Sequence	Embedding Loss	Neural Network Prediction Loss	Symbolic Regression Prediction Loss
2 Community	41.67	47.91	35.11
Long Tail	2.408	1.193	2.524
3 Community	35.43	35.50	35.57

FIGURE 1. A summary of the mean loss of each prediction at the fifth time step. The embedding loss comes from the true target node and loss is only introduced by the SVD. The neural network and symbolic prediction loss comes from the RDPG being reconstructed but with the position of the target node replaced with the location of the respective prediction. Loss is calculated as the total number of incorrectly predicted edges. That is an edge not being present when it should as well as the reverse.

1. Results and Discussion

1.1. **Network Prediction.** To compare our models, we look at the loss of the predictions. This includes the loss from the SVD.

To find the approximation of the edges of node i, ${}_tA_{i,\cdot}$. We use:

$$ {}_tA_{i,\cdot} \approxeq (\bar{p})_t\hat{A} \tag{1}$$

Where $\bar{p}$ is the location of the node i in the embedding at time t.

To find calculate the loss of our prediction at time t, we use:

$$ L = ||{}_tA_{i,\cdot} - (\bar{p})_t\hat{A}||_1 \tag{2}$$

When we do this for the SVD, NN, and symbolic regression, we get Figure 1.

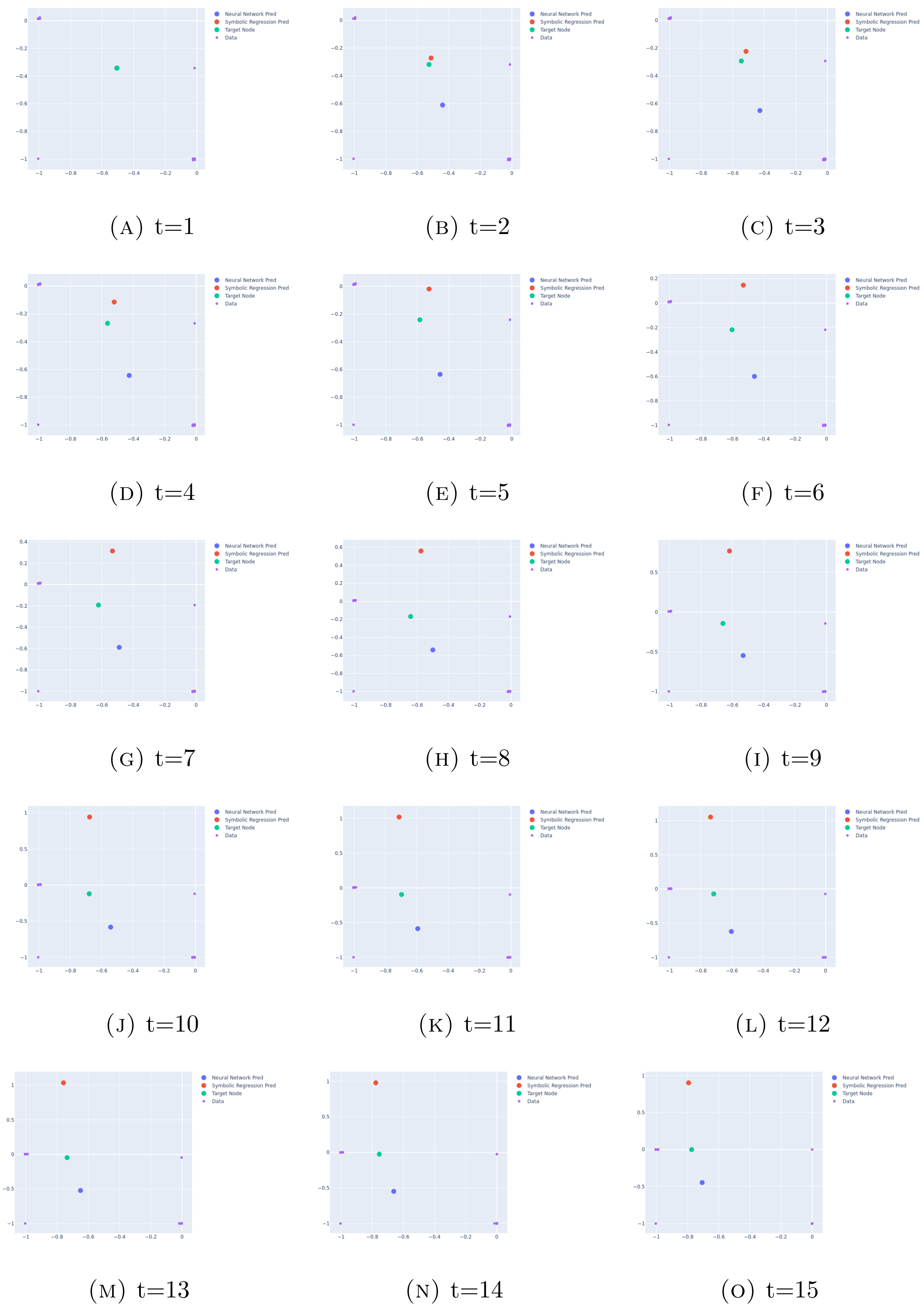
The loss of one system should not be compared with another. That is the 2 community losses should not be compared to the long tail. This is because the long tail system will only ever have one edge to predict, whereas the 2 community will have 50 edges that need to be predicted.

Instead, we may compare predictions within the same system. However, we do not see much difference between these predictions, and so we plot our predictions at each time step to further understand the behavior of our framework.

For each of these systems, we took embedded the temporal network in two dimensions at each time step. The temporal embeddings were then divided into a training set of 20 and a testing set of 15. The output of the trained neural networks was then used to train a symbolic regression model that could use a simple set of addition, subtraction, division, and multiplication.

In fig Figure 2, Figure 3,Figure 4 the green points are the embedded coordinates of the node in each of the communities. Each cluster is one separate community. The orange point is the true coordinate of the embedded target node at each time step. The blue point is our model prediction from the true target node at the first time step. As the time progresses, we see the true node move from one community cluster to the other. This is expected from the embedding; the target node starts as very similar to the first community (as it has many connections with nodes in that community) and as time progresses, it gradually becomes less and less similar to the first community and more similar to the second (as the edges between the target node and the first community are replaced with edges to the second community). As such, we see the target node move towards the second community.

1.2. **2 Community.** We see the neural network model quickly jumps to around $(-0.5, -0.5)$ where it sits for the entire simulation. This happened across many training periods; either a few time steps after the training period or right at the end



(A) t=1 (B) t=2 (C) t=3

(D) t=4 (E) t=5 (F) t=6

(G) t=7 (H) t=8 (I) t=9

(J) t=10 (K) t=11 (L) t=12

(M) t=13 (N) t=14 (O) t=15

FIGURE 2. 2 Community test series. This series shows the comparison of the neural network model, the symbolic regression model trained on the neural network model, and the true solution of the two community system.

of it, the neural network model moved towards a stable point where it remained for the test period.

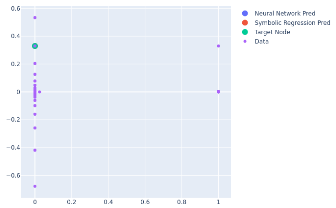
In contrast, the symbolic regression predicted the location of the target node very well at the beginning of the test period, but after about 5 steps starts to drift away.

The poor predictive behaviour of the neural network especially may be attributed to this being a symmetrical problem. That is the same set of distances as input into the neural network, the target node needs to move away from its nearest neighbours in the first half, then move towards its nearest neighbours in the second half. Potential ways of breaking this symmetry may be to include information of the previous time step, or to include the absolute position of the target node as input into the neural network.

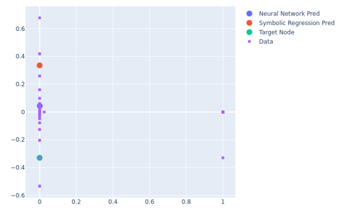
Notably, using the small neural network model to train the symbolic regression gave much better predictions than the large neural network**??**.

1.3. **Long Tail.** In fig Figure 3 we see the true embedding of the target node jump from one arm to the other at each time step. It is clear that neither the symbolic regression model nor the neural network model capture this movement. We also see that the symbolic regression model maintains a relatively stable position when compared to the neural network model.

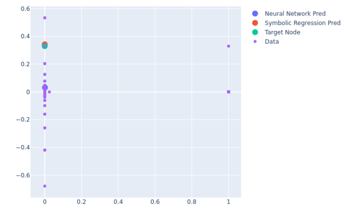
We see that neither the neural network model capture the movement of the target node to any real extent. One reason for this may again be an issue of symmetrical input. The distances from the k nearest neighbours to the target node do not seem to change much as the target node jumps from one arm to the other, but the neural network somehow needs to learn to jump left then right at alternating time steps. The solutions to this would be the same as for the two community system.



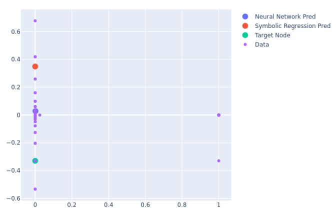
(A) t=1



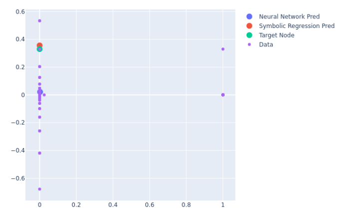
(B) t=2



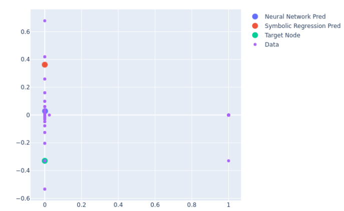
(C) t=3



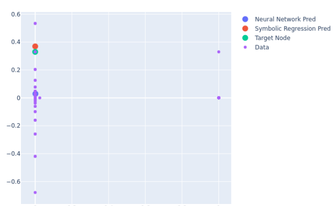
(D) t=4



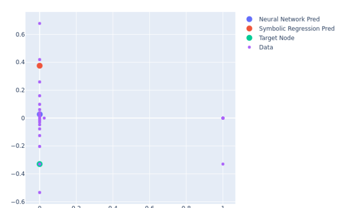
(E) t=5



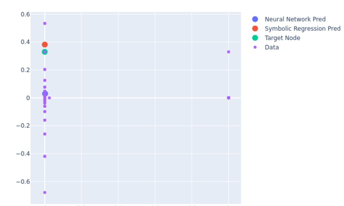
(F) t=6



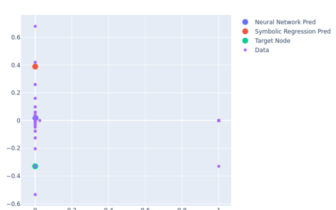
(G) t=7



(H) t=8

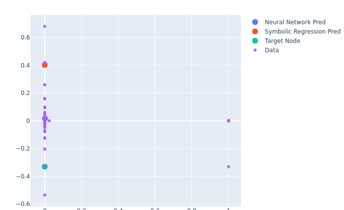


(I) t=9

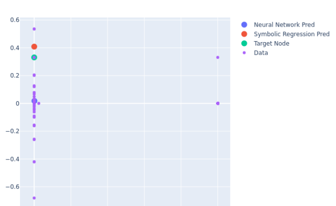


(J) t=10

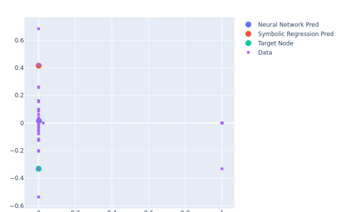
(K) t=11



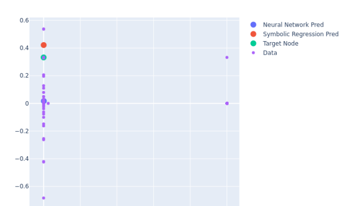
(L) t=12



(M) t=13

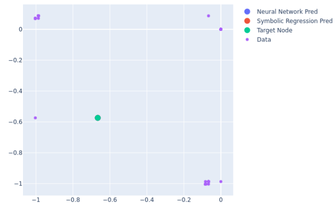


(N) t=14

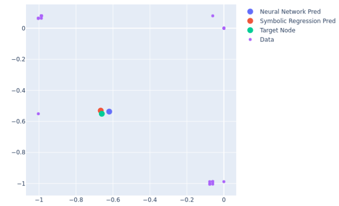


(O) t=15

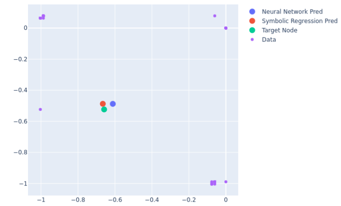
FIGURE 3. Long Tail test series. This series shows the comparison of the neural network model, the symbolic regression model trained on the neural network model, and the true solution of the long tail system.



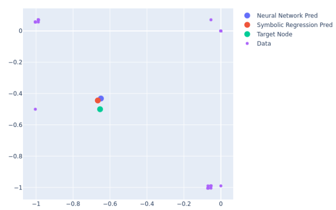
(A) t=1



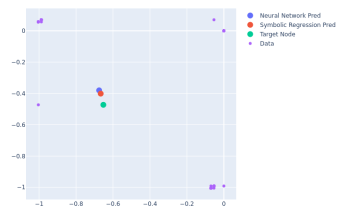
(B) t=2



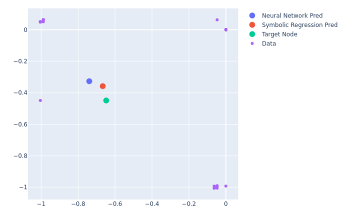
(C) t=3



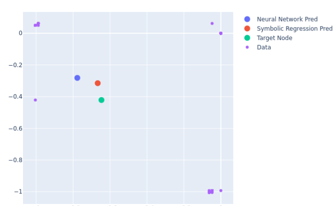
(D) t=4



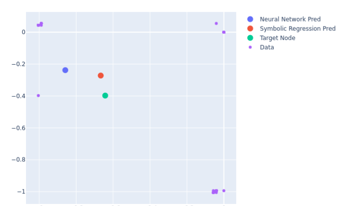
(E) t=5



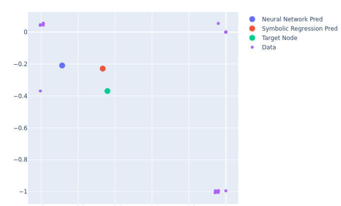
(F) t=6



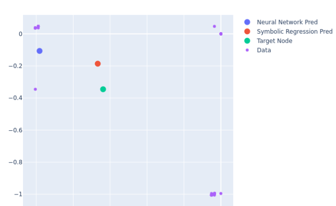
(G) t=7



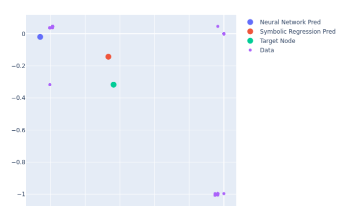
(H) t=8



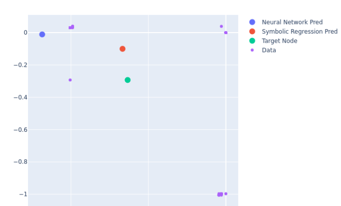
(I) t=9



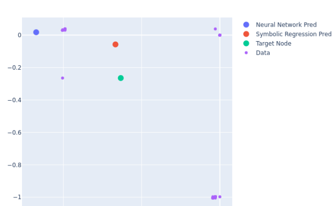
(J) t=10



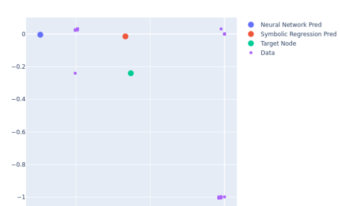
(K) t=11



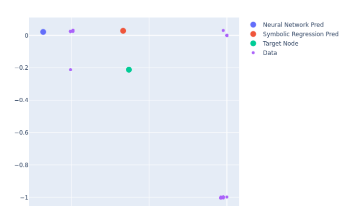
(L) t=12



(M) t=13



(N) t=14



(O) t=15

FIGURE 4. 3 community test series. This series shows the comparison of the neural network model, the symbolic regression model trained on the neural network model, and the true solution of the 3 community system.

1.4. **Three Community.** In fig Figure 4 we again, as the edges between the community with the fewest edges are removed, we see the target node move away from it in the embedding and towards the other two clusters of points (communities).

For the first ten time steps, we see similar behaviour between the neural network model and the neural network model. However, around the eleventh time step, the symbolic regression model begins to diverge from the neural network. The symbolic regression moves back towards the target node towards the end. This could indicate that the symbolic regression is a somewhat more robust model when extrapolating, but more testing should be done to corroborate this.

Notably the symbolic regression trained on the large neural network performed extremely well, while the predictions of the large neural network model itself did not seem to make any significant improvement**??**. This may indicate that...XXXXX

In the neural network model, we see the prediction remain at close to its initial location for the duration of the test. This seems to indicate an attractor in the model.

In contrast, we see that the symbolic regression model quickly moves ahead of the true location where it remains for the duration of the test.

1.5. **Summary.** We see a large difference in the training predictions between the 2 and 3 community systems. In the 2 community system, even in the training period, the predictions tended to wander and be less accurate. Whereas the 3 community training predictions remained very close to the true target node. The movement of the target node in the embedding was very similar between the two systems, and so the difference seems to be that the 2 community has symmetrical inputs.

These were trained on a relatively small neural network (one hidden layer of 64 nodes), when training this small network one of the issues we encountered was that the weights would become stuck in local minima. To manage this we found that simulate annealing was helpful. When training on a larger network (4 hidden layers (64,8,8,8)) this does not seem to be as much of a problem.

2. Conclusion

In this paper, we proposed a novel framework for modelling temporal networks. We then tested this framework on three types of small, synthetic network sequences. Throughout these tests the neural network performed generally more poorly than the symbolic regression model, with the large three community sequence being especially notable due to th symbolic regression remaining so close to the target node throughout the test period.