Geometric Deep Learning Towards the Iterative Classification of Graph-Based Aircraft Thermal Management Systems

AIAA-2024-0684

Anthony Sirico Jr., Daniel R. Herber

Colorado State University – Department of Systems Engineering
anthony.sirico@colostate.edu

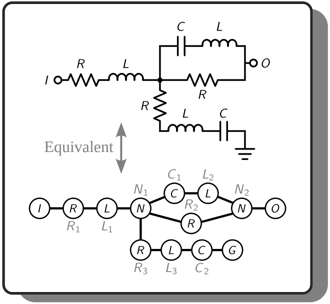
January 9, 2024; Orlando, FL

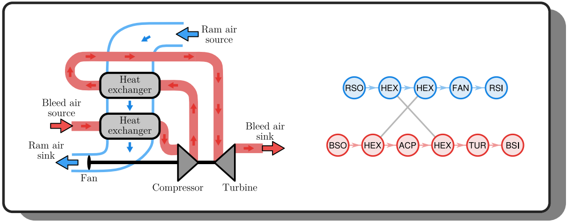
1

Introduction

Graph Representation of Engineered Systems

- **Graphs** can be used as a model for a variety of engineered systems
- Here we consider labeled graphs denoted by G
- We seek to determine whether this is a useful graph or not





Design Situation of Interest

- Often, the performance (or value or utility) of a given graph $J(G_i)$ can be determined through analysis
- Consider the following three types of graph-centric design problems[1] with T is the amount of time allocated to complete the graph design study:

Type 0 All desired graphs can be generated, and so can their performance metric $J(G_i)$ within time T

Type 1 All desired graphs can be generated, but only some of the performance metrics $J(G_i)$ can be evaluated within time T; the analysis is too expensive

Type 2 All desired graphs cannot be generated within time T

- This **work focuses on methods for Type 1 problems** (using data from a large Type 0 study)

[1] Guo, Herber, and Allison 2018

Case Study: Aircraft Thermal Management System (TMS)

- The design case study here seeks to **identify the top set of TMSs** (each represented by graph G_i) assessed by the value $J(G_i)$ of:

$$\text{minimize: } J = w_1 J_1 + w_2 J_2 + w_3 J_3 + w_4 J_4 + w_5 J_5 \quad (1a)$$

$$= w_1(\bar{T}_{HLF} - 343K) + w_2(\bar{T}_{HLR} - 300K) + w_3 C_{total} + w_4 \dot{m}_{BSI} + w_5 \dot{m}_{RSI} \quad (1b)$$

where an aircraft TMS is tasked with controlling the temperatures of two loads: flight control heat load $\bar{T}_{HLF}$, and radar heat load $\bar{T}_{HLR}$[1]

- It is feasible to generate all unique graphs up to a specific size by utilizing previously developed efficient graph enumeration methods[2]
- 32,612 potential architectures were generated
- But, evaluating this many graphs using physics-based Modelica models is extremely expensive, making this a suitable dataset to explore posed on Slide 3

[1] Herber et al. 2020 [2] Buettner et al. 2021

Case Study (2)

- Is the graph capable of being compiled (i.e., can a physics-based model be automatically constructed from a given graph)?
- Is the graph capable of being simulated (i.e., given a complied model, are there valid simulation outcomes)?
- Of the 32,612 potential architectures, only 5,585 successfully **compiled**
- Out of the 5,585 *compiled* graphs, only 2,098 were able to **simulate**
- *Simulatability* and best performance $J(G_i)$ were determined by attempting to simulate *compiled* models using 200 different parameter sets[1]
  - If at least one set produced any valid results, it is considered *simulatable*, and the lowest J value is assigned to its respective graph

[1] Herber et al. 2020

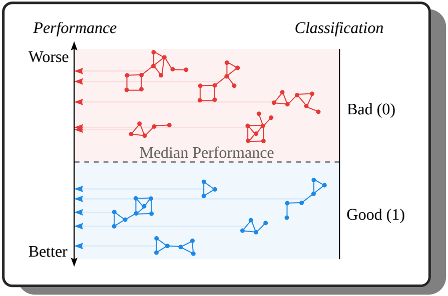
Key Questions

- Often the goal is not to narrow the potential graphs down to *one* particular graph, but rather a *group* of “useful”[1] graphs that would be analyzed further
  - Sometimes at a higher fidelity due to assumptions made in modeling
  - To explore trade-offs (e.g., performance vs. complexity)
- ? Using a predefined portion of the 32,612 graphs, can a GDL model determine if the remaining graphs are **compilable** and **simulatable**?
- ? Given a Type 1 problem, can we provide a reasonable likely set of “useful” graphs without evaluating each of their performance $J(G_i)$?
- ? If so, how should we approach this challenge to reduce overall design study computational cost?

[1] Useful is defined as satisfying a specific set of conditions

Graph Classification

- **Graph classification** approaches seek to assign graphs to a class based on a predetermined criteria — optimized assessment $J(G_i)$ from Eq. (1)
- Predictive models for classification are less concerned about absolute positioning than correct class assignment — aligned with the search for the top potential candidates
- **Why not regression?** — Facilitates better down-selection in the early stages of conceptual design



2

Methodology

Geometric Deep Learning (GDL) & Graph Neural Networks (GNNs)

- We consider **Geometric Deep Learning (GDL)** as a potential strategy for the classification goal
- GDL is an umbrella term encompassing a technique that generalizes neural networks to Euclidean and non-Euclidean domains, such as graphs, manifolds, meshes, or string representations[1]
- In essence, GDL encompasses approaches that incorporate information on the input variables' structure space and symmetry properties and leverage it to improve the quality of the data captured by the model
- GDL uses **Graph Neural Networks (GNNs)**[2], which have convolutional layers to determine node embeddings and pooling layers to average node embeddings
- GDL has been used in a variety of areas[3]

[1] Bronstein, Bruna, Cohen, et al. 2021; Bronstein, Bruna, LeCun, et al. 2017 [2] Bengio 2012 [3] Wong et al. 2022; Pfaff et al. 2021; Park and Park 2019; Zhang, He, and Katabi 2019; Xiao, Ahmed, and Sha 2023; Ferrero et al. 2021; Atz, Grisoni, and Schneider 2021; Gainza et al. 2020; Krokos, Bordas, and Kerfriden 2022

Multi-Label Classification for Compilation and Simulatability

- When there are multiple labels, we can represent them for graph G_i as:

$$\mathbf{L}(G_i) = [l_1, l_2, \ldots, l_{c-1}, l_c] \tag{2}$$

where l_c represents a different aspect or sub-label

- The answers to these questions gives us four possible categories:
  1. Graphs that will not compile
  2. Graphs that will compile
  3. Graphs that will not simulate
  4. Graphs that will simulate

$$\mathbf{L}(G_i) = \begin{cases} [0,1,0,1] & \text{if } G_i \text{ will compile and simulate} \\ [0,1,1,0] & \text{if } G_i \text{ will compile but not simulate} \\ [1,0,1,0] & \text{if } G_i \text{ will not compile} \end{cases} \tag{3}$$

Iterative Classification for Down-Selection

- We aim to select a smaller subset with higher median performance from $\mathcal{G}_{all}$ by iteratively constructing GDL models, as outlined in the following steps:

Algorithm (Iterative Classification for Down-Selection)

$k \leftarrow 1$ ▷ *Initialize iteration counter*
Create initial $\mathcal{G}^k_{\text{known}}$ ▷ *Create initial set of known graphs*
while $k \leq n$ ***do*** ▷ *Iterate until specified limit*
    Create GDL model $m^k(G_i)$ *using* $\mathcal{G}^k_{\text{known}}$
    Divide $\mathcal{G}^k_{\text{known}}$ *into "Known 1" and "Known 0" based on median* J *using* $\mathcal{G}^k_{known}$
    Use $m^k(G_i)$ *to predict classes of* $\mathcal{G}^k_{\text{unknown}}$
    Form sets "Predicted 1" and "Predicted 0" from these predictions
    Set $\mathcal{G}^{k+1}_{\text{known}} \leftarrow$ *"Known 1"*
    Set $\mathcal{G}^{k+1}_{\text{unknown}} \leftarrow$ *"Predicted 1"*
    Exclude remaining graphs, assuming them to be less valuable or useful
    $k \leftarrow k + 1$ ▷ *Increment iteration counter*
end while

→ Tools and Code Availability

Table: List of Tools

Tool	Version
Python	3.9
PyTorch-Geometric[1]	2.1.0
PyTorch	1.12.1
Networkx	2.8.7
SciPy	1.9.1
Pandas	1.5.0

[1] Fey and Lenssen 2019

Code and dataset:
https://github.com/anthonysirico/GDL-for-Engineering-Design

3

Results

A Model for Multi-Label Classification for Compilation and Simulatibility

- *Known Set Size and Training Epochs*[1]
  - What is an appropriate portion (%) of the dataset is needed to construct a *reasonable* GDL model?

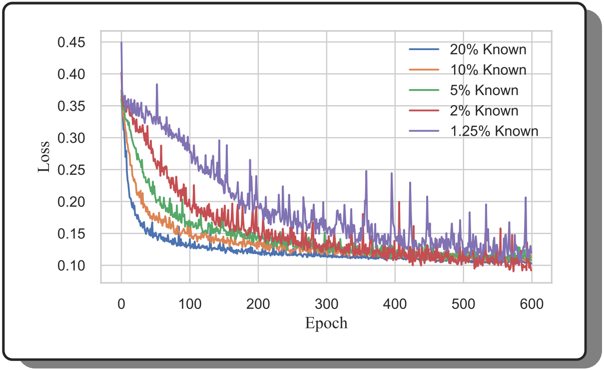
Known Size %	Mean Accuracy	Mean AUC
20	0.915	0.860
10	0.909	0.845
5	0.902	0.845
2.5	0.881	0.825
1.25	0.866	0.824

Table: The model metrics for different N_{known} averaged over the seven independent runs.

[1] Precision, Recall, and F1 can be found in the paper.

Multi-Label Classification (2)

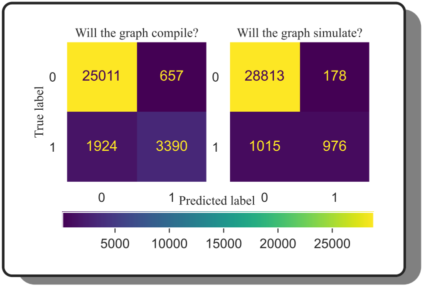
- *Known Set Size and Epochs continued*
  - Several sizes of the known dataset (all graphs have $J(G_i)$ known) were selected
  - Many epochs were also used to help determine a typical stopping condition

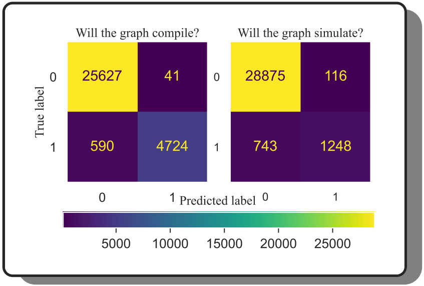


Multi-Label Classification (3): Feature Engineering

- **Harmonic Centrality** the sum of the reciprocals of the shortest path distances d from all other nodes to a specific node u

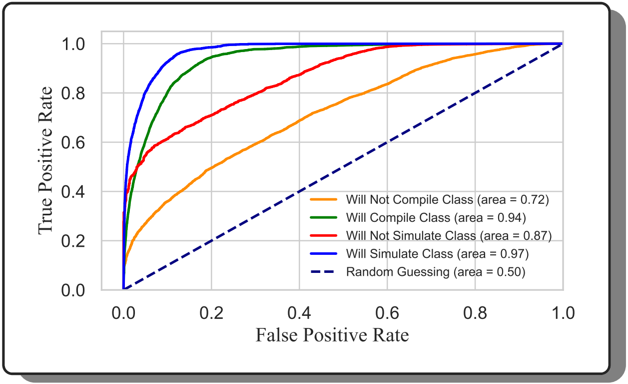
$$C(u) = \sum_{v \neq u} \frac{1}{d(v,u)} \tag{4}$$





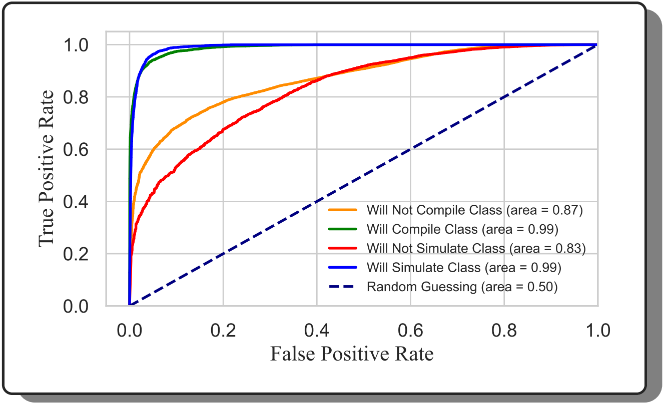
Multi-Label Classification (4): ROC & AUC without Feature Engineering

- **Receiver Operating Characteristic Curve (ROC)**
  - Displays the performance of the model at all classification thresholds
- **Area Under the Curve (AUC)**
  - Offers a comprehensive performance assessment over all potential classification thresholds



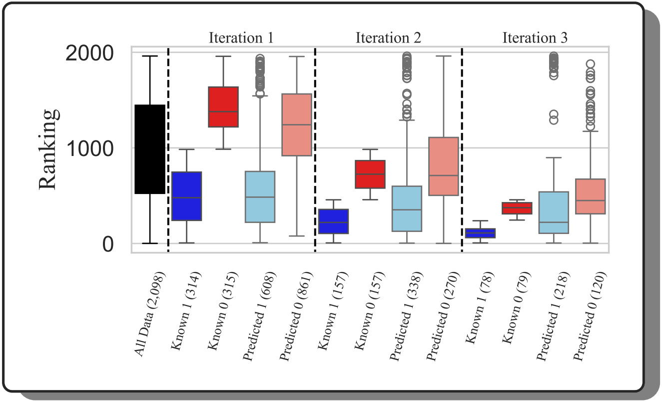
Multi-Label Classification (5): ROC & AUC with Feature Engineering

- Adding *harmonic centrality* greatly increased the models ability to discern patterns, leading to more accurate and efficient models



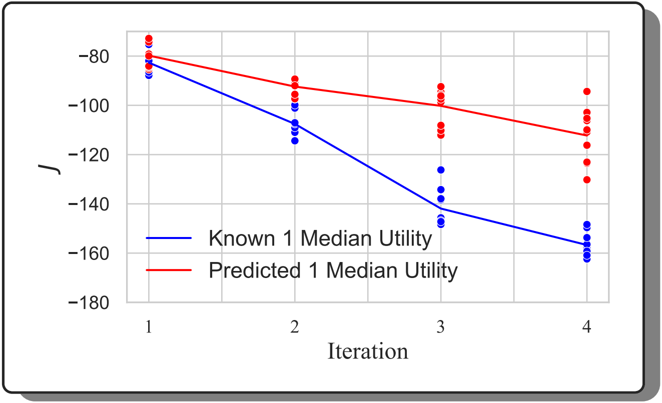
Iterative Classification

- Using the approach described on Slide 10, three iterations were performed
- Good separation occurs between the Predicted 1 and Predicted 0 sets at Iteration 3



Iterative Classification (2)

- Here we show the median values of the “Known 1” and “Predicted 1” sets averaged over ten runs using the iterative GDL classification approach
- The previous slide is one of the ten runs in this figure



4

Conclusions & Future Work

Conclusions and Future Work

- We presented a Geometric Deep Learning (GDL) approach for classifying and down-selecting graph-based aircraft Thermal Management Systems (TMSs) toward sets of better-performing solutions
- Observations were made of intriguing trade-offs between accuracy and computational cost for this task
- Potential future work items include:
  - Merging the two main tasks from this study into one main workflow
  - Regression approaches for predicting graph performance
  - A Pareto set of solutions is employed, where it may include multiple instances of the same graph, each with varying parameter values

5

References

→ References

- K. Atz, F. Grisoni, and G. Schneider (2021). *Geometric deep learning on molecular representations*. DOI: 10.48550/arXiv.2107.12375
- Y. Bengio (2012). *Practical recommendations for gradient-based training of deep architectures*. preprint v2. arXiv:1206.5533
- M. M. Bronstein, J. Bruna, T. Cohen, et al. (2021). *Geometric deep learning: grids, groups, graphs, geodesics, and gauges*. preprint v2. arXiv:2104.13478
- M. M. Bronstein, J. Bruna, Y. LeCun, et al. (2017). "Geometric deep learning: going beyond euclidean data". *IEEE Signal Processing Magazine* 34.4. DOI: 10.1109/msp.2017.2693418
- R. Buettner et al. (2021). "An automated design tool for the generation and selection of optimal aircraft thermal management system architectures". *AIAA Propulsion and Energy 2021 Forum*. DOI: 10.2514/6.2021-3718
- V. Ferrero et al. (2021). "Classifying component function in product assemblies with graph neural networks". *ASME J. Mech. Design* 144.2. DOI: 10.1115/1.4052720
- M. Fey and J. E. Lenssen (2019). "Fast graph representation learning with PyTorch Geometric". *ICLR Workshop on Representation Learning on Graphs and Manifolds*
- P. Gainza et al. (2020). "Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning". *Nature* 17. DOI: 10.1038/s41592-019-0666-6
- T. Guo, D. R. Herber, and J. T. Allison (2018). "Reducing evaluation cost for circuit synthesis using active learning". *ASME International Design Engineering Technical Conferences*. DETC2018-85654. doi: 10.1115/DETC2018-85654. DOI: 10.1115/DETC2018-85654

→ References (Continued)

- D. R. Herber et al. (2020). "Architecture generation and performance evaluation of aircraft thermal management systems through graph-based techniques". *AIAA 2020 Science and Technology Forum and Exposition*. AIAA 2020-0159. DOI: 10.2514/6.2020-0159
- V. Krokos, S. P. A. Bordas, and P. Kerfriden (2022). *A graph-based probabilistic geometric deep learning framework with online physics-based corrections to predict the criticality of defects in porous materials*. DOI: 10.48550/arXiv.2205.06562
- J. Park and J. Park (2019). "Physics-induced graph neural network: an application to wind-farm power estimation". *Energy* 187. DOI: 10.1016/j.energy.2019.115883
- T. Pfaff et al. (2021). *Learning mesh-based simulation with graph networks*. DOI: 10.48550/arXiv.2010.03409
- J. C. Wong et al. (2022). "Graph neural network based surrogate model of physics simulations for geometry design". *IEEE Symposium Series on Computational Intelligence*. DOI: 10.1109/SSCI51031.2022.10022022
- Y. Xiao, F. Ahmed, and Z. Sha (2023). "Graph neural network-based design decision support for shared mobility systems". *ASME J. Mech. Design* 145.9. DOI: 10.1115/1.4062666
- G. Zhang, H. He, and D. Katabi (2019). "Circuit-GNN: graph neural networks for distributed circuit design". *International Conference on Machine Learning*

Questions?

Geometric Deep Learning Towards the Iterative Classification of Graph-Based Aircraft Thermal Management Systems
AIAA-2024-0684

Anthony Sirico Jr. and Daniel R. Herber
Colorado State University
{anthony.sirico, daniel.herber}@colostate.edu
www.engr.colostate.edu/~drherber

Links to the code on GitHub:
https://github.com/anthonysirico/GDL-for-Engineering-Design
doi: 10.48550/arXiv.2303.09770