A DESIGN STRATEGIES AND BEST PRACTICES FOR NGMS

We share some of the design strategies and best practices that we developed while working with NGMs here. This is to give insights to the readers on our approach and help them narrow down the architecture choices of NGMs for applying to their data. We hope that sharing our thought process and findings here will foster more transparency, adoption and help identify potential improvements to facilitate the advancement of research in this direction.

- *Choices for the structure loss function.* We narrowed down the loss function choice to Hadamard loss $\|(\Pi_i|W_i|) * S^c\|$ vs square loss $\|(\Pi_i|W_i|) - S\|^2$. We also experimented with various choices of Lagrangian penalties for the structure loss. We found that ℓ_2 worked better in most cases. Our conclusion was to use Hadamard loss with either ℓ_1 vs ℓ_2 penalty.
- *Strategies for λ initialization.* (I) Keep it fixed to balance between the initial regression loss and structure loss. We utilize the loss balance technique mentioned in Rajbhandari et al. (2019). (II) Use the proximal initialization technique clubbed with increasing λ value as described in Alg. 1. Both the techniques seem to work well, although (I) is simpler to implement and gives equivalent results.
- *Selecting width and depth of the neural view.* We start with hidden layer size $H = 2 \times |\text{In}|$ twice the input dimension. Then based on the regression and structure loss values, we decide whether to go deeper or have a larger number of units. In our experience, increasing the number of layers helps in reducing the regression loss while increasing the hidden layer dimensions works well to optimize for the structure loss.
- *Choices of non-linearity.* For the MLP in the neural view, we played around with multiple choices of non-linearities. We ended up using ReLU, although $\tanh$ gave similar results.
- *Handling imbalanced data.* NGMs can also be adapted to utilize the existing imbalanced data handling techniques Chawla et al. (2002); Shrivastava et al. (2015); Bhattacharya et al. (2017) which improved results in our experience.
- *Calculate upper bound on regression loss.* Try fitting NGM by assuming fully connected graph to give the most flexibility to regression. This way we get an upper bound on the best optimization results on just the regression loss. This helps to select the depth and dimensions of MLPs required when the sparser structure is imposed.
- *Convergence of loss function.* In our quest to figure out a way to always get good convergence on both the losses (regression & structure), we tried out various approaches. (I) Jointly optimize both the loss functions with a weight balancing term λ, Eq. 2. (II) We tested out an Alternating Method of Multipliers (ADMM) based optimization that alternately optimizes for the structure loss and regression loss. (III) We also ran a proximal gradient descent approach which is sometimes suitable for loss with ℓ_1 regularization terms. Choice (I) turned out to be effective with reasonable λ values.

In the current state, it can be tedious to optimize NGMs and needs decent amount of experimentation. It is a learning experience for us as well and we are always on a lookout to learn new techniques from the research community.

B INFANT MORTALITY ANALYSIS

We created an NGM to model infant mortality data. The dataset is based on CDC Birth Cohort Linked Birth – Infant Death Data Files of Health et al.. It describes pregnancy and birth variables for all live births in the U.S. together with an indication of an infant's death before the first birthday. We used the data for 2015 (latest available), which includes information about 3,988,733 live births in the US during 2015 calendar year.

We recovered the graph strucure of the dataset using `uGLAD` (Shrivastava et al., 2022a) and using Bayesian network package `bnlearn` (Scutari, 2010) with Tabu search and AIC score. The graphs are shown in Fig. 7 and 6 respectively. Since `bnlearn` does not support networks containing both continuous and discrete variables, all variables were converted to categorical for `bnlearn` structure learning and inference. In contrast, `uGLAD` and NGMs are both equipped to work with mixed types of variables and were trained on the dataset prior to conversion.

Both graphs show similar sets of clusters with high connectivity within each cluster:

- describing both parents' race and ethnicity (`mrace` and `frace` variables),

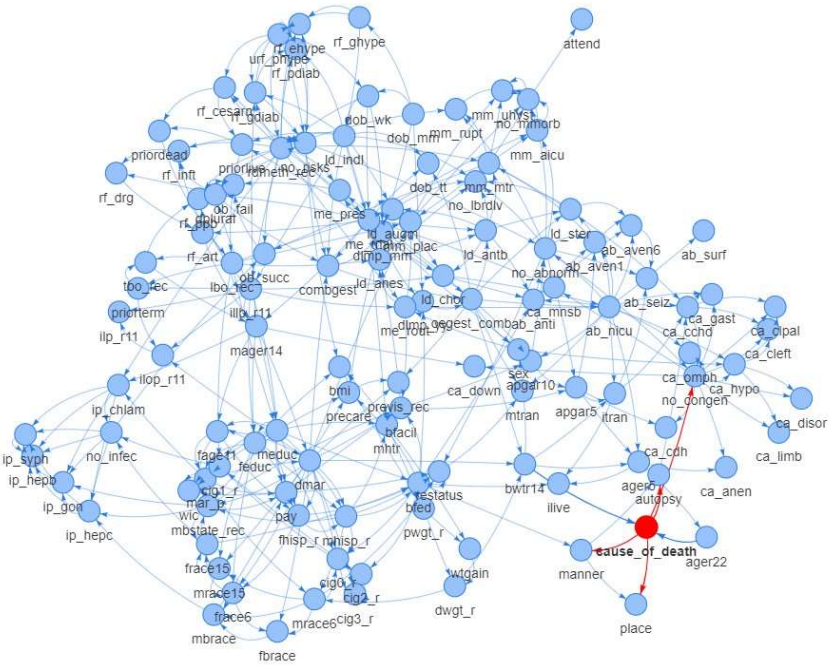

Figure 6: The Bayesian network graph learned using score-based method for the Infant Mortality 2015 data.

- related to mother's bmi, height (`mhtr`) and weight, both pre-pregnancy (`pwgt_r`) and at delivery (`dwgt_r`),
- consisting of maternal morbidity variables marked with `mm` prefix (e.g., unplanned hysterectomy),
- showing pregnancy related complications such as hypertension and diabetes (variables prefixed with `rf` and `urf`),
- consisting of variables related to parents' STD infections (`ip` prefix),
- related to delivery complications and interventions (variables prefixed with `ld`),
- showing interventions after delivery (`ab` prefix) such as ventilation or neonatal ICU,
- describing congenital anomalies diagnosed in the infant at the time of birth (variables prefixed with `ca`),
- related to infant's death: age at death, place, autopsy, manner, etc.

Apart from these clusters, there are a few highly connected variables in both graphs: gestational age (`combgest` and `oegest`), delivery route (`rdmeth_rec`), Apgar score, type of insurance (`pay`), parents' ages (`fage` and `mage` variables), birth order (`tbo` and `lbo`), and prenatal care.

With all these similarities, however, the total number of edges varies greatly between the two graphs and the number of edges unique to each graph outnumbers the number of edges the two graphs have in common (see Figure 8).One reason for the differences lies in the continuous-to-categorical conversion performed prior to Bayesian network structure discovery and training. The two graph recovery algorithms are very different in both algorithmic approach and objective function. We plan to further explore NGMs' sensitivity to input graph recovery algorithm in future work.

Infant mortality dataset is particularly challenging, since cases of infant death during the first year of life are (thankfully) rare. Thus, any queries concerning such low probability events are hard to estimate with accuracy.

NGM-generic architecture: Since we have mixed input data type, real and categorical data, we utilize the NGM-generic architecture as shown in Fig. 3. We consider a 2-layer neural view with hidden layer dimension as $H = 1000$. The categorical input was converted to its one-hot vector representation and added to the real features which gave us roughly ~ 500 features as input. The neural view input from the encoder had the same dimension as input. Similarly, we maintained same

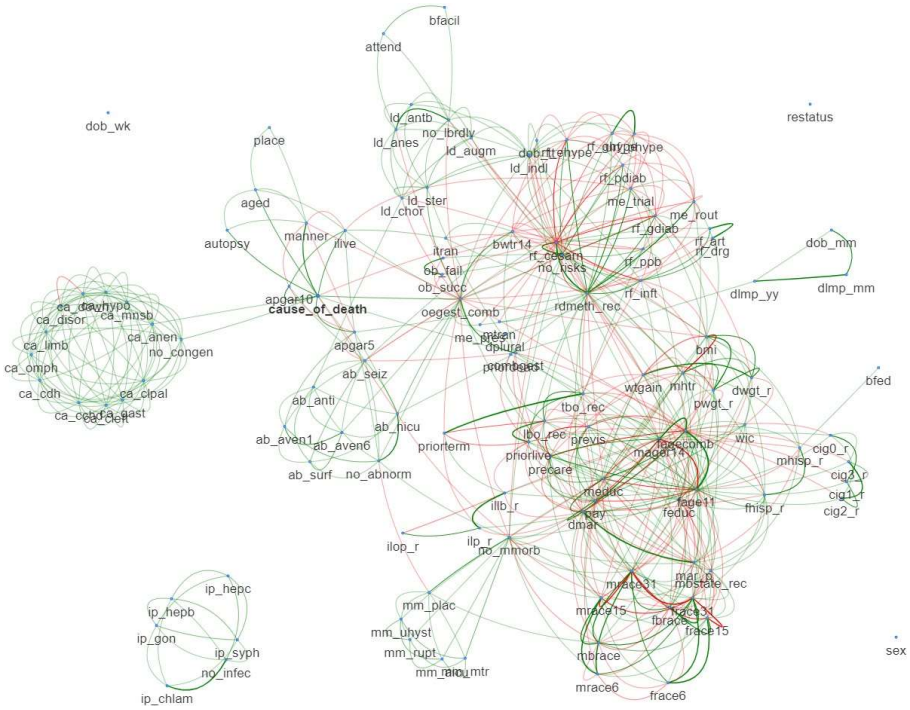

Figure 7: The CI graph recovered by uGLAD for the Infant Mortality 2015 data.

dimension from the neural view output to the decoder output. The entire NGM-generic parameters were learned by minimizing the eq. 4 using the 'adam' optimizer.

Sensitivity to the input graph: To study the effect of different graph structures on NGMs, we train separate models on the Bayesian Network graph (after moralizing) and the CI graph from uGLAD given in Fig. 6 & 7 respectively. We plot the dependency functions between pairs of nodes based on the common and unique edges found in the comparison plots of Fig. 8. For each pair of features, say (f_1, f_2), the dependency function is obtained by running inference $P(f_1|f_2)$ by varying the value of f_2 over its range as shown in Fig. 9.

Comparing NGM inference in models with different input graphs shows some interesting patterns:

- Strong positive correlation of mother's delivery weight (dwgt_r) with pre-pregnancy weight (pwgt_r) is shown in both models.
- Similarly, both models show that married mothers (dmar= 1) are likely to gain more weight than unmarried (dmar= 2).
- Both models agree that women with high BMI tend to gain less weight during their pregnancies than women with low BMI.
- A discrepancy appears in cases of the dependence of both BMI and weight gain during pregnancy on mother's height (mhtr). According to the NGM trained with a BN graph, higher weight gain and higher BMI are more likely for tall women, while the CI-trained NGM shows the opposite.
- Possibly the most interesting are the graphs showing the dependence of the timing a women starts prenatal care (precare specifies the month of pregnancy when prenatal care starts) on the type of insurance she carries. For both models, Medicaid (1) and private insurance (2) mean early start of care and there is a sharp increase (delay in prenatal care start) for self-pay (3) and Indian Health Service (4). Models disagree to some extent on less common types of insurance (military, government, other, unknown).

Our experiments on infant mortality dataset demonstrate usefulness of NGMs to model complex mixed-input real-world domains. We are currently running more experiments designed to capture more information on NGMs' sensitivity to input graph recovery algorithm and inference accuracy.

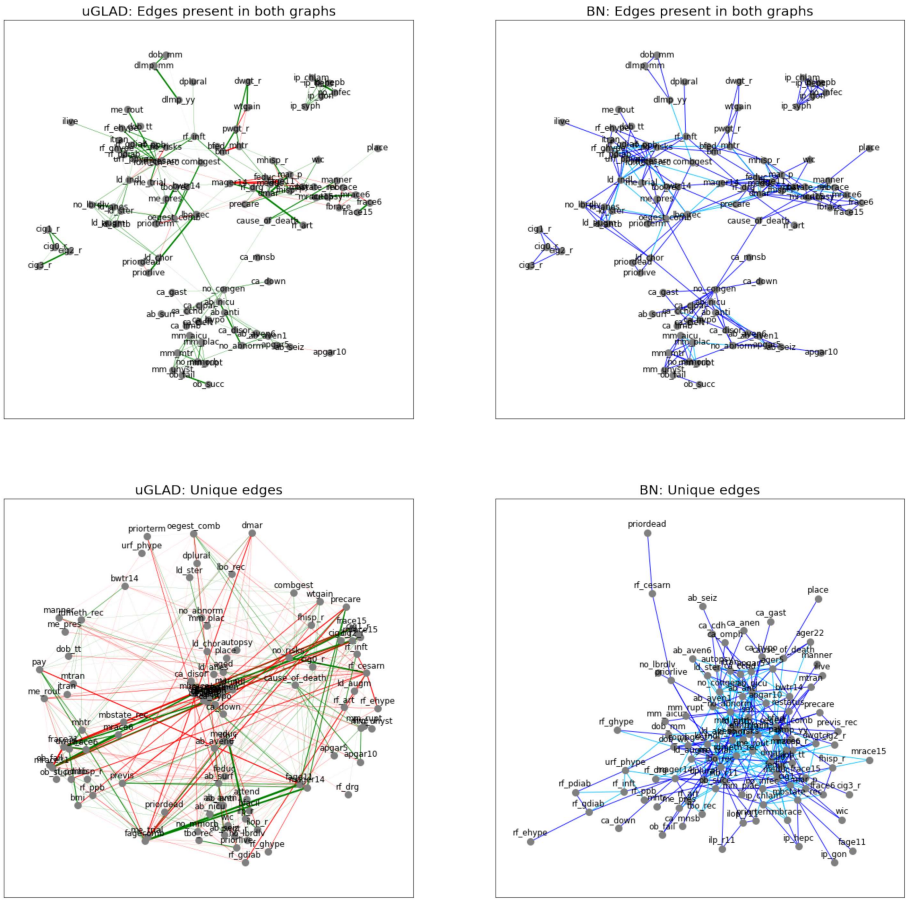

Figure 8: Comparing the graphs recovered by uGLAD and Bayesian Network recovery package (Scutari, 2010) after moralization (moralized edges are denoted by 'skyblue').

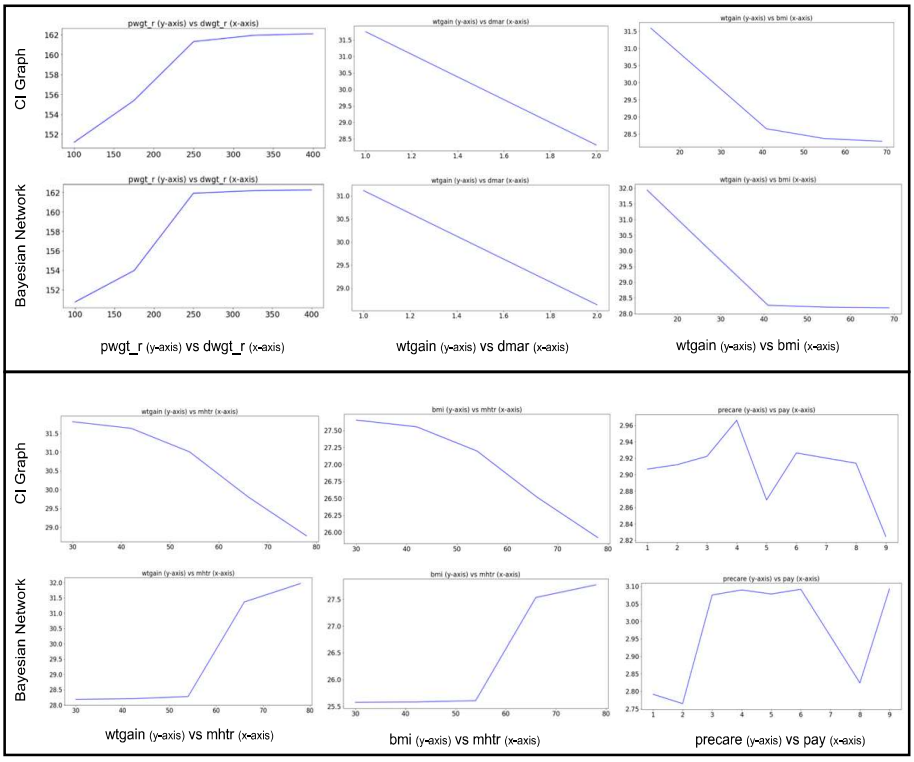

Figure 9: **Evaluating effects of varying input graphs for learning NGMs.** Comparing the NGM dependency plots recovered by using Bayesian Network graph vs the CI graph obtained by running `uGLAD` . Similar architecture of NGMs were chosen and the data preprocessing was also kept as alike as possible. For the feature pairs in the top box, the trends match for both the graphs, while in the bottom box the dependency plots differ. We observed that the dependency trends discovered by the NGM trained on the CI graph matches the correlation of the CI graph. Common edges present in both the graphs [(pwgt-r, dwgt-r), (wtgain, mhtr), (bmi, mhtr), (precare, pay)], edges only present in CI graph [(wtgain, dmar), (wtgain, bmi)]. It is interesting to observe that even for some common edges, eg. (wtgain, mhtr), that represents strong direct dependence between the features, the trends can still differ significantly. This highlights the importance of the input graph structure chosen to train NGMs.