0.12	0.20	0.0	0.23
admbuster	98.64%	99.75%	100.00%	100.00%	0.97	1.0	1.0	1.0

Table B.53
Overview of AFGCN approximation results for DC-SST ordered by band. Accuracy to 2 decimal places, MCC to 2 significant figures.

Band	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
0	48.41%	87.99%	21.85%	86.12%	0.25	0.61	-0.33	0.51
1	87.44%	88.61%	33.25%	91.00%	0.60	0.58	0.023	0.62
2	57.31%	92.93%	26.20%	90.90%	0.44	0.46	-0.091	0.55
3	67.09%	69.39%	62.44%	70.58%	0.26	0.38	0.19	0.31
4	70.63%	74.92%	66.29%	76.47%	0.12	0.20	0.10	0.25
5	70.64%	73.49%	71.06%	74.34%	0.17	0.31	0.15	0.34
6	80.19%	84.45%	76.76%	84.43%	0.37	0.50	0.42	0.58
7	91.43%	94.17%	90.47%	95.25%	0.56	0.72	0.81	0.85
8	95.53%	97.55%	98.31%	99.21%	0.71	0.82	0.96	0.98
9	98.18%	99.23%	100.00%	100.00%	0.66	0.88	1.0	1.0

Table B.54
Overview of AFGCN approximation results for DC-STG ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	97.47%	98.03%	97.43%	97.87%	0.48	0.66	0.52	0.63
AFGen	58.36%	58.03%	54.60%	61.28%	0.018	0.051	-0.10	0.10
Barabasi-Albert	88.82%	93.64%	49.93%	93.77%	0.69	0.78	0.31	0.79
Erdős–Rényi	72.94%	95.05%	29.37%	70.96%	0.37	0.82	-0.67	0.68
Grounded	98.13%	98.46%	100.00%	99.71%	0.71	0.77	1.0	0.97
LBA	84.12%	96.77%	0.90%	94.74%	-0.10	0.40	-0.80	0.10
Planning2AF	82.29%	82.98%	62.06%	81.62%	0.62	0.65	0.42	0.63
Stable	70.77%	71.71%	63.70%	70.23%	0.38	0.43	0.22	0.38
Traffic	83.55%	85.35%	25.81%	83.30%	0.46	0.55	-0.054	0.52
Watts-Strogatz	78.19%	77.75%	75.25%	75.28%	0.27	0.20	0.0	0.024
admbuster	99.78%	99.62%	100.00%	100.00%	1.0	0.99	1.0	1.0

Looking first at the equally weighted results for credulous acceptance in Fig. B.15, we find the GCN-WITH-GR model performing best both in accuracy and MCC terms. The GR-ONLY model performs relatively poorly under this semantics, which also implies a slight dip in performance for the HYBRID-GCN-GR model.

The complete balanced setting shows the now familiar increase in accuracy and the HYBRID-GCN-GR model performing the best followed by the GR-ONLY model. Once again, we see the pattern revert to one closer to the equally weighted setting once we remove the two large grounded-focused benchmarks. This is consistent with what we have seen for other semantics.

We see benchmark specific behaviour that in many ways is familiar from other semantics, especially semi-stable ones. This is true for ABA2AF, AFGen, Barabasi-Albert, Grounded, Planning2AF, Traffic, and admbuster benchmarks. However, we can note that the GR-ONLY model performs unusually poorly on ER graphs under these semantics. There is slightly better performance from GCN-models on Stable and Watts-Strogatz models and slightly worse performance from all models on the LBA benchmark compared to credulous semi-stable semantics. (See Table B.54.)

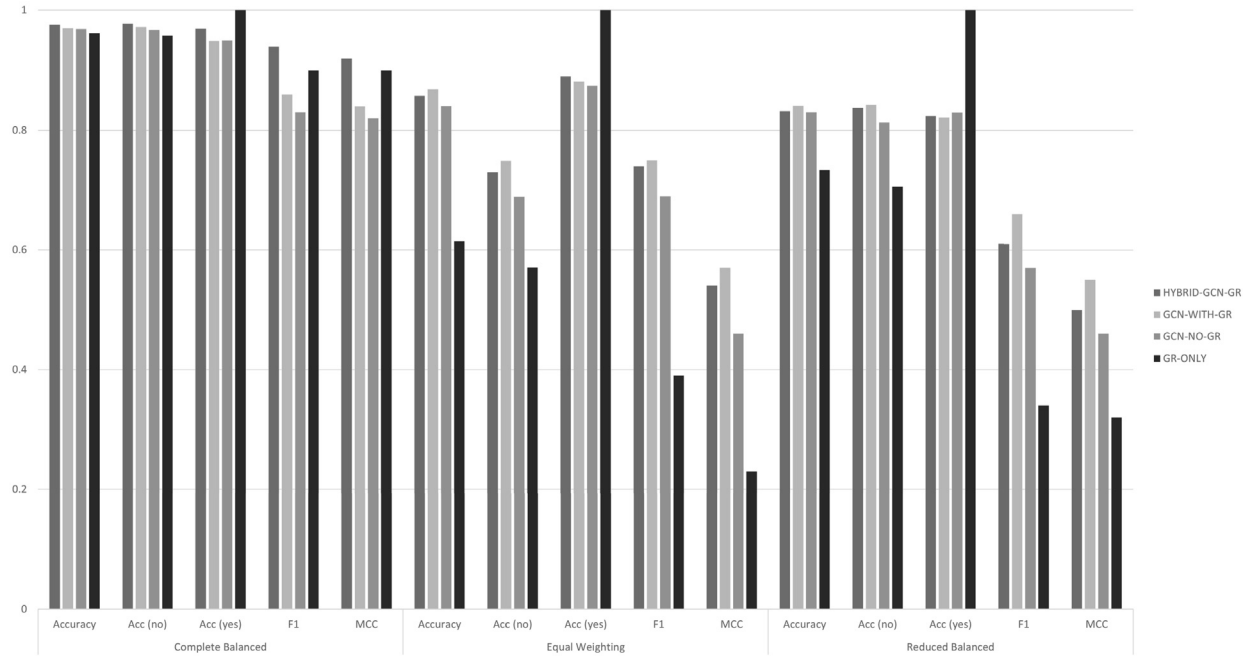


Fig. B.15. Overview of AFGCN approximation results for DC-STG.

Table B.55
Overview of AFGCN approximation results for DC-STG ordered by band. Accuracy to 2 decimal places, MCC to 2 significant figures.

Band	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
0	82.21%	93.17%	14.12%	93.32%	0.23	0.58	-0.52	0.44
1	83.88%	88.88%	33.25%	85.49%	0.45	0.61	0.023	0.54
2	90.23%	93.94%	26.20%	92.96%	0.30	0.51	-0.091	0.35
3	66.02%	72.77%	44.26%	68.06%	0.28	0.41	-0.17	0.46
4	74.71%	74.40%	67.24%	74.81%	0.27	0.21	0.14	0.24
5	73.73%	71.81%	70.70%	72.08%	0.31	0.37	0.14	0.18
6	85.28%	87.25%	79.85%	84.26%	0.38	0.56	0.36	0.48
7	92.58%	91.99%	87.13%	92.41%	0.74	0.75	0.63	0.75
8	97.69%	98.10%	98.31%	99.12%	0.80	0.83	0.96	0.98
9	99.18%	99.23%	100.00%	100.00%	0.91	0.92	1.0	1.0

When we turn to the analysis based on size bands in Table B.55 we see the usual patterns of peaks at small and large bands for the GCN models, while the GR-ONLY model has an especially pronounced dip at the lowest band for these semantics.

B.2. Additional sceptical results

Semi-stable semantics Considering sceptical acceptance under the semi-stable semantics, the results are less divergent than for credulous acceptance. The equally weighted setting in Fig. B.16 shows the HYBRID-GCN-GR model having the best performance on MCC and effectively equal on accuracy with the two other GCN-based models. This is due to increased positive accuracy from the grounded reasoner, as both of the other models have better negative accuracy.

In the complete balanced setting, considering the large grounded frameworks, the performance of the GR-ONLY model and the HYBRID-GCN-GR model are effectively identical.

Removing the two large grounded benchmarks, leads to the HYBRID-GCN-GR model again coming out ahead. But under these semantics the GR-ONLY model remains very competitive. For both of the balanced settings the GCN-NO-GR model outperforms the GCN-WITH-GR model, indicating that for these semantics the model has not learnt to reason effectively with grounded features.

Relative to the stable semantics the results are mainly consistent across benchmarks as shown in Table B.56. ABA2AF is approximable again for sceptical acceptance as well, which is consistent with preferred semantics. There is a drop in performance for LBA frameworks as there was for credulous acceptance. In contrast, there is a performance increase for the Traffic domain.

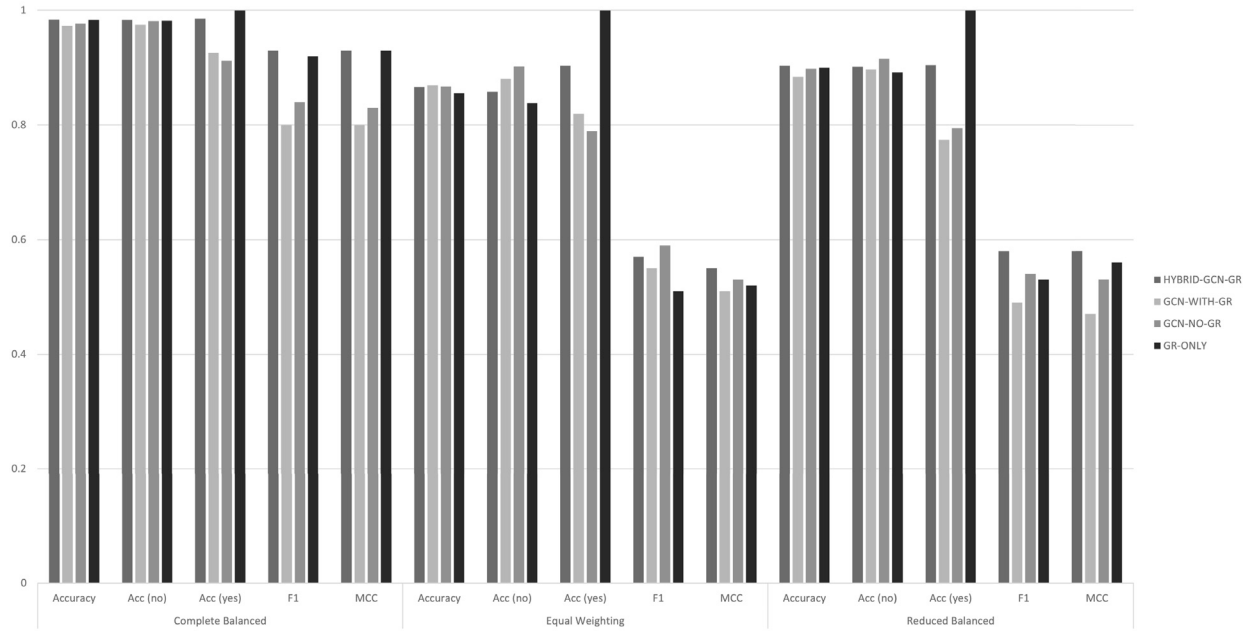


Fig. B.16. Overview of AFGCN approximation results for DS-SST.

Table B.56
Overview of AFGCN approximation results DS-SST ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	99.10%	99.09%	99.43%	99.44%	0.65	0.63	0.82	0.81
AFGen	93.96%	93.96%	93.96%	93.96%	0.0	0.0	0.0	0.0
Barabasi-Albert	82.93%	83.10%	85.38%	84.43%	0.66	0.65	0.74	0.70
Erdős–Rényi	96.04%	96.04%	96.04%	96.04%	0.67	0.67	0.67	0.67
Grounded	98.14%	97.72%	100.00%	100.00%	0.72	0.64	1.0	1.0
LBA	71.23%	77.34%	53.34%	64.67%	0.49	0.57	0.11	0.30
Planning2AF	87.38%	82.11%	88.44%	85.84%	0.70	0.55	0.72	0.66
Stable	79.47%	79.78%	78.75%	80.32%	0.32	0.29	0.23	0.32
Traffic	68.46%	69.34%	69.26%	70.40%	0.42	0.40	0.36	0.36
Watts-Strogatz	82.50%	83.03%	82.03%	82.92%	0.14	0.16	0.0	0.15
admbuster	99.82%	99.68%	100.00%	100.00%	1.0	0.99	1.0	1.0

Table B.57
Overview of AFGCN approximation results for DS-SST ordered by band. Accuracy to 2 decimal places, MCC to 2 significant figures.

Band	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
0	69.87%	76.89%	60.54%	64.18%	0.45	0.56	0.24	0.30
1	69.55%	70.28%	65.09%	70.32%	0.44	0.38	0.26	0.33
2	86.37%	85.41%	88.12%	87.57%	0.52	0.49	0.56	0.55
3	86.28%	84.96%	86.01%	87.22%	0.44	0.43	0.46	0.46
4	89.42%	88.10%	89.37%	89.40%	0.22	0.18	0.22	0.22
5	87.00%	87.21%	86.44%	87.64%	0.28	0.26	0.21	0.32
6	89.99%	89.49%	87.87%	88.46%	0.58	0.56	0.58	0.61
7	93.93%	92.11%	94.86%	94.86%	0.71	0.65	0.83	0.83
8	97.93%	97.18%	99.87%	99.75%	0.84	0.76	1.0	0.99
9	99.46%	99.35%	100.00%	100.00%	0.89	0.87	1.0	1.0

The performance based on size reveals overall lower performance in the smaller size bands as shown in Table B.57. However, there is the same overall pattern that we have seen in general for sceptical acceptance that performance increases with size.

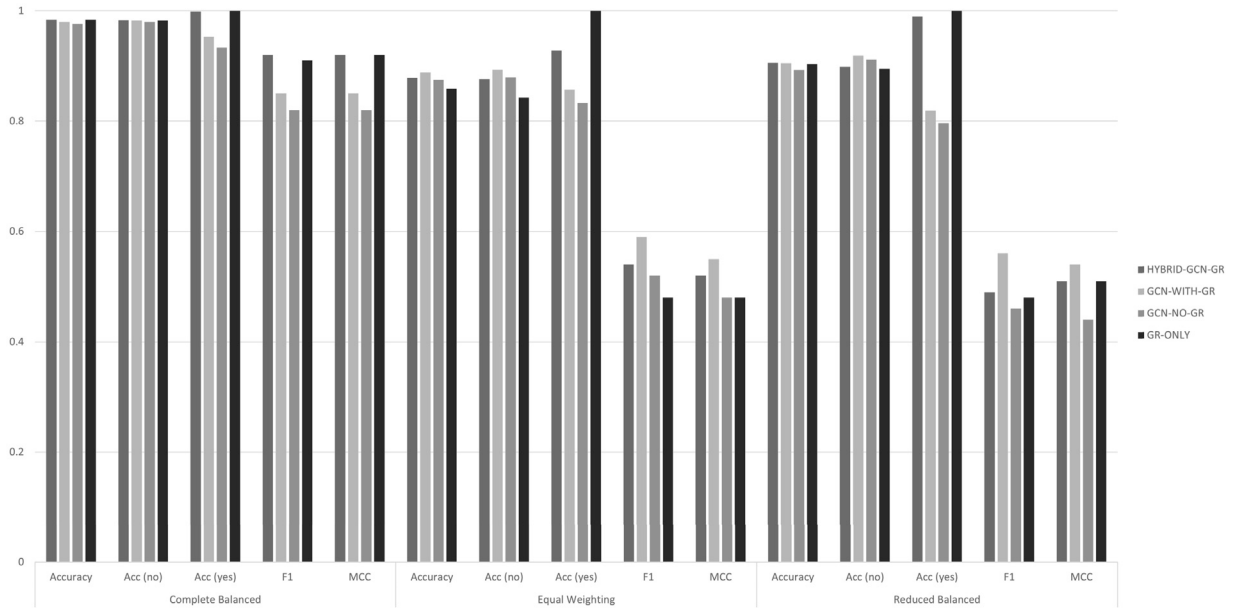


Fig. B.17. Overview of AFGCN approximation results for DS-STG.

Table B.58
Overview of AFGCN approximation results for DS-STG ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	98.36%	98.86%	98.72%	98.72%	0.49	0.66	0.67	0.67
AFGen	93.91%	93.97%	93.91%	93.91%	0.0	0.027	0.0	0.0
Barabasi-Albert	88.11%	89.35%	89.52%	90.30%	0.77	0.78	0.81	0.82
Erdős–Rényi	94.73%	94.73%	94.73%	94.73%	0.0	0.0	0.0	0.0
Grounded	97.92%	98.43%	100.00%	100.00%	0.73	0.78	1.0	1.0
LBA	81.96%	85.60%	57.96%	71.60%	0.61	0.69	0.046	0.33
Planning2AF	85.14%	86.41%	87.69%	87.14%	0.64	0.68	0.71	0.69
Stable	80.08%	81.14%	80.03%	80.26%	0.29	0.36	0.26	0.27
Traffic	63.27%	67.03%	64.97%	70.34%	0.21	0.35	0.29	0.35
Watts-Strogatz	81.92%	83.42%	81.92%	81.92%	0.0	0.23	0.0	0.0
admbuster	99.73%	99.95%	100.00%	100.00%	0.99	1.0	1.0	1.0

Stage semantics Moving on to sceptical acceptance under stage semantics, we see in Fig. B.17 that the GCN-WITH-GR model is the best performer overall in the equally weighted setting as it was for credulous acceptance. As expected the GR-ONLY model does much better in the sceptical context and is on par with the GCN-NO-GR model measured by MCC.

The HYBRID-GCN-GR model and the GR-ONLY model are indistinguishable in terms of performance in the complete balanced setting. The other models also retain good performance in this setting.

The reduced balanced setting has results closer in accuracy terms than we've seen previously, but taking MCC into account, the GCN-WITH-GR slightly outperforms the pack as in the equally weighted setting.

There is a fair degree of overlap with semi-stable semantics in the case of benchmark specific performance. We can see from Table B.58 that seven benchmarks have similar patterns including ABA2AF, AFGen, Grounded, Planning2AF, Stable, Watts-Strogatz, and admbuster benchmarks. There is increased performance for all models on Barabasi-Albert graphs, while ER graphs are unapproximable under these semantics. All GCN-based models perform slightly better on LBA frameworks and all models perform slightly worse on Traffic frameworks.

The size based results for stage semantics are shown in Table B.59. They are consistent with what we have seen for previous semantics and do not present a distinctive pattern for consideration.

Ideal semantics The ideal semantics are defined by the largest admissible set that is a member of all preferred extensions. As such it is related to the grounded extension and like the grounded extension one cannot distinguish between sceptical and credulous acceptance as the ideal extension is unique.

Table B.59
Overview of AFGCN approximation results for DS-STG ordered by band. Accuracy to 2 decimal places, MCC to 2 significant figures.

Band	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
0	69.06%	76.87%	56.77%	69.21%	0.41	0.57	0.20	0.41
1	74.84%	75.57%	67.02%	73.87%	0.41	0.47	0.19	0.34
2	89.59%	90.25%	89.55%	89.80%	0.39	0.41	0.38	0.37
3	90.52%	90.51%	91.30%	90.99%	0.41	0.44	0.43	0.42
4	86.30%	86.74%	87.31%	87.27%	0.26	0.28	0.29	0.28
5	86.50%	87.64%	86.59%	86.82%	0.14	0.35	0.20	0.21
6	90.59%	91.10%	89.70%	89.70%	0.49	0.56	0.56	0.56
7	92.74%	93.46%	93.88%	93.88%	0.65	0.76	0.70	0.70
8	97.42%	98.40%	99.87%	99.86%	0.83	0.86	1.0	0.99
9	99.42%	99.61%	100.00%	100.00%	0.89	0.90	1.0	1.0

Table B.60
Overview of AFGCN approximation results DS-ID ordered by benchmark. Accuracy to 2 decimal places, MCC to 2 significant figures.

Benchmark	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
ABA2AF	99.31%	99.13%	99.36%	99.39%	0.70	0.70	0.77	0.78
AFGen	93.96%	93.96%	93.96%	93.96%	0.0	0.0	0.0	0.0
Barabasi-Albert	85.89%	86.10%	87.03%	86.96%	0.72	0.71	0.77	0.74
Erdős–Rényi	96.04%	96.04%	96.04%	95.38%	0.67	0.67	0.67	0.65
Grounded	97.41%	97.94%	100.00%	99.99%	0.69	0.69	1.0	1.0
LBA	65.74%	73.68%	49.72%	69.76%	0.39	0.53	0.12	0.50
Planning2AF	83.23%	87.22%	88.12%	91.60%	0.61	0.69	0.72	0.80
Stable	80.63%	80.41%	79.24%	80.02%	0.34	0.30	0.23	0.27
Traffic	68.69%	67.59%	69.26%	65.95%	0.40	0.32	0.36	0.30
Watts-Strogatz	82.42%	82.17%	82.03%	82.36%	0.11	0.15	0.0	0.080
admbuster	99.90%	99.90%	100.00%	100.00%	1.0	1.0	1.0	1.0

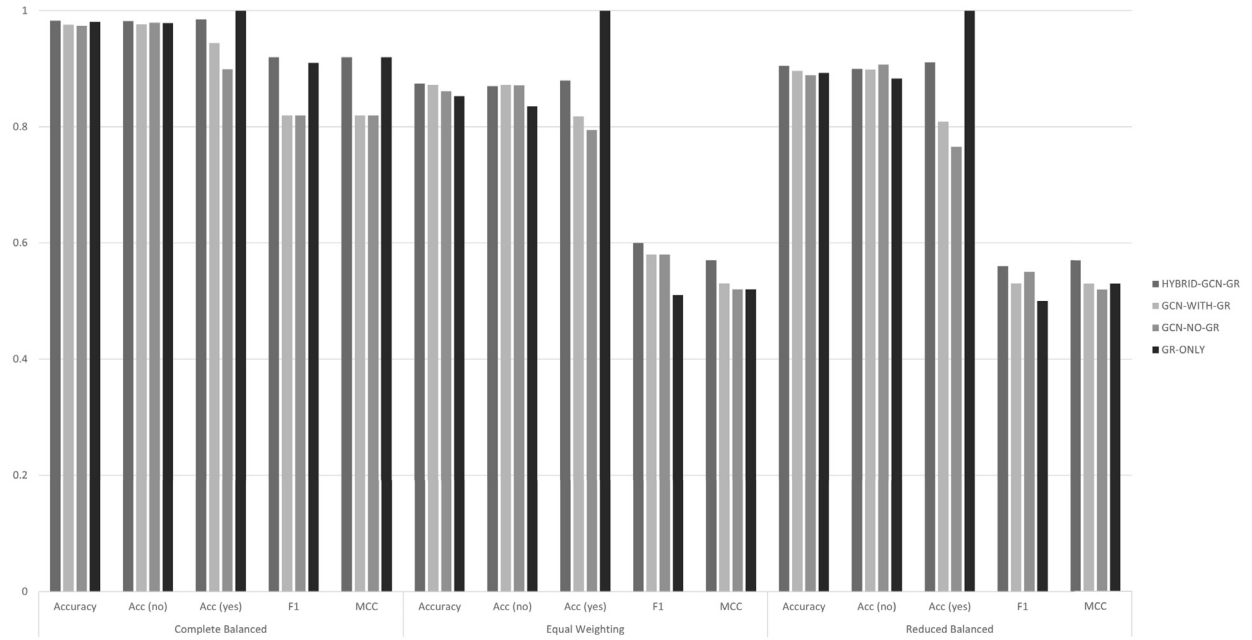


Fig. B.18. Overview of AFGCN approximation results for DS-ID.

Considering the results for sceptical acceptance using the equally weighted setting, shown in Fig. B.18, we find that despite the conceptual similarity with grounded reasoning, the GR-ONLY model does not perform exceptionally well under these semantics. Instead, the HYBRID-GCN-GR model has overall best performance, followed by the GCN-WITH-GR model.

Table B.61
Overview of AFGCN approximation results for DS-ID ordered by band. Accuracy to 2 decimal places, MCC to 2 significant figures.

Band	Accuracy				MCC			
	NO-GR	W/GR	GR	HYBR	NO-GR	W/GR	GR	HYBR
0	69.57%	72.74%	66.60%	72.31%	0.42	0.44	0.37	0.46
1	67.53%	69.86%	64.37%	65.67%	0.38	0.40	0.30	0.34
2	81.04%	82.79%	74.53%	84.22%	0.41	0.46	0.33	0.49
3	91.21%	91.74%	91.61%	91.81%	0.37	0.40	0.39	0.39
4	86.74%	89.62%	89.70%	90.92%	0.36	0.40	0.41	0.44
5	87.71%	87.38%	86.89%	87.51%	0.35	0.33	0.22	0.29
6	91.95%	91.85%	90.28%	92.21%	0.62	0.61	0.61	0.66
7	89.05%	89.70%	90.77%	90.77%	0.57	0.58	0.68	0.68
8	97.85%	98.28%	99.89%	99.88%	0.84	0.83	1.0	1.0
9	99.46%	99.51%	100.00%	100.00%	0.87	0.88	1.0	1.0

The picture for the complete balanced setting is the familiar one with the GR-ONLY and HYBRID-GCN-GR models performing more or less equivalently with the other GCN models following somewhat behind.

Unsurprisingly, this picture changes if we remove the two large grounded-focused frameworks from the equation. This evaluation setting results in the HYBRID-GCN-GR model outperforming the rest, which are relatively close in performance.

The benchmark specific performance is nearly indistinguishable from that under sceptical preferred semantics. All 11 benchmarks are sufficiently close that it is hard to ascribe the minor deviations to anything but chance, excepting a slight reduction across the board for the ABA2AF benchmark. This makes a certain amount of sense, as the overlap between the set of arguments contained in the largest admissible subset of all preferred extensions of an argumentation framework will share much with the set of arguments sceptically accepted for that framework under preferred semantics. (See Table B.60.)

The size band based analysis, shown in Table B.61 shows the pattern of generally ascending performance with size that we are used to, but the performance at the low end is relatively good compared to some other semantics.

Appendix C. Additional runtime results

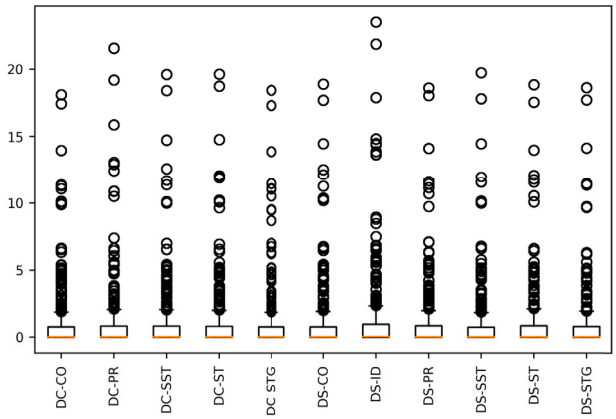
If we look at the median runtime to classify a single argument broken down by semantics shown in Table C.62, we see that the difference per argument of including the grounded features is approximately 0.015 ms. This may be within the acceptable boundary for many applications. We can also observe significant differences in the runtime for different semantics and a general tendency for runtime to be slightly slower for sceptical than for credulous acceptance. See Fig. C.19 for the distribution.

The difference in runtime by benchmark is nearly two orders of magnitude between the fastest, Planning2AF, and the slowest, ABA2AF. While some variation would be expected by benchmark, this is unexpectedly large. You can roughly group the benchmarks into three classes: Fast, Medium, Slow. Fast benchmarks include Planning2AF, Stable, admbuster, Watts-Strogatz, and Barabasi-Albert. Medium include AFGen, ER, Grounded, LBA, and Traffic. ABA2AF is in its own slow category. Interestingly, this partitioning does not straightforwardly map to the classification performance for these benchmarks. See Fig. C.20 for an overview of the distribution. (See Table C.63.)

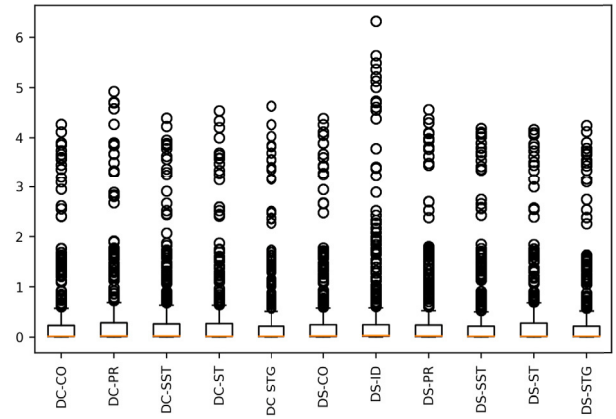
Turning to the results by size band in Table C.64, we do see an obvious and expected pattern. Large frameworks as expected result in longer runtimes and this increases with size fairly reliably, although the band (547.0, 696.0] is an outlier in this regard presumably because it contains more samples from hard benchmarks than the other bands. See Fig. C.21 for an overview of the distribution.

Table C.62
Overview of AFGCN runtime results ordered by semantics. Median runtime given. Results in seconds.

Semantics	Runtime w/GR	Runtime No GR
DC-CO	0.027	0.020
DC-PR	0.031	0.022
DC-SST	0.031	0.021
DC-ST	0.029	0.020
DC-STG	0.029	0.020
DS-CO	0.029	0.022
DS-ID	0.042	0.031
DS-PR	0.028	0.022
DS-SST	0.029	0.019
DS-ST	0.029	0.020
DS-STG	0.027	0.020



(a) w/GR

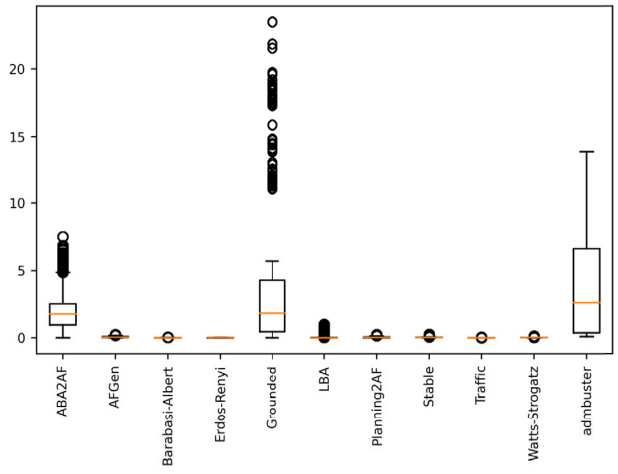


(b) No GR

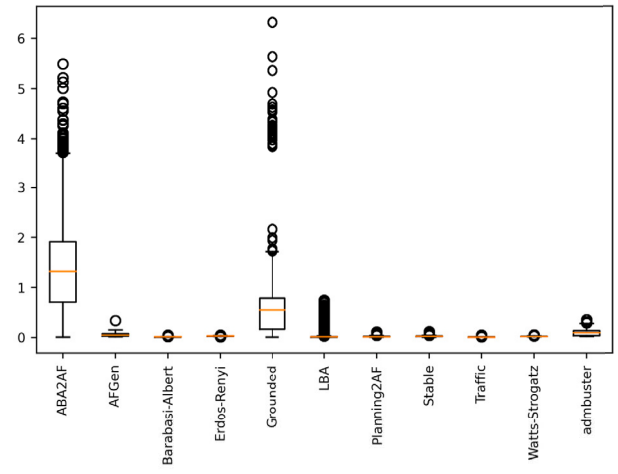
Fig. C.19. Runtime distribution by semantics for the AFGCN solver experiments.

Table C.63
Overview of AFGCN runtime results ordered by benchmark. Median given. Results in seconds.

Benchmark	Runtime w/GR	Runtime No GR
ABA2AF	1.79	1.32
AFGen	0.06	0.05
Barabasi-Albert	0.01	0.01
Erdos-Renyi	0.03	0.03
Grounded	1.84	0.55
LBA	0.01	0.01
Planning2AF	0.02	0.01
Stable	0.04	0.02
Traffic	0.01	0.01
Watts-Strogatz	0.02	0.02
admbuster	2.61	0.10



(a) w/GR

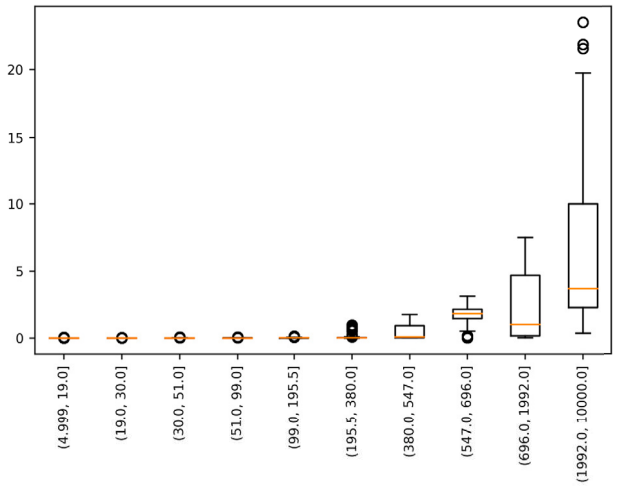


(b) No GR

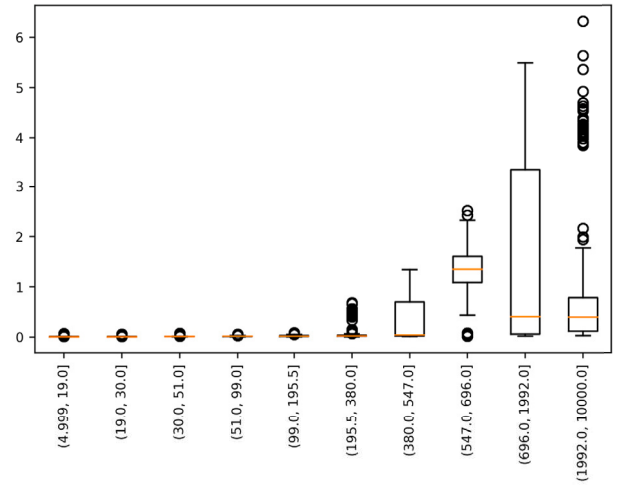
Fig. C.20. Runtime distribution by benchmark for the AFGCN solver experiments.

Table C.64
Overview of AFGCN runtime results ordered by size. Median given. Results in seconds.

Band	Runtime w/GR	Runtime No GR
(4.999, 19.0]	0.01	0.01
(19.0, 30.0]	0.01	0.01
(30.0, 51.0]	0.01	0.01
(51.0, 99.0]	0.02	0.02
(99.0, 195.5]	0.02	0.02
(195.5, 380.0]	0.03	0.02
(380.0, 547.0]	0.10	0.04
(547.0, 696.0]	1.85	1.35
(696.0, 1992.0]	1.02	0.41
(1992.0, 10000.0]	3.69	0.40



(a) w/GR



(b) No GR

Fig. C.21. Runtime distribution by size for the AFGCN solver experiments.

References

[1] F.H.v. Eemeren, R. Grootendorst, A Systematic Theory of Argumentation: The Pragma-Dialectical Approach, Cambridge University Press, 2004.
[2] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (1995) 321–357.
[3] H. Prakken, G. Sartor, Law and logic: a review from an argumentation perspective, Artif. Intell. 227 (2015) 214–245.
[4] A.J.T.T. Norman, A.E.F. Cerutti, R.W.O.M. Srivastava, N. Oren, T.A.D.J.A. Honeywell, U.P. Sullivan, Supporting reasoning with different types of evidence in intelligence analysis, in: AAMAS '15: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, Istanbul, 2015, pp. 781–789.
[5] M. Wooldridge, An Introduction to MultiAgent Systems, John Wiley & Sons, 2009.
[6] L. Malmqvist, T. Yuan, P. Nightingale, Improving misinformation detection in tweets with abstract argumentation, CMNA Workshop Proc. 21 (2937) (2021) 40–46.
[7] P. Novák, C. Witteveen, Context-aware reconfiguration of large-scale surveillance systems: argumentative approach, Argument Comput. 6 (2015) 3–23.
[8] M. Wooldridge, An Introduction to MultiAgent Systems, 2 ed., John Wiley and Sons, New York, NY, USA, 2009.
[9] P.E. Dunne, Computational properties of argument systems satisfying graph-theoretic constraints, Artif. Intell. 171 (2007) 701–729.
[10] I. Kuhlmann, M. Thimm, Using graph convolutional networks for approximate reasoning with abstract argumentation frameworks: a feasibility study, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: LNAI, vol. 11940, 2019, pp. 24–37.
[11] L. Malmqvist, T. Yuan, P. Nightingale, S. Manandhar, Determining the acceptability of abstract arguments with graph convolutional networks, in: 3rd International Workshop on Systems and Algorithms for Formal Argumentation 2020 @ COMMA, vol. 2672, 2020, pp. 47–56.

[12] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, 2017.
[13] L. Malmqvist, T. Yuan, P. Nightingale, Improving misinformation detection in tweets with abstract argumentation, in: Proceedings of the 21st Workshop on Computational Models of Natural Argument, 2021, pp. 40–46.
[14] L. Malmqvist, Approximate Solutions to Abstract Argumentation Problems Using Graph Neural Networks, Ph.D. thesis, University of York, 2022.
[15] P. Baroni, An introduction to abstract argumentation, in: Ecole Professionnelle Commerciale de Lausanne, Basic Training Camp, 2013.
[16] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, Knowl. Eng. Rev. 26 (2011) 365–410, https://doi.org/10.1017/S0269888911000166.
[17] P.M. Dung, P. Mancarella, F. Toni, Computing ideal sceptical argumentation, Artif. Intell. 171 (2007) 642–674.
[18] B. Verheij, Two approaches to dialectical argumentation: admissible sets and argumentation stages, in: Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96), Utrecht, NL, 1996, pp. 357–368.
[19] M.W.A. Caminada, I. Centre, Semi-stable semantics, J. Log. Comput. 22 (2011) 1207–1254.
[20] M. Caminada, An algorithm for stage semantics, Front. Artif. Intell. Appl. 216 (2010) 147–158.
[21] P.E. Dunne, M. Wooldridge, Complexity of abstract argumentation, in: Argumentation in Artificial Intelligence, Springer, 2009, pp. 85–104.
[22] S. Gaggl, Computational complexity of abstract argumentation, Technical Report, TU Dresden, 2013.
[23] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[24] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: a survey, Knowl.-Based Syst. 151 (2018) 78–94.
[25] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, CoRR http://arxiv.org/abs/1606.09375, 2016, arXiv:1606.09375.
[26] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. (2017) 1025–1035.
[27] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, Y. Bengio, Graph attention networks, 6th international conference on learning representations, in: ICLR 2018, Vancouver, 2018, pp. 1–12.
[28] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, D. Tao, A survey on vision transformer, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[29] S. Yun, M. Jeong, R. Kim, J. Kang, H.J. Kim, Graph transformer networks, arXiv:2106.06218 [cs.LG], 2019.
[30] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, Long Beach, 2017, pp. 1025–1035.
[31] D. Craandijk, F. Bex, Deep learning for abstract argumentation semantics, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, 2020, pp. 1667–1673, Main track.
[32] M. Wang, Y. Tang, J. Wang, J. Deng, Premise selection for theorem proving by deep graph embedding, in: Advances in Neural Information Processing Systems, 2017-Decem, 2017, pp. 2787–2797.
[33] H. Lemos, M. Prates, P. Avelar, L. Lamb, Graph colouring meets deep learning: effective graph neural network models for combinatorial problems, in: IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, 2019, pp. 879–885.
[34] G. Charwat, W. Dvořák, S.A. Gaggl, J.P. Wallner, S. Woltran, Methods for solving reasoning problems in abstract argumentation – a survey, Artif. Intell. 220 (2015) 28–63.
[35] F. Cerutti, M. Thimm, M. Vallati, An experimental analysis on the similarity of argumentation semantics, Argument Comput. 11 (2020) 269–304.
[36] M. Thimm, Harper++: using grounded semantics for approximate reasoning in abstract argumentation, in: ICCMA 21, 2021.
[37] D. Craandijk, F. Bex, Enforcement heuristics for argumentation with deep reinforcement learning, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 5573–5581.
[38] J. Klein, I. Kuhlmann, M. Thimm, Graph neural networks for algorithm selection in abstract argumentation, in: ArgML@ COMMA, 2022, pp. 81–95.
[39] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: International Conference on Learning Representations, 2019, https://openreview.net/forum?id=ryGs6iA5Km.
[40] I. Kuhlmann, T. Wujek, M. Thimm, On the impact of data selection when applying machine learning in abstract argumentation, in: Computational Models of Argument, 2022, pp. 224–235.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[42] J. Zhang, L. Meng, Gresnet: graph residual network for reviving deep gnns from suspended animation, arXiv:1909.05729 [cs.LG], 2019, 1–18.
[43] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: ICLR 2015, 2015, pp. 1–15.
[44] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, vol. 9, 2010.
[45] R.A. Johnson, D.W. Wichern, Applied Multivariate Statistical Analysis, 5. ed., Prentice Hall, 2002.
[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS 2017, 2017.
[47] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, Z. Zhang, Deep graph library: a graph-centric, highly-performant package for graph neural networks, arXiv:1909.01315 [cs.LG], 2019.
[48] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with numpy, Nature 585 (2020) 357–362.
[49] M. Caminada, M. Podlaszewski, Admbuster: a benchmark example for (strong) admissibility, in: International Competition on Computational Models of Argumentation, 2017.
[50] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, arXiv preprint, arXiv:1801.07606, 2018.
[51] K. Oono, T. Suzuki, Graph neural networks exponentially lose expressive power for node classification, in: International Conference on Learning Representations, 2019.
[52] G. Li, M. Muller, A. Thabet, B. Ghanem, Can gcns go as deep as cnns?, arXiv preprint, arXiv:1904.03751, 2019.
[53] M. Alviano, The pyglaf argumentation reasoner, OpenAccess Ser. Inform. 58 (2018) 2–5.
[54] L. Malmvist, Yonas: an experimental neural argumentation solver, in: International Competition on Computational Models of Argumentation, 2019.
[55] C. Cayrol, M.C. Lagasquie-Schiex, Bipolarity in argumentation graphs: towards a better understanding, Int. J. Approx. Reason. 54 (2013) 876–899.
[56] P.M. Dung, R.A. Kowalski, F. Toni, Assumption-based argumentation, Argument. Artif. Intell. (2009) 199–218.
[57] M. Thimm, P. Baroni, M. Giacomin, P. Vicig, Probabilities on extensions in abstract argumentation, in: Proceedings of the 2017 International Workshop on Theory and Applications of Formal Argument (TAFA'17), 2017.