

# FastLiGS System Description for ICCMA 2025

Paul Cibier  
Sorbonne Université  
Paris, France  
paul.cibier@gmail.com

Jean-Guy Mailly  
IRIT - Université Toulouse Capitole  
Toulouse, France  
jean-guy.mailly@irit.fr

**Abstract**—We present FASTLiGS, an approximate solver for evaluating the acceptability of arguments under various extension-based semantics. Our solver is based on several consecutive linear layer, and can solve the problems DC-CO, DC-ST, DS-PR, DS-ST, DC-SST, DC-ID and DS-SST of the heuristic track.

**Index Terms**—Abstract Argumentation, Approximate Algorithms, Neural Networks

## I. INTRODUCTION

We submit FASTLiGS to the International Competition on Computational Models of Argumentation (ICCMA 2025). Our solver is able to solve acceptability problems under several Dung semantics [1].

## II. SYSTEM DESCRIPTION

The general process is as follows: we have trained a neuronal network with some linear layer to solve credulous and skeptical acceptability problems under classical Dung semantics. To do so, we have made the choice to implement the inference solver using Rust only instead of Python so the solver can run with a minimum dependency like him-self and the time to process an instance is also reduced.

Notice that some preprocessing is done. First of all, for all cases (except DS-ST), the solver directly answers NO for self-attacking arguments. Then, following several other approaches in the literature [2], [3], the solver first verifies whether the query argument belongs to the grounded extension or is attacked by it, which prevents running the neural network when the grounded semantics is sufficient for solving the problem.

### A. Node Embedding

For each argument in the argumentation framework, we compute features corresponding to the score of the argument w.r.t. several gradual semantics from the literature (h-categorizer [4], no self-attacker [5], Max-based [6], Card-based [6], Euclidian-based [7]) and the grounded semantics (1 if the argument is in the grounded extension, 0 if it is attacked by it, 0.5 otherwise). Other features correspond to the in-degree centrality (number of incoming edges divided by the total number of nodes), the out-degree centrality (number of outgoing edges divided by the total number of nodes), and a

value indicating whether the argument is self-attacking (0 if it is self-attacking, 0.5 otherwise).

### B. Network Architecture

We propose a personal structure of neural network composed of four linear layers and a dropout layer:

- First layer: 9 input features to  $9^2$  output features,
- Second layer:  $9^2$  input features to  $9^2$  output features,
- Dropout layer: to help preventing overfitting, with a probability of 20%,
- Third layer:  $9^2$  input features to 9 outputs features,
- Fourth layer: 9 input features to 1 output with the sigmoid activation function.

The leaky relu activation function was used between all the layers. The output of the sigmoid function represents the probability of acceptability of the argument. The final answer of the solver is then YES if and only if this probability is strictly higher than 0.5.

### C. Training and Running the Solver

a) *Training*: The training was done using PyTorch with Python. 2500 epoch were needed to train the neural network with a dataset composed of a fusion between the benchmarks from the ICCMA 2017 and ICCMA 2023 and soem instances generated with crusti 2gio tools with a maximum of 200 000 arguments per instances (this limit comes from a memory consideration).

After the training, the architecture of the neural network and the associate weight were saved to the ONNX format. When compiling the Rust solver the architecture and the weight are extracted for the ONNX file and integrated into the rust solver.

b) *Running*: If the binary is build with cargo in release mode, the binary is generally in the target/release directory. Running FASTLiGS follows the requirements of ICCMA 2025, i.e.:

```
./fligs -p TASK -f FILE -a argument
```

where TASK is one of DC-CO, DC-ST, DS-PR, DS-ST, DC-SST, DC-ID and DS-SST, FILE is the path to a file describing an AF in the CNF format, and argument is the name of an argument in FILE.

One can obtain information on the solver by typing:  

```
./fligs -h.
```

## III. CONCLUSION

The source code of our solver is available here: <https://github.com/Paulo-21/FastLiGS>.

The second author is funded by the French National Research Agency (ANR, Agence National de la Recherche) under the grants ANR-22-CE23-0005 and ANR-22-CPJ1-0061-0.

## REFERENCES

- [1] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artif. Intell.*, vol. 77, no. 2, pp. 321–358, 1995. [Online]. Available: [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X)
- [2] M. Thimm, “Harper++ v1.1.1,” in *Fifth International Competition on Computational Models of Argumentation (ICCMA’23)*, September 2023.
- [3] P. Cibir and J.-G. Mailly, “Graph convolutional networks and graph attention networks for approximating arguments acceptability,” in *Computational Models of Argument - Proceedings of COMMA 2024, Hagen, Germany, September 18-20, 2024*, ser. Frontiers in Artificial Intelligence and Applications, C. Reed, M. Thimm, and T. Rienstra, Eds., vol. 388. IOS Press, 2024, pp. 25–36. [Online]. Available: <https://doi.org/10.3233/FAIA240307>
- [4] P. Besnard and A. Hunter, “A logic-based theory of deductive arguments,” *Artif. Intell.*, vol. 128, no. 1-2, pp. 203–235, 2001. [Online]. Available: [https://doi.org/10.1016/S0004-3702\(01\)00071-6](https://doi.org/10.1016/S0004-3702(01)00071-6)
- [5] V. Beuselinck, J. Delobelle, and S. Vesic, “A principle-based account of self-attacking arguments in gradual semantics,” *J. Log. Comput.*, vol. 33, no. 2, pp. 230–256, 2023. [Online]. Available: <https://doi.org/10.1093/logcom/exac093>
- [6] L. Amgoud, J. Ben-Naim, D. Doder, and S. Vesic, “Acceptability semantics for weighted argumentation frameworks,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, C. Sierra, Ed. ijcai.org, 2017, pp. 56–62. [Online]. Available: <https://doi.org/10.24963/ijcai.2017/9>
- [7] A. Libman, N. Oren, and B. Yun, “Abstract weighted based gradual semantics in argumentation theory,” *CoRR*, vol. abs/2401.11472, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2401.11472>