

Deep learning resilience inference for complex networked systems

摘要

弹性，即在故障和错误中维持基本功能的能力，对于复杂的网络系统至关重要。大多数分析方法依赖于节点活动动态的预定义方程并简化对网络拓扑的假设，限制了它们对现实世界系统的适用性。该文章提出了 ResInf，这是一个集成了 Transformer 和图神经网络的深度学习框架，可以直接从观测数据推断弹性。ResInf 在不简化假设的情况下学习节点活动动态和网络拓扑的表示，从而实现准确的弹性推理和低维可视化。实验结果表明，ResInf 的性能显著，还可以推广到看不见的拓扑和动力学，并在观测干扰的情况下保持稳健的性能。

关键词：弹性；复杂网络；图神经网络

1 引言

复杂网络是由数量巨大的节点和节点之间错综复杂的关系共同构成的网络结构，如基因调控网络等。基于网络中各节点之间相互影响，复杂网络的弹性定义为网络在发生错误、故障和环境变化时调整其活动以保持其基本功能的能力。现有研究 [1, 2, 4, 7, 8, 10, 11] 已经表明，复杂网络弹性丧失将对网络产生巨大的负面影响，例如，在生态系统中，鲁棒性的缺失将导致生态链失衡最终使大量物种灭绝。为了避免网络弹性丧失带来严重的后果，我们需要有效的方法判断并预测复杂网络当前状态。

网络的动态演化过程由如下微分方程表示，其中 $F(x_i)$ 代表网络节点 i 自身的动力学公式， A_{ij} 为交互权重矩阵，表示节点 i 和 j 之间的交互强度， $G(x_i, x_j)$ 代表节点 i 和 j 之间的交互动力学公式。

$$\frac{dx_i}{dt} = F(x_i) + \sum_{j=1}^N A_{ij} G(x_i, x_j), \quad (1)$$

当上述公式存在稳定的非零解形式时，复杂网络具有弹性，反之，如果各节点解趋向于零，表明网络丧失恢复平衡稳态的调节能力，不再具有弹性。由于复杂网络节点相互作用的具体数学形式难以确定，相关研究 [9, 16] 通过假设网络节点间动力学公式和网络拓扑形式简化问题，降低求解难度。例如，GBB [3] 框架聚焦于计算网络系统的弹性参数 β ，当且仅当弹性参数超过一定的临界阈值 β_s 时，网络系统被认定为有弹性。然而弹性参数 β 和临界阈值 β_s 的估计依赖于预定的假设，即节点动力学公式是线性的。相关实验表明，当该模型的前提假设遭到破坏时，模型的准确率受到挑战。为了突破预定义假设的限制，该文献引入深度学习方法促进对复杂网络弹性预测的研究。

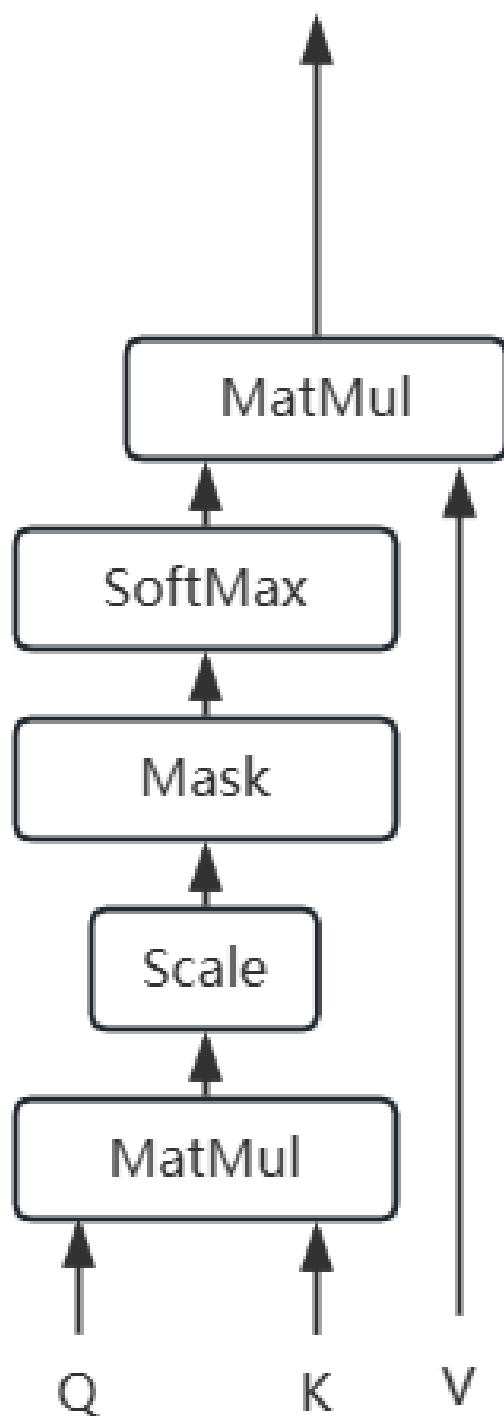


图 1. Transformer架构

2 相关工作

该文章有效地集成了Transformer框架和图神经网络(GNN)模型分别对复杂网络的动力学时序信息和网络拓扑结构进行学习并应用于系统弹性预测。

2.1 Transformer

Transformer模型 [5, 13–15, 17, 18]作为自然语言处理的序列到序列(seq2seq)模型，遵循编码器-解码器的结构，如图2所示，并结合位置编码保留序列的时序信息。

自注意力机制(Self-Attention)是Transformer模型的核心方法，如图1所示。首先接收编码器的输入或者上一个编码器层的输出，根据输入序列各数据之间的相关性确定对不同数据的信息的保留情况，计算相关性我们需要两个矩阵Q(查询)，K(键值)，两者相乘可以得到各数据相关性，将其进行一系列处理例如输入softmax函数得到最终的注意力得分，最后再与矩阵V(值)相乘从而从输入序列数据中提取信息，Q，K，V三个矩阵都由三个线性变换矩阵与输入相乘得到，三个线性变换矩阵由神经网络在训练中学习得出。一般而言，基于数据的复杂性，只学习一种类型的相关性不一定能提取到所有有用的信息，于是多头自注意力机制(Multi-head Self-attention)，通过学习多组Q，K，V矩阵得到多种相关性并将其以一定方式聚合，可以得到更好的效果。

Transformer编码器中含有数层编码层，每个编码层的结构相同，首先将注入位置编码的数据输入多头自注意力层得到输出，之后经过由残差层和归一化层组成的Add&Norm层，其中残差层将编码层的输入与其输出直接相加，使模型避免随着层数的增加模型效果难以提高的瓶颈限制，防止神经网络退化，同时还可以避免梯度消失的问题。归一化层主要将残差层之后的输出进行归一化操作，加快网络收敛的速率并且提高稳定性。经过上述操作后，数据进入Feed Forward层，共包含两层全连接层，第一层采用ReLU激活函数，第二层不设置激活函数。综上所述即为一层编码器层的操作，重复执行数次即可得到编码器的输出。将其作为输入进入解码器。

解码器与编码器相似，但在细节上存在一定差异：与编码器相比，每个解码器层共包含两个多头自注意力层，并且在第一个层次中，首先对输入解码器的数据即编码器的最终输出数据进行Mask操作，也就是将输出序列第i个数据其后数据进行遮蔽，避免在预测操作时超前得知后面的数据信息，但可以利用之前的数据信息，并且在实际输入解码器时，解码器在数据的首部添加一个特殊数据表示开始预测并得到首位数据的预测输出，以此类推进行预测，并在最后通过输出一个特殊的数据表示来结束预测。在训练过程中，我们可以使用Teacher Forcing方法来实现并行化操作，即在输入时，不使用上一个预测结果数据，而采用真实的数据输入。第一层输入Mask数据并遵循多头自注意力机制操作。第二层与编码器多头自注意力层类似，只是所采用的K，V矩阵有所变化，该层根据编码器的输出来计算得到两个矩阵，再根据同一解码器层内上一层多头自注意力机制的输出计算矩阵Q。接着经过数层解码器层的操作，将最后一层注意力层的输出通过全连接层计算并通过SoftMax函数激活，得到最终的预测输出序列。

该文章利用Transformer模型编码时序信息的能力对复杂网络中节点的动力学变化过程进行学习，从而得到合适的节点嵌入。

2.2 图神经网络

GNN [6, 12]是一类处理图结构数据并进行推理的深度学习方法。GNN通过聚合各个节点的邻居信息学习图数据的拓扑结构信息。对于一个具有t层的GNN，节点可以聚合距离自身t跳邻居内的信息。第t层的变换如公式所示：

$$h_i^{(t)} = \sum_{j \in N(i)} f(x_i, x_{(i,j)}^e, x_j, h_j^{(t-1)}), \quad (2)$$

其中 $h_i^{(t)}$ 代表节点i在图神经网络第t层的嵌入信息， $f(\cdot)$ 是一个参数函数，能够聚合节点i及其邻居节点j的交互信息。不同的聚合函数设计为不同的GNN模型。经过多层聚合后得到的节

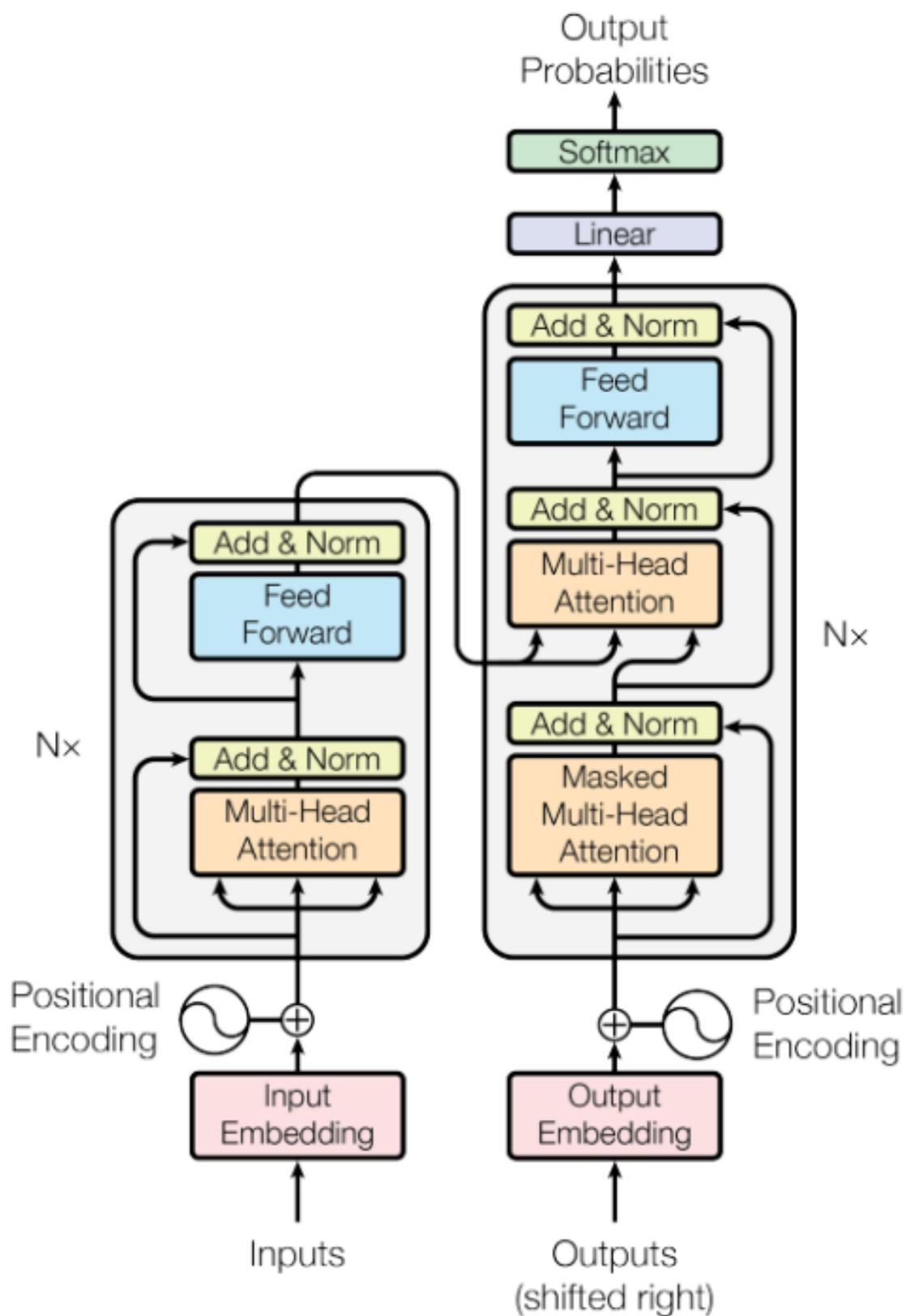


图 2. Transformer架构

点或图的高级表示用于不同的下游任务。在这项工作中，利用 GNN 来学习复杂网络中的拓扑信息，并引入一个虚拟节点表示全图的整体结构信息。

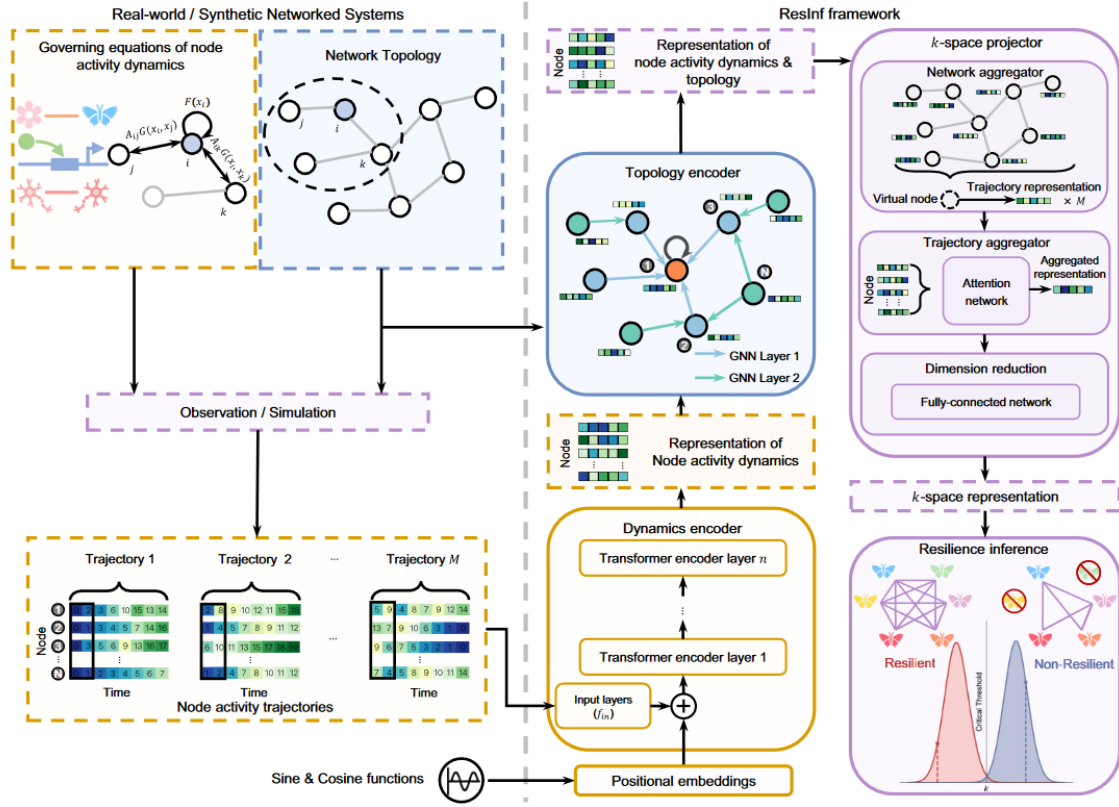


图 3. 方法示意图

3 本文方法

该文章提出了ResInf模型，整体框架如图3所示，包含动态编码器，拓扑编码器和K-空间投影三个部分，最终得到一维的结果输出作为对复杂网络弹性的判定，并且便于进一步的可视化对比。

3.1 动态编码器

ResInf模型利用Transformer网络作为动态编码器。无需输入任何先验知识，动态编码器直接并行学习多个节点的不同动态轨迹数据，并且为每个轨迹生成一个节点的动态活动表示从而捕捉复杂网络的底层动态。

首先将输入的轨迹数据使用多层感知机(MLP)嵌入到隐空间，并且添加位置嵌入标志每个时间点的节点活动信息的相对位置。之后通过一系列的多头自注意力机制、残差链接和归一化等操作得到动态编码器的输出，并且采用输出的最后一个时间点的嵌入作为最终节点活动动态表示。

3.2 拓扑编码器

该文章设计了一个具有多跳信息传递机制的GNN来模拟复杂网络中节点的多跳邻域之间的相互影响。给定网络的邻接矩阵 A 和消息传递算子 Φ ，可以得到第 l 层消息传递的公式：

$$Z_p^{(l)} = \tanh(Z_p^{(l-1)}W_F^{(l)}) + \tanh(\Phi Z_p^{(l-1)}W_G^{(l)}), \quad (3)$$

表 1. 数据集相关信息

数据集	平均节点数	平均边数	轨迹长度	平均节点轨迹数
Micro	11	55	11	1
Mutualistic	19	44	11	100
Gene Regulatory	22	54	11	100
Neuronal	21	55	11	100

其中 $Z_p^{(l)}$ 表示第 p 条轨迹经过 l 层消息传递后的节点表示, $W_F^{(l)}$ 和 $W_G^{(l)}$ 为该 GNN 层的可训练参数矩阵。由于弹性是一个网络全局具有的属性, 该模块引入了一个虚拟节点来融合所有节点的信息。

3.3 K-空间投影

ResInf 最后设计了 K-空间投影器得到整个模型用于弹性预测的结果。首先针对之前模块的输出, 投影器引入了注意力模块整合所有轨迹的嵌入信息, 并且使用虚拟节点的最终表示作为复杂网络的潜在表示。之后, 采用 MLP 模型作为复杂网络的弹性预测器输出一维空间的结果值, 通过映射后与定义的临界阈值对比从而判定网络的弹性。

4 复现细节

4.1 与已有开源代码对比

该文章已有针对各个数据集的开源代码(<https://github.com/tsinghua-fib-lab/ResInf>) and Zenodo), 但部分代码存在问题无法直接成功运行, 为此进行少量修改。除此之外, 基于该文章将模型应用于实际的目的, 我向所示数据集中注入了不同程度的噪声数据从而模拟现实数据中存在的噪声并观察模型的抗噪声能力从而改进模型的鲁棒性。

4.2 实验设置

在复现过程中, 实验主要针对于 micro(现实微生物系统)、mutualistic(生成生物互惠系统)、gene regulatory(生成基因调控系统)和 neuronal(生成人类神经元系统)四个数据集进行。数据集的信息如表 1 所示。

5 实验结果分析

首先研究了该文章提出的 ResInf 模型在上述各个数据集上的准确率表现, 所采用的主要评估指标是 F1 分数。

$$F1 = 2 \frac{precision \cdot recall}{precision + recall} = 2 \frac{TruePositive}{2 \cdot TruePositive + FalsePositive + FalseNegative}, \quad (4)$$

表 2. 在不同数据集进行实验的F1分数结果

训练数据集	测试数据集	F1均值	F1方差
Micro	Micro	0.8506	0.0419
人工合成数据	Micro	0.7897	0.0225
Mutualistic	Mutualistic	0.9783	0.0165
人工合成数据	Mutualistic	0.8415	0.0929
Gene Regulatory	Gene Regulatory	0.9495	0.0416
人工合成数据	Gene Regulatory	0.8445	0.0853
Neuronal	Neuronal	0.9349	0.0492
人工合成数据	Neuronal	0.9053	0.1059

$$TruePositive = \sum_{i=1}^P 1(y_i = 1 \wedge \bar{y}_i = 1), \quad (5)$$

$$FalsePositive = \sum_{i=1}^P 1(y_i = 0 \wedge \bar{y}_i = 1), \quad (6)$$

$$TrueNegative = \sum_{i=1}^P 1(y_i = 0 \wedge \bar{y}_i = 0), \quad (7)$$

$$FalseNegative = \sum_{i=1}^P 1(y_i = 1 \wedge \bar{y}_i = 0), \quad (8)$$

$$precision = \frac{TruePositive}{TruePositive + FalsePositive}, \quad (9)$$

$$recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (10)$$

其中 P 为所有样本的数量, y_i 代表第 i 个系统的弹性标签, \bar{y}_i 代表第 i 个系统的弹性标签,

实验所得结果如表2所示,ResInf模型在四个数据集上都取得了接近0.8及以上的F1分数。ResInf 可以有效地将弹性推断扩展到现实世界的微生物系统, 而无需先验知识和假设, 即使节点活动动态复杂且困难难以用公式明确表述, 也可以在各种网络系统中进行弹性推断。同时, 在部分现实场景中, 由于复杂系统尚未崩溃, 所获得的训练数据有限, 因此, 实验中选择利用模拟数据作为替代训练ResInf模型, 再在真实数据上测试, 也取得了较高的F1分数, 它强调了 ResInf 在现实场景中的实际适用性, 消除了从实验室和真实环境中收集大量数据的困难。在互惠、基因调控和神经元动力学控制的网络系统中, ResInf 在 F1 分数方面同样优秀, 表明它可以有效地从观察到的数据中捕获潜在的弹性损失模式以及拓扑和节点活动动态之间的复杂相互作用, 强调基于学习的方法在理解复杂网络系统方面的潜力。

之后, 为了检测ResInf模型在面对现实数据中普遍存在的噪声的表现水平, 我分别向原始微生物网络数据中注入了不同程度的高斯噪声模拟现实噪声, 结果如图4所示。当分别向

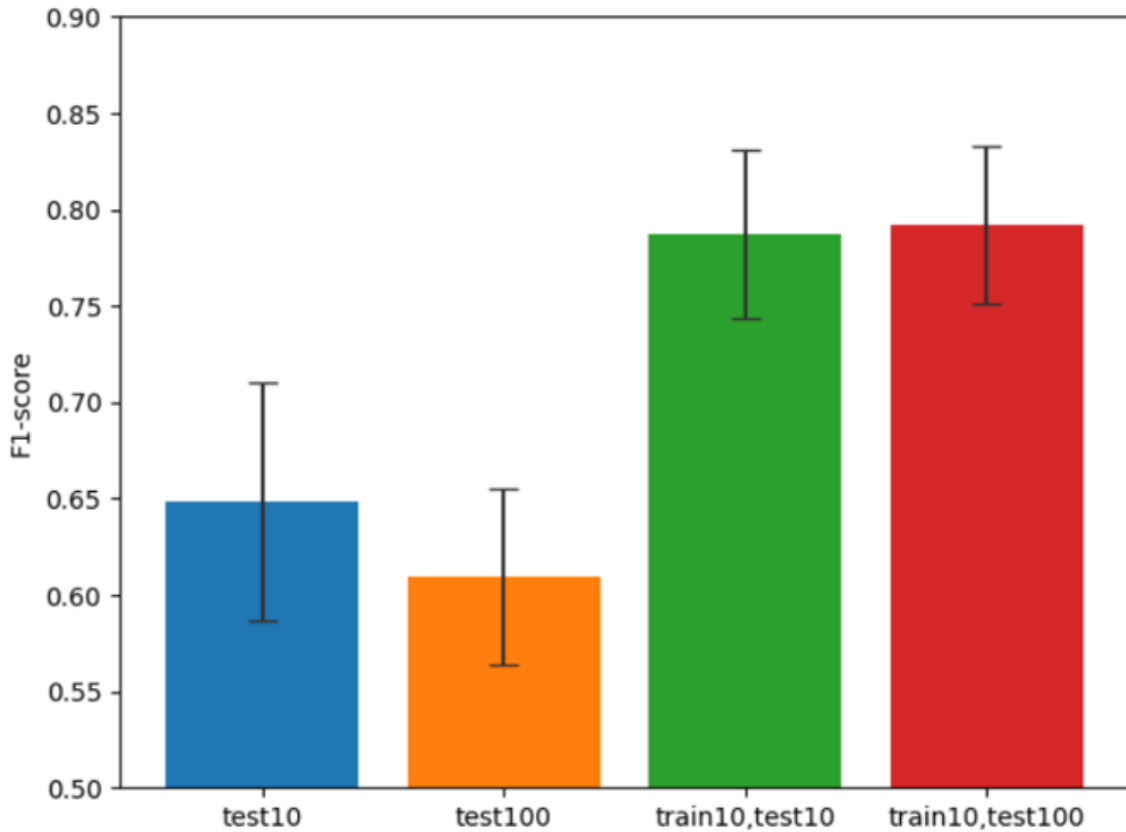


图 4. ResInf模型面对不同程度噪声的表现情况

测试集数据注入1%和10%程度的高斯噪声时，模型结果出现了较大幅度的下降，而只在训练集中加入较低程度的噪声之后，就可以提高模型面对含不同程度噪声数据的鲁棒性，该方法有效的提高了ResInf模型的抗噪声能力。

6 总结与展望

本人对文章提出的ResInf模型进行了复现实验，并且针对原模型抗噪声能力一般的不足进行了解决，实验证明了ResInf优越的复杂网络弹性预测能力和泛化能力，以及解决方法提高模型鲁棒性的有效性。未来可以继续分析网络的关键节点、边，了解模型判断依据，增强其可解释性，便于实际应用，还可以利用ResInf模型中间输出包含的系统弹性相关的拓扑和时序信息，尝试用于预测弹性丧失的节点以做预防。

参考文献

- [1] Baruch Barzel and Ofer Biham. Quantifying the connectivity of a network: The network correlation function method. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(4):046104, 2009.
- [2] Payel Das, Tom Sercu, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero Dos Santos, Pin-Yu Chen, et al. Accelerated

- antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering*, 5(6):613–623, 2021.
- [3] Jianxi Gao, Baruch Barzel, and Albert-László Barabási. Universal resilience patterns in complex networks. *Nature*, 530(7590):307–312, 2016.
 - [4] Jorge García-Girón, Alfio Alessandro Chiarenza, Janne Alahuhta, David G DeMar Jr, Jani Heino, Philip D Mannion, Thomas E Williamson, Gregory P Wilson Mantilla, and Stephen L Brusatte. Shifts in food webs and niche stability shaped survivorship and extinction at the end-cretaceous. *Science advances*, 8(49):eadd5040, 2022.
 - [5] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2. Minneapolis, Minnesota, 2019.
 - [6] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
 - [7] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.
 - [8] Rounak Meyur, Anil Vullikanti, Samarth Swarup, Henning S Mortveit, Virgilio Centeno, Arun Phadke, H Vincent Poor, and Madhav V Marathe. Ensembles of realistic power distribution networks. *Proceedings of the National Academy of Sciences*, 119(42):e2205772119, 2022.
 - [9] Flaviano Morone, Gino Del Ferraro, and Hernán A Makse. The k-core as a predictor of structural collapse in mutualistic ecosystems. *Nature physics*, 15(1):95–102, 2019.
 - [10] Charles Murphy, Edward Laurence, and Antoine Allard. Deep learning of contagion dynamics on complex networks. *Nature Communications*, 12(1):4720, 2021.
 - [11] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.
 - [12] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.
 - [13] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
 - [14] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

- [15] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.
- [16] Lu Zhong, Dimitri Lopez, Sen Pei, and Jianxi Gao. Healthcare system resilience and adaptability to pandemic disruptions in the united states. *Nature Medicine*, 30(8):2311–2319, 2024.
- [17] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.
- [18] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR, 2022.