10-708 PGM - Final Report: PGM-Based Multi-Model Calibration for Onboard Simulations in Deep Space Habitats

Nicolas Gratius
PhD student, CEE
ngratius@andrew.cmu.edu

Adiel Felsen
MS student, MLD
afelsen@andrew.cmu.edu

Sudeep Agarwal
MS student, MLD
sudeepag@andrew.cmu.edu

Advisor: Fan Pu Zeng
10-708 TA
fzeng@andrew.cmu.edu

1 Introduction

Future space habitats will need to be more independent from mission control on earth due to complications such as communication delays between deep space and earth. Onboard reasoning capabilities can be enhanced using computational models that replicate the behavior of the spacecraft and is sometimes referred to as a 'digital twin'. Calibrating these models during flight is challenging as they must remain consistent with each other to run co-simulations and are often highly heterogeneous. In this work, we propose abstracting all model parameters as random variables in a probabilistic graphical model (PGM), such that performing inference in the network results in jointly calibrating the onboard models. We consider the scenario described in Figure 1. A vehicle is approaching a space habitat and the crew members initiate system check procedures. They query an onboard computational tool to simulate the expected CO_2 concentration onboard after docking to decide whether the manoeuvre is safe. Both the vehicle and space habitat are equipped with a CO_2 removal system, but the one in the habitat is experiencing a valve stiction fault. In this study, we assume that a diagnosis module has already identified the fault, and our work focuses on calibrating and executing the simulation models.

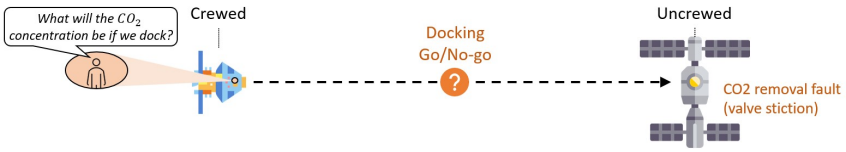


Figure 1: Scenario: System check before docking

2 Background

We organized this work in two parts that address accuracy and scalability respectively. The focus of the midway report is described in section 5.1, where we demonstrate how the use of PGMs for model calibration leads to a higher accuracy over other baselines when evaluating the final simulation output of the combined models. In Section 5.2, we build upon this by exploring the scalability of this method, comparing the computational costs of exact inference with approximate inference methods such as forward sampling, importance sampling, and rejection sampling. Contrary to the first section, we define accuracy in this section as the proportion of correct node assignments for approximate

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

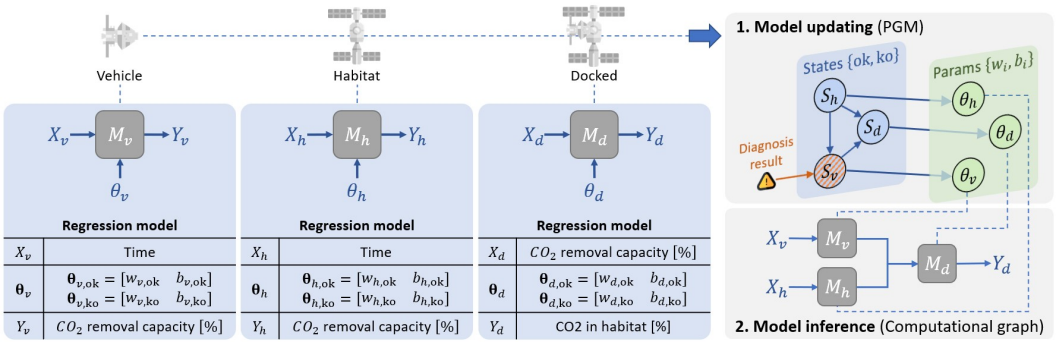


Figure 2: Simulation models and integration setup

inference methods compared to the exact inference methods. We show that there is an inherent trade-off between inference accuracy and speed.

3 Related Work

Literature Review The use of PGMs to combine different models is a method that has been implemented in different studies. Some research has focused on creating a Bayesian network made of nodes corresponding to the inputs (X_i) and outputs (Y_i) of each models (i) to be integrated [1] [2]. These approaches however do not represent the model parameters (θ_i) as random variables in the network. Other studies do include the model parameters but these are either represented as a single parameter (which we assume unrealistic given the complexity of existing space habitat models) or a set of independent parameters (which is usually not representative of an integrated system) [3].

External Resources In addition to a review on existing work in the field, we also reference [4] for an overview of some approximate inference algorithms like importance sampling. Our code made extensive use of pgmpy [5], which is an open-source library for Bayesian networks with implementations of several inference methods. We also used [6] to randomly generate directed acyclic graphs for our simulations.

4 Methodology

4.1 Part 1: Simulation Accuracy Analysis

Synthetic Data Generation Since empirical CO_2 data is not available to us, and to enable ease of modeling, we generated synthetic data for (1) the CO_2 removal capacity of the vehicle, (2) the CO_2 removal capacity of the habitat, and (3) the combined CO_2 concentration when the vehicle is docked to the habitat. Synthetic data for each model is generated using a linear function with different slopes and intercepts, to which Gaussian noise is added.

Modeling We developed regression models for each of the three following systems: (1) the CO_2 removal system of the vehicle, (2) the CO_2 removal system of the habitat, and (3) the combined cabin air when the vehicle is docked to the habitat (see Figure 1). The first two categories of models simulate the CO2 removal capacity, i.e., the ratio between the inlet CO_2 concentration and the outlet CO_2 concentration, and the last one uses the outputs of the two previous models to simulate the CO_2 concentration in the combined habitable space. Each of the three systems is associated with two regression models which represent nominal behavior, labeled 'OK', and off-nominal behavior, labeled 'KO'. A Bayesian network is used to represent the dependencies between system states and model parameters (see Figure 2).

Calibration The proposed calibration process is described in Figure 3. First, the model parameter for the habitat is derived from the output of an existing diagnosis module. The most likely parameters for the vehicle and the combined configuration are then updated according to their respective

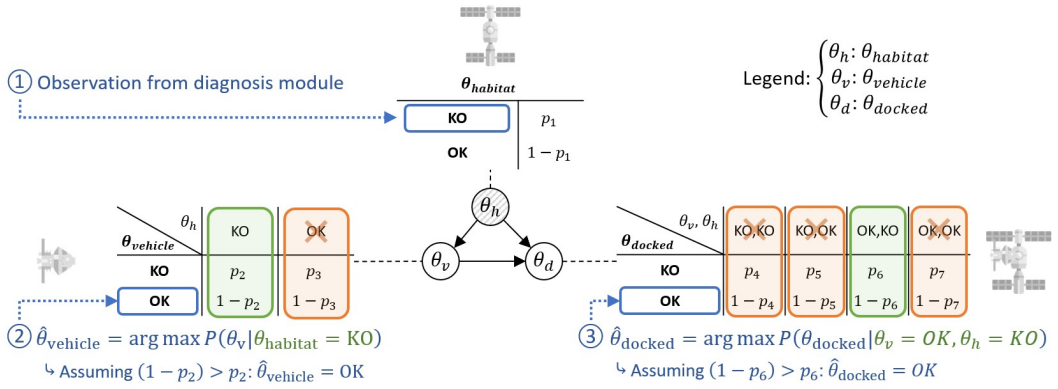


Figure 3: Calibration process

conditional probability tables. Finally, these most likely parameters are passed to the simulation models so that inference is performed according to the latest system state known.

4.2 Part 2: Computational Cost Analysis

To evaluate the scalability of our method we focused on the inference task performed by the PGM only. Simulation models such as the three regressions from Section 4.1 are not executed here as we are only interested in inferring their parameters.

Graph Construction In order to scale our model, we need to generate directed acyclic graphs of arbitrary size. This is not trivial as randomly assigning connections between nodes can lead to cycles which is incompatible with Bayesian networks.

We utilize a library designed to randomly generate directed acyclic graphs [6]. This library allows setting parameters α and β which respectively control the width and regularity of the graph. We used the default parameters for this work. An example of a generated graph can be seen in Figure 4.

In our analysis, we created three sub-graphs to reflect the system components of the use case. We assume that sub-graphs are composed of single root nodes (h_0, v_0, c_0) associated to models of the power systems, and single leaf nodes $(h_{CO2}, v_{CO2}, c_{CO2})$ associated to CO2 simulation models. We adjust graph size by altering the number of internal nodes in each sub-graph, e.g., in the habitat sub-graph, the internal nodes are h_1 through h_{15}.

Exact Inference Method Similar to our analysis in Section 4.1, we explore the use of variable elimination, which is an exact inference method. Given the diagnosis evidence, we use exact inference to determine the state of all the other variables. We demonstrate that as the size of the graph increases, both in terms of the number of nodes in each sub-graph as well as the cardinality of each node, exact inference quickly becomes intractable.

Approximate Inference Methods We assume that spacecraft operation is a domain likely to require large graphs due to the complexity of the systems involved. This motivates the need for approximate inference like sampling and Markov Chain Monte Carlo (MCMC) methods. In particular, we explore the use of forward sampling, rejection sampling, and importance sampling.

Accuracy Measurement We explore the trade-off between accuracy and computational cost for each of the methods by evaluating the proportion of nodes with the correct assignment using exact inference as the ground truth to be recovered. By running this for 50 trials per setting, we also generate a 95% confidence interval on the accuracy.

Computational Cost Measurement For each of the approximate inference methods, we measure 1) the time taken for inference (in seconds) and 2) the memory usage (in MB). For each measurement, we explore two different settings understand the impact of scale and cardinality. In the first, we fix the cardinality of each node to 3 and scale the number of nodes in each sub-graph from 10 to 20. In

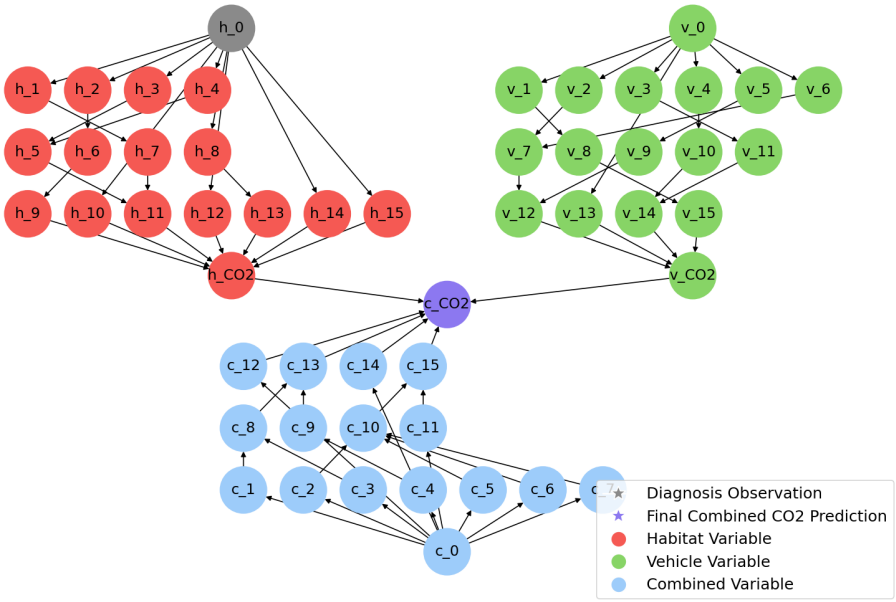


Figure 4: Randomly generated graph with fifteen internal nodes per module (h: habitat, v: vehicle, c: combined)

the second, we fix the number of nodes in each sub-graph to 5 and scale the cardinality of each node from 2 to 9.

5 Results and Evaluation

5.1 Part 1: Simulation Accuracy

We implement three linear regression models. Their parameters are made dependent using a graphical model for which we assume the conditional probability tables. The simulation outcomes that would be seen by the crew onboard the habitat are shown in Fig.5.

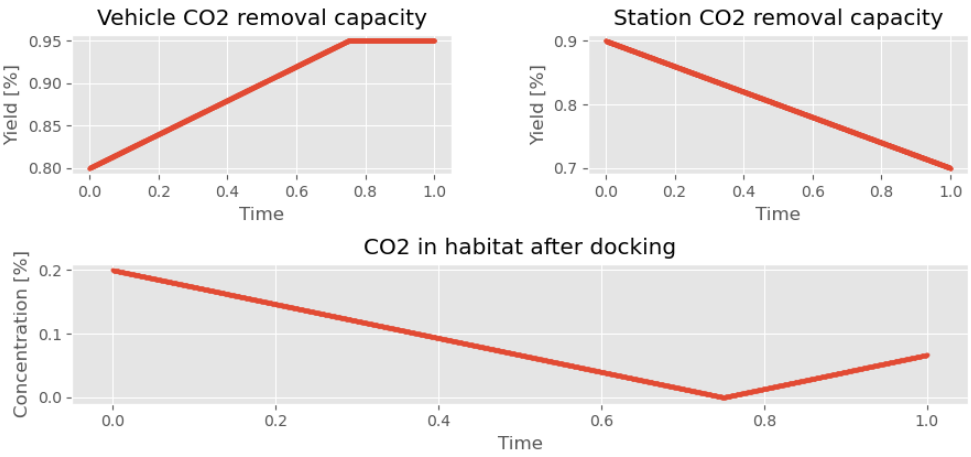


Figure 5: Simulation results

To evaluate our approach we measured the accuracy of the final simulated output, i.e., CO2 in habitat after docking, by computing the mean squared error between the estimated value $\hat{Y}_d$ and

Table 1: Average mean square error for 1,000 trials, the final simulation output is in the last column

	Habitat model	Vehicle model	**Docked model**
Random Baseline	0.005254	0.031046	**0.006126**
Marginal Baseline	0.002120	0.015604	**0.003708**
PGM	0.000100	0.015604	**0.003553**

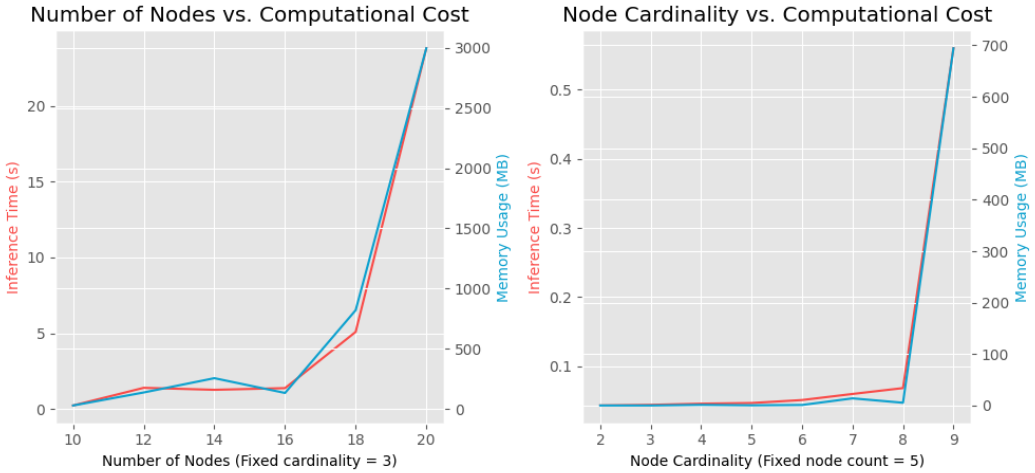


Figure 6: Exact inference (variable elimination)

the ground truth value Y_d. We implemented three baselines which all have different levels of integration between the model parameters. Each model can be assigned one of two parameters: Habitat model: $\{\theta_{h,ok}, \theta_{h,ko}\}$, Vehicle model: $\{\theta_{v,ok}, \theta_{v,ko}\}$, Docked model: $\{\theta_{d,ok}, \theta_{d,ko}\}$. The following approaches describe how these parameters are being selected for each model:

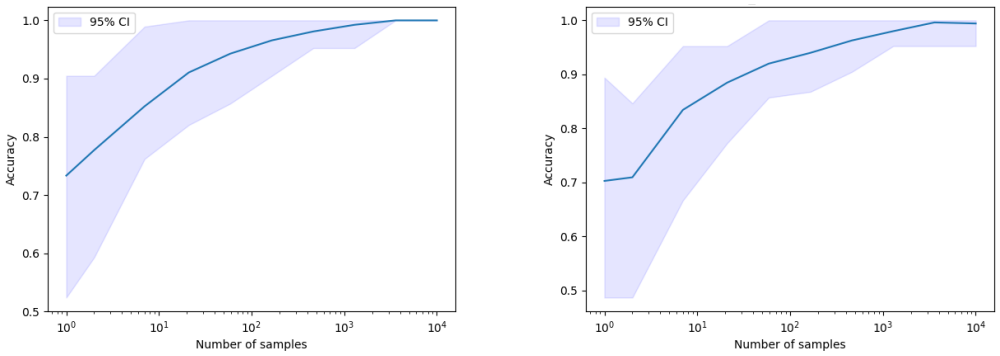
1. **Random Baseline**: Parameters are selected randomly
2. **Marginal Baseline**: Parameters are selected according to marginal probabilities, i.e., for each model i, pick $argmax_{\theta_i}[P(\theta_{i,ok}), P(\theta_{i,ko})]$.
3. **PGM**: Parameters are selected using belief propagation as described in Fig.3.

We demonstrate that encoding domain knowledge according to the baselines is too simplistic and that accounting for model parameter dependencies leads to lower mean squared error. The results are shown in Table 1.

5.2 Part 2: Computational Cost

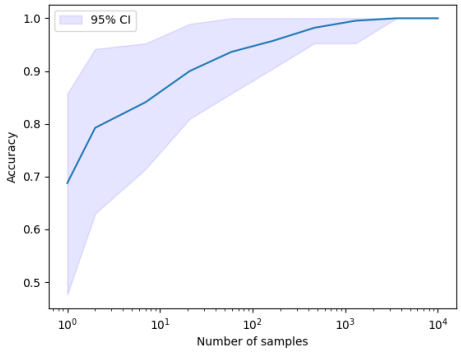
Exact Inference The computational cost of exact inference increases rapidly as we processed graphs with a larger number of nodes or a larger cardinality (see Figure 6). Memory usage was the limiting factor when increasing the number of nodes beyond 18 with a fixed cardinality of 3, or increasing the cardinality beyond 8 with a fixed node count of 5. Limiting the number of models and their parameter space in this way is assumed to be too restrictive for space operations. In the next section, we investigate approximate inference methods to increase the range of possible simulations that can be performed onboard a spacecraft.

Approximate Inference Approximate inference methods are sampling approaches expected to be faster but less accurate than exact inference. We selected three approximate inference methods namely forward, importance, and rejection sampling as these were available in the pgmpy package. Their accuracy was measured by counting the number of correct predictions, i.e., predictions that match the exact inference algorithm. We use a PGM of 10 internal nodes and a cardinality of 3 as a heuristic to approximate the number of samples needed to achieve approximately 95% accuracy



(a) Forward sampling

(b) Importance sampling



(c) Rejection sampling

Figure 7: Accuracy of approximate inference methods

which we found to be 1000. In the following, we, therefore, measured the computational cost of these methods by generating 1000 samples. The results of the computational costs for the largest node count and cardinality are summarized in Table 2.

The computational costs of the approximate methods are significantly more tractable as compared to that of the exact inference methods. The three methods we explore, forward sampling, importance sampling, and rejection sampling, generally display similar trends in both the amount of memory usage, as well as in inference time.

Rejection sampling seems to have a higher memory usage than the other methods when we scale the number of nodes in each sub-graph, but a lower memory usage when we scale the cardinality. It also seems to have the highest inference times compared to forward and importance sampling, which seems reasonable, since rejection sampling is likely to generate a large number of samples that do not get accepted.

We then use forward sampling to investigate the graph size from which computational cost starts being an issue with an approximate inference method. These experiments show that inference time is now the limiting factor as opposed to memory for exact inference. When setting the cardinality to three, the number of nodes per sub-graph can reach 49 within 184 seconds and 96 MB. When setting the number of nodes per sub-graph to 5, the cardinality can reach 16 within 242 seconds and 97 MB.

6 Discussion

6.1 Key Contributions

In our work, we demonstrate the benefit of PGMs in the domain of multi-model calibration for onboard simulations in deep space habitats. With the use of synthetic data simulations, we show how the use of belief propagation in PGMs leads to higher accuracy in predicting the final simulated output compared to a commonsense and random baseline. Finally, we demonstrate that exact inference

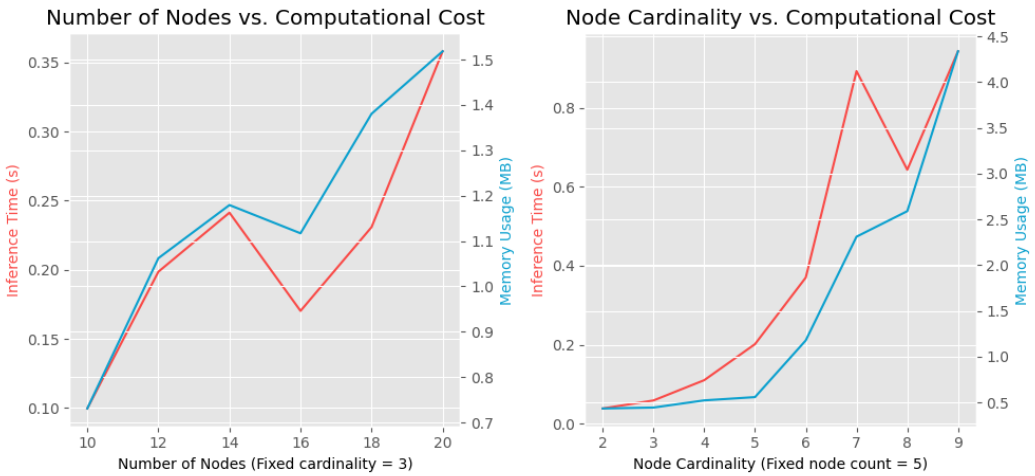


Figure 8: Forward sampling

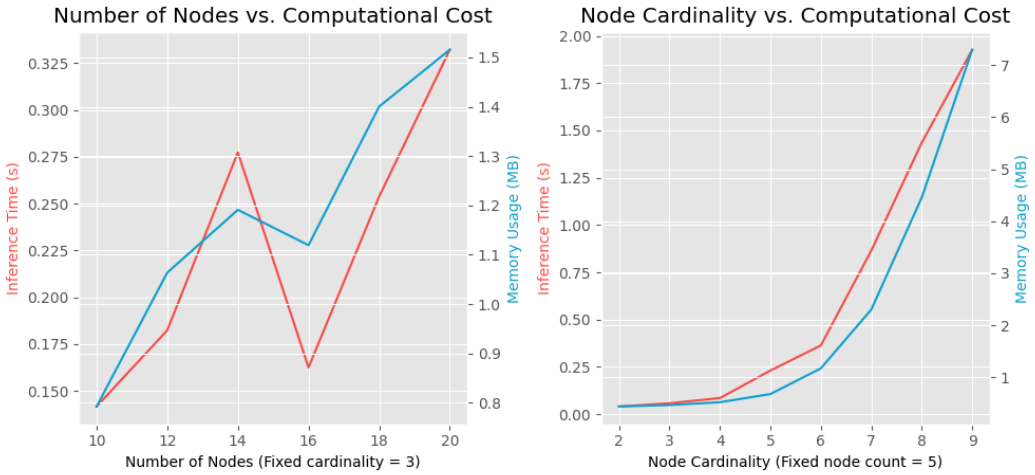


Figure 9: Importance sampling

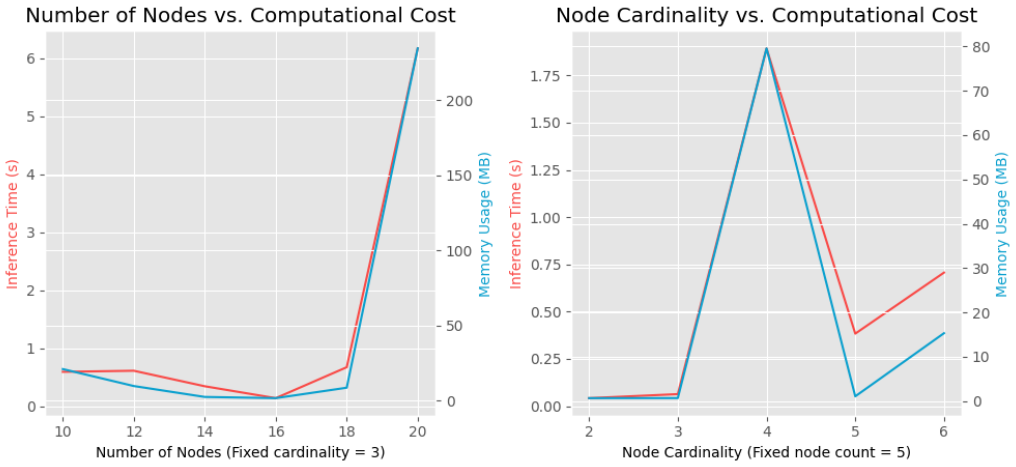


Figure 10: Rejection sampling

Table 2: Computational costs of the largest graphs

Sub-graph	Performance	Exact inference	Approximate inference		
		Variable elimination	Forward	Importance	Rejection
Node: 20 Cardinality: 3	Memory (MB)	3000	1.5	1.5	250
	Time (s)	25	0.35	0.325	6
Node: 5 Cardinality: 9	Memory (MB)	700	4.4	7	15
	Time (s)	0.6	0.9	1.9	0.75

quickly becomes intractable as the graph becomes larger, and how different approximate inference methods can help mitigate the computational cost by trading off accuracy.

6.2 Challenges

Computational cost limitations While approximate inference methods increase the range of possible graphs that can be processed, their computational cost still increase exponentially with the size of the graph. Studying how to best encode CO_2 removal domain knowledge into a PGM may be a promising approach to address this issue, i.e., identifying which simulation models are the most relevant to be incorporated in the PGM and with which parameter cardinality.

MCMC Methods When running MCMC methods using the pgmpy framework, we observe a very long inference time which is surprising for an approximate inference method. We noticed that similar implementation issues had been raised on the library's Github repository and we suspect these are related to the problem we encountered. After attempting to fix it, we realized we would need a deeper understanding of the library to investigate further, which we were not able to accomplish within the time frame of this project.

Reproducability of Results The use of randomly generated graphs made it difficult to get consistent results. Some of the algorithms like rejection sampling also had very high variance between simulations. As a next step, we would like to modify our simulation to have a more controlled environment, so we can remove randomness across trials whenever possible, and better quantify the tradeoff between accuracy and computation cost in approximate inference methods.

7 Teammates and Work Division

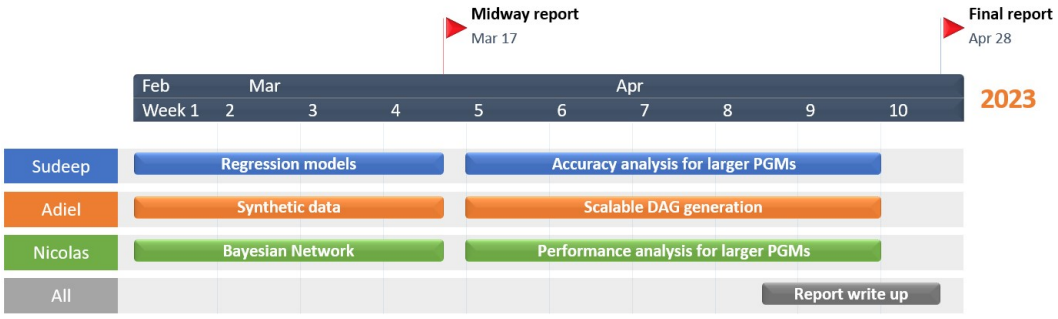


Figure 11: Timeline

8 Access to Code

Our code can be found at the following URL: **https://github.com/afelsen/pgm-calibration**. A readme file is provided and outlines how to run the script. Note that the GitHub repository is private, and our TA Fan Pu Zeng has been added as a collaborator.

References

[1] Afsaneh Kaghazchi, S. Mehdy Hashemy Shahdany, and Abbas Roozbahani. Simulation and evaluation of agricultural water distribution and delivery systems with a Hybrid Bayesian network model. *Agricultural Water Management*, 245, February 2021.

[2] Siyu Tao, Anton van Beek, Daniel W. Apley, and Wei Chen. Multi-Model Bayesian Optimization for Simulation-Based Design. *Journal of Mechanical Design*, 143(11), May 2021.

[3] Shankar Sankararaman and Sankaran Mahadevan. Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. *Reliability Engineering & System Safety*, 138:194–209, June 2015.

[4] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, July 2009.

[5] Ankur Ankan and Abinash Panda. Pgmpy: Probabilistic Graphical Models using Python. In *Python in Science Conference*, pages 6–11, Austin, Texas, 2015.

[6] livion. DAG_Generator. `https://github.com/Livioni/DAG_Generator`, April 2023.