TSA-TICER: A Two-Stage TICER Acceleration Framework for Model Order Reduction

1st Pengju Chen
School of Automation
Southeast University
Nanjing, China
pjchen@seu.edu.cn

2nd Dan Niu
School of Automation
Southeast University
Nanjing, China
danniu1@163.com

3rd Zhou Jin
School of Information Science and Engineering
China University of Petroleum-Beijing
Beijing, China
jinzhou@cup.edu.cn

4th Changyin Sun
School of Artificial Intelligence
Anhui University
Hefei, China
504513464@qq.com

5th Qi Li
School of Automation
Southeast University
Nanjing, China
liqikj@seu.edu.cn

6th Hao Yan
School of Integrated Circuits
Southeast University
Nanjing, China
yanhao@seu.edu.cn

Abstract— To enhance the post-simulation efficiency of large-scale integrated circuits, various model order reduction (MOR) methods have been proposed. Among these, TICER (Time-Constant Equilibration Reduction) is a widely-used resistor-capacitor (RC) network reduction algorithm. However, the time constant computation for eliminated-node classification in TICER is quite time-consuming. In this work, a two-stage TICER acceleration framework (TSA-TICER) is proposed. First, an improved graph attention network (named BCTu-GAT) equipped with betweenness centrality metric (BCM) based sample selection strategy and bi-level aggregation-based topology updating scheme (BiTu) is proposed to quickly and accurately determine all the eliminated nodes one time in the TICER. Second, an adaptive merging strategy for the new fill-in capacitors are designed to further accelerate the insertion stage. The proposed TSA-TICER is tested on RC networks with the size from 2k to 2 million nodes. Experimental results show that the proposed TSA-TICER achieves up to 796.21X order reduction speedup and 10.46X fill-in speedup compared to the TICER with 0.574% maximum relative error.

***Index Terms*—Model Order Reduction, TICER, Graph Neural Network, Merging Strategy**

I. Introduction

With the dramatic increase in the complexity and scale of modern VLSI circuits, post-layout simulation for large-scale integrated circuits directly can be very computationally challenging. In order to enhance the computation efficiency and reduce the memory footprint, model order reduction (MOR) is widely used to first reduce the scale of parasitic RC networks, which are generated by parasitic extraction tools in post-simulation[1]. After reduction, the equivalent RC(L) networks must be annotated back to the logic or analog circuit networks for further verification simulations [2, 3].

Recently, various MOR methods have been proposed and they can be divided into three categories: Krylov-subspace based MOR via moment matching [4], multigrid-like or node aggregation-based methods [5] and node elimination-based methods such as famous TICER [6, 7]. Among them, the Krylov-subspace based methods such as famous PRIMA [4] are not realizable for RC circuits and become inefficient for circuits with large number of I/O ports [2], since Krylov subspace dimension and the density of the projection matrices would grow quickly. Node aggregation reduction methods are based on the observation that those adjacent nodes with almost the same voltages can be aggregated together. They can generate realizable and sparse reduced models, but lack effective control of error [8]. Compared with them, the TICER (Time-Constant Equilibration Reduction) proposed by Sheehan [7] is a topological RC reduction method, which directly works on the network topology (so called realizable reduction methods) and is more suitable for multiple-port passive networks [2]. Amin et al. [9] extends the TICER to parasitic circuits including inductances. Moreover, only one internal node is eliminated in each step of TICER's recursive reduction process, many node elimination steps must be executed for the large-scale RC networks [3]. Thus some matrix partitioning based multi-dimensional reduction methods are attempted [10]. In [2], the HD-TICER (High-Dimensional TICER) algorithm is proposed to extend the TICER to its high-dimensional version, in which each step eliminates a whole subcircuit not just one circuit node and then extra elements are connected to the neighboring nodes to obtain an approximately equivalent circuit. However, the calculation of time constant τ to determine the eliminated nodes in TICER are quite time-consuming. It is also required to be executed repeatedly due to that the connection topology and RC information of the adjacent nodes will vary after one node is eliminated [3]. Therefore, whether and how the time-consuming eliminated-

This work was supported by the National Natural Science Foundation of China (Grant No. 62374031), National Key RD Program of China (Grant No. 2022YFB4400400), National Natural Science Foundation of China (Grant No. 62204265, 62234010). (Corresponding author: Dan Niu and Zhou Jin)

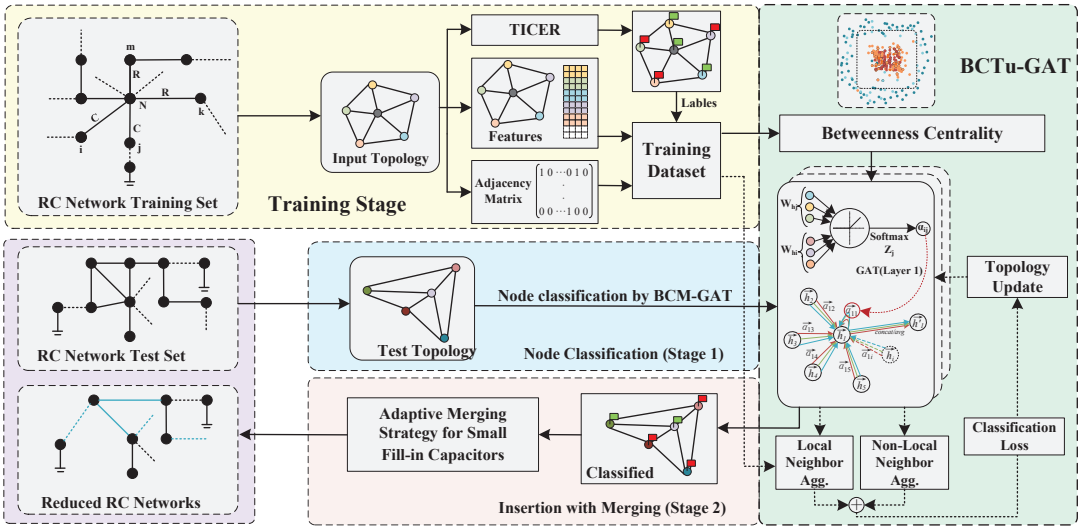

Fig. 1: The two-stage acceleration framework TSA-TICER for order reduction of parasitic RC networks.

node determination process can be improved or even removed is critical to enhance the reduction efficiency of large-scale RC networks using TICER.

It is known that the natural representation of circuits, netlists and layouts are graphs [11, 12]. GNN (Graph Neural Network) can directly operate on graphs and deal with graph-structured data [13]. Some models like GCN (Graph Convolutional Networks) have been widely used as feature extractors in EDA [11] and achieves observation point candidates prediction, component labeling in flattened netlists and so on [13]. In this work, an improved graph attention network based eliminated-node classification method working with adaptive merging strategy for small fill-in capacitors are proposed to achieve two-stage order reduction acceleration, which can significantly improve the reduction efficiency. The followings are the main contributions of this paper:

1) A two-stage acceleration framework for TICER (TSA-TICER) is proposed, which achieves quick eliminated-node determination and decreases the amount of fill-in capacitors. It can significantly enhance the order reduction efficiency of the TICER method for parasitic RC networks.
2) In eliminated-nodes classification stage, an improved graph attention network (named BCTu-GAT) equipped with betweenness centrality metric (BC)-based sample selection strategy and bi-level aggregation scheme based topology updating (BiTu) method is proposed to determine all eliminated nodes one time for TICER accurately, which completely removes the time-consuming time constant calculations and largely speeds up the node classification.
3) In the insertion stage, an adaptive merging strategy (AMS) is designed to deal with the new small capacitors by adaptive threshold to decrease the amount of fill-ins, which achieves remarkable insertion speedup but with just slight accuracy loss.

II. Preliminary

A. TICER Algorithm

The TICER is a topological RC network reduction method. It achieves the reduction of RC networks by selectively preserving nodes within a specific frequency range while eliminating the nodes with few neighbors and small nodal time constants [7]. Consider an N-terminal star network. The center of the star is node N, and the N terminals are labled 0 to N-1 (0 being ground). Then the nodal time constants τ can be calculate as follows [2]:

$$\tau = \frac{\sum_{k=0}^{N-1} c_{kN}}{\sum_{k=0}^{N-1} g_{kN}} \tag{1}$$

where the quantity g_{kN} is the conductance, and c_{kN} is the capacitance joining N to k. In practice, many of these values will be zero.

For a large-scale parasitic RC network, usually the computation of τ (division operation of floating-point number) for all nodes is quite time-consuming. Besides, TICER is one-dimensional in that only one internal node is eliminated in each step of its recursive reduction process, then the τ needs to be calculated repeatedly, since the connection and RC information of the adjacent nodes will vary in each elimination process [2]. Therefore, if the τ-based eliminated-node classification can be accelerated or even directly removed, the computation efficiency of order reduction can be enhanced largely.

In addition, in order to maintain the simulation accuracy of the RC networks after the node reduction, new resistors and capacitors will be inserted between the former neighbors of node N in TICER [7]. The insertion process also takes much time especially when a large number of new capacitors/resisters are generated and added.

B. Graph Attention Network (GAT)

Graph Neural Network (GNN) is a subfield of deep learning that focuses on analyzing and modeling graph-structured data

[11]. Graph Attention Network (GAT) is a reprensentive class of GNNs that introduce attention mechanism in feature aggregation. Recently, GAT has been effectively used as feature extractor in EDA since it can capture the structural dependencies of circuit topology [11]. Unlike other GNNs, GAT distinguishes the neighbors of each node with dynamic attention coefficients and adopts a weighted sum for feature aggregation [13], which has better representation ability.

In GAT, each node embedding is computed with all the neighbors by the self-attention mechanism. The node embedding process can be mathematically represented by the following equation:

$$\mathbf{h}_m^{(l+1)} = \phi\left(\sum_{n\in N(m)} \alpha_{(m,n)}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_n^l\right) \tag{2}$$

where $\mathbf{h}_m^{(l+1)}$ represents $(l+1)$-th layer embedding of node m, ϕ is the non-linear activation function, $\alpha_{(m,n)}^{(l)}$ denotes the normalized attention score of the node n to m that is obtained using the l-th attention mechanism, and $\mathbf{W}^{(l)}$ represents the trainable matrice that learns features of the neighbors [13].

III. Proposed two-stage TICER acceleration framework TSA-TICER

In this section, a two-stage acceleration framework (TSA-TICER) is proposed to enhance the order reduction efficiency of the TICER for parasitic RC networks, which is shown in Fig. 1.

First, in node classification stage, an improved graph attention network named BCTu-GAT is proposed to extract the node embedding features, which include node topology, degree, surrounding capacitances and conductances, to quickly classify all the eliminated nodes one time. It removes the time-consuming calculation for time constants and largely enhances the node classification efficiency. Second, for the insertion (fill-in) stage, an adaptive merging strategy is introduced to deal with the new small capacitors to decrease the amount of fill-in and further improve the insertion efficiency of TICER.

A. BCTu-GAT based Node Classification

In this part, an improved graph attention network (named BCTu-GAT) equipped with betweenness centrality metric (BCM) based sample selection strategy and bi-level aggregation based topology updating scheme (BiTu) is proposed to quickly and accurately determine all the eliminated nodes one time in TICER. It completely removes the time-consuming computation process for nodal time constants τ and significantly enhances the node classification efficiency for TICER. The BCM-based sample selection strategy and bi-level aggregation based topology updating scheme are designed to further enhance the node classification accuracy.

- BCM-based Training Sample Selection

In this eliminated-nodes classification task for model order reduction of parasitic RC networks, the different number of eliminated-labeled nodes and uneliminated-labeled nodes will lead to unbalanced training samples, which usually influences the training speed and convergence of graph neural network.

Betweenness Centrality Metric (BCM) is widely used to identify the important nodes of the network based on their structural positions [14]. In this work, BC metric instead of the random sampling selection is designed to select important training nodes for GNN, which can achieve high-efficiency information propagation and property transfer from node to node.

The betweenness centrality metric $BCM(x)$ of a node x is defined as

$$BCM(x) = \sum_{p,q\neq x,\sigma_{pq}\neq 0} \frac{\sigma_{pq}(x)}{\sigma_{pq}} \tag{3}$$

where σ_{pq} is the number of shortest paths from p to q while $\sigma_{pq}(x)$ stands for the number of these paths passing through x [14]. Because the BCM only executes only once in pre-training stage, it isn't expensive to compute in the whole progess.

The buliding process of the proposed BCM is also shown in Fig. 1. First, the nodes and the feature matrix in the training parasitic RC networks are input. Then the nodes with high BCM value are selected as the training dataset. Next, the selected training nodes along with the feature and adjacency matrices are supplied to GAT to train. Finally, the node classification results (eliminated nodes or uneliminated nodes) can be obtained as the output. It can be verified that the training efficiency and classification accuracy of GAT model can be further improved when nodes with high BCM are selected as training nodes for the model.

- Bi-level Aggregation-based Topology Updating

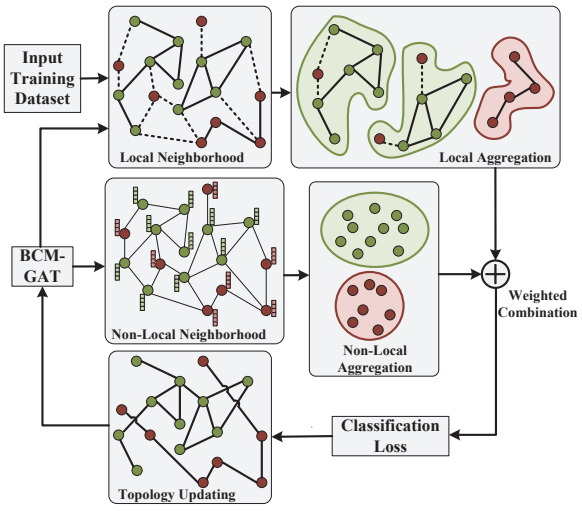

Fig. 2: The framework of bi-level aggregation-based topology updating.

In the eliminated-node and uneliminated-node classification of parasitic RC networks, the different types of nodes are usually mixed up. We define the local neighborhood as an area composed of nodes that are closely connected to the target node and are related to each other in the feature space. In this area, there are always a small number of nodes of another type. For GAT, it is difficult to distinguish between eliminated-nodes

and uneliminated-nodes in their local neighborhood, especially when the number of nodes of one type is small. It will lead worse classification accuracy. Moreover, another problem in the node classification task is that the same type nodes might widely distribute in the topology of RC parasitic networks. In this case, the GAT fails to incorporate information from distant but same type nodes. Apart from the local neighborhood, we also define the non-local neighborhood as the area containing nodes of the same type, and these nodes are not confined to local areas.

In order to improve the connectivity of nodes of the same type, we propose the bi-level aggregation-based topology updating scheme (BiTu), as shown in Fig. 2. The proposed BiTu combines the local aggregation and non-local aggregation, and considers the graph topology and feature-based correlations between nodes to update the RC network topology.

The local neighborhood in the topology of RC parasitic network is determined as follows:

1) The weights $W_{u,v}$ on edges are defined as:

$$W_{u,v} = \frac{max(\|\tau_i\|_2) - \|\tau_u - \tau_v\|_2}{max(\|\tau_i\|_2)} (i = 1, 2, ..., n) \tag{4}$$

where τ_i is the time constant of node i.

In the local neighborhood, the connection of nodes should also be considered. Thus, the connecting parameter $\delta_{u,v}$ is defined. The $\delta_{u,v} = 1$ when node u and node v are connected, otherwise 0. Then, the weighted score of a local neighborhood can be formulated as:

$$\mathcal{Q}(P_j) = \frac{1}{2|W|} \sum_{u,v} \left[W_{uv} - \frac{\tau(u)\tau(v)}{2|W|} \right] \delta_{uv} \tag{5}$$

where $|W|$ is the sum of weights of all edges. The value of $\mathcal{Q}$ indicates the better partitioning sub-topology P_j of the whole topology into local neighborhoods.

2) We use the Louvain algorithm [15] to search the set of local neighborhoods in the topology with starting each nodes as a single neighborhood. Then, we merge the different neighborhoods when weighted score is maximized. After the optimal partitioning $P^* = argmax_{P_j \in P} Q(P_j)$ is obtained, the local neighborhood $N_{\mathcal{L}}$ can be defined as $N_{\mathcal{L}}(u) = \{\forall v \in V : \gamma_v == \gamma_u\}$, and the γ_u denotes the local neighbors of node u.

Moreover, the preliminary results of classification by BCM-GAT are used to determine non-local neighborhood as a group of nodes that are strongly correlated in their latent space but not local according to graph topology. It is defined as $N_{\widetilde{\mathcal{L}}}(u) = \{\forall v \in V : t_v == t_u\}$, and the t_u denotes the type of node u .

After obtaining the local neighborhoods and non-local neighborhoods, the attentive aggregation (AA) [16] can be utilized to aggregate features from relevant-densely connected nodes in $N_{\mathcal{L}}(u)$ and aggregate features from distant but informative nodes in $N_{\widetilde{\mathcal{L}}}(u)$:

$$\mathcal{H}_{\mathcal{L}}(u) = \mathrm{Re}\, LU\left(W_{\mathcal{L}} \cdot AA\left(\{X_v, \forall v \in \mathcal{N}_{\mathcal{L}}(u)\}\right)\right) \tag{6}$$

$$\mathcal{H}_{\widetilde{\mathcal{L}}}(u) = \mathrm{Re}\, LU\left(W_{\widetilde{\mathcal{L}}} \cdot AA\left(\{X_v, \forall v \in \mathcal{N}_{\widetilde{\mathcal{L}}}(u)\}\right)\right) \tag{7}$$

Next, we combine the local and non-local aggregation, and multiply with weights to compute the final classification loss:

$$\mathcal{H}_{\mathcal{F}}(u) = \mathrm{ReLU}\left(W_{\mathcal{F}} \cdot \left(\mathcal{H}_{\mathcal{L}}(u) \oplus \mathcal{H}_{\widetilde{\mathcal{L}}}(u)\right)\right) \tag{8}$$

where the $W_{\mathcal{L}}$, $W_{\widetilde{\mathcal{L}}}$ and $W_{\mathcal{F}}$ are the learnable weights for corresponding equations, $\oplus$ means concatenation.

At last, the input topology is updated based on the combined results of the bi-level aggregation, where the inter-class edges between different types of nodes are removed and the intra-class edges between same type of nodes are added. It can efficiently improve the accuracy of node classification.

B. Adaptive Merging Strategy for Fill-in Capacitors

After the eliminated nodes are determined by the above-mentioned BCTu-GAT based node classification algorithm, node elimination and insertion of new resistors and capacitors between former neighbors of N will be conducted according to the TICER rules to maintain an approximate circuit-level equivalence. The value of new inserted conductance g_{ij} and capacitor c_{ij} can be calculated as [11]:

$$g_{ij} = \frac{g_{iN} g_{jN}}{\sum_{k=0}^{N-1} g_{kN}} \tag{9}$$

$$c_{ij} = \frac{c_{iN} g_{jN} + g_{iN} c_{jN}}{\sum_{k=0}^{N-1} g_{kN}} \tag{10}$$

where g_{iN} and g_{jN} represent the conductance of node i and node j connected to the node N, respectively. The c_{iN} and c_{jN} represent the capacitor of node i and node j connected to the node N, respectively.

From Eq. (10), when the number of neighboring nodes connected to the eliminated node N is large or the values of conductance g_{iN} are relatively decentralized, large number of fill-in small capacitors will be generated and inserted to the surrounding nodes during the insertion stage of TICER. They usually have a relatively slight impact on the simulation accuracy, but have a serious impact on the amount of successive fill-ins (the creation of new resistors and capacitors when a node is eliminated) and further influence the reduction efficiency.

In this case, merging some new small capacitors will effectively reduce the fill-ins and enhance the insertion efficiency. The minimum threshold C_{min} for retaining new fill-in capacitor is critical, which is required to balance the fill-in efficiency and reduction accuracy. In this work, the capacitor threshold C_{min} is adaptively determined as follows. We compute the new fill-in capacitors for all eliminated nodes obtained from the above BCTu-GAT classification method and randomly select the k new capacitors in each remaining Net, and then arrange the $k*m$ new capacitors in descending order. The m is the number of remaining Net. The threshold C_{min} can be adaptively determined by selecting at a set position of this descending order. Next, the new fill-in capacitors which

are smaller than the threshold C_{min} will be merged. The detailed merging strategy is shown in Algorithm 1. Note that, the threshold C_{min} can not be adaptively obtained in the TICER, since the eliminated nodes in the TICER can not be determined one time, but obtained one by one. Therefore, it is impossible to compute all the new fill-in capacitors in advance and establish an order.

Algorithm 1 Merging of small new capacitors

Require:
The RC network T_{RC};
The adaptive threshold C_{min};
Ensure:
Set the new fill-in capacitor set F_N of Node N;
Set the adjacency capacitor set A_j of Node j;
Name the capacitor between Node i and Node j as C_{ij}.
for Node N in the T_{RC} **do**
  Put the new fill-in capacitors of Node N into F_N;
  for $C_{ij} \in F_N$ & $C_{ij} < C_{min}$ **do**
    Name the node i's Net as Net_i;
    for $C_{kj} \in A_j$ **do**
      if Node $k \in Net_i$ **then**
        C_{kj} + = C_{ij};
        Delete C_{ij};
      end if
    end for
    Update the T_{RC};
  end for
  The compensation capacitors of node $i \leftarrow C_{kj}$;
end for

In all, by employing the proposed BCTu-GAT to quickly determine all the eliminated nodes one time in the node classification stage and the adaptive merging strategy (AMS) for new small capacitors in the insertion (fill-in) stage, the proposed two-stage acceleration framework (TSA-TICER) will significantly improve the reduction efficiency of the TICER for parasitic RC networks while keeping high reduction accuracy.

IV. Experiments and Results

A. Experimental Environment

In this work, to fully evaluate the computation efficiency and accuracy of MOR, the proposed TSA-TICER and the TICER method are tested on the parasitic RC networks with different orders of magnitude. The Ngspice is used to conduct the transient analysis for original RC networks and the reduced RC networks by MOR to obtain maximum solution errors and gives the visible solution curves. The BCTu-GAT based node classification network is trained on the AMAX X12SPA-TF server with NVIDIA RTX 3090 24GB GPU.

B. Order Reduction Efficiency

First, the running time for MOR is computed to verify the reduction efficiency. Moreover, the TICER with GCN-based classification and adaptive merging strategy (named GCN-based TICER) is also achieved. For comparison, the TICER, GCN-based TICER as well as the proposed TSA-TICER (proposed BCTu-GAT + adaptive merging strategy (AMS)) are tested on the different sizes of parasitic RC networks from 2k to 2 million nodes. In addition, multiple ports are set up in each benchmark to verify the output signal. In order to ensure the fairness, we use the priority queue implementation to update the τ in the TICER. The test results are shown in Table I.

From this table, it is clear that both the GCN-based TICER and proposed TSA-TICER can significantly decrease the run time for MOR compared with the TICER, due to the removal of the time-consuming calculation for time constant τ and adaptive merging strategy for small fill-in capacitors. Note that, the speed-up ratio will sharply increase as the scale of RC networks increases and can reach nearly 800X in maximum.

In addition, ablation experiment is conducted to verify the fill-in acceleration performance of the adaptive merging strategy (AMS), which is shown in Table II. By comparing the insertion time of TICER and TICER with AMS, it is obvious that the proposed AMS can effectively reduce the number of fill-in capacitors and shorten the insertion time, and as the scale of RC network increases, the proposed AMS can achieve higher insertion acceleration.

TABLE II: Insertion efficiency comparisons with different methods

Circuits	Number of nodes	Insertion time (s)		speed-up
		TICER	TICER+AMS	vs TICER
circuit1	2094	0.0001	0.0001	1.00X
circuit2	77214	0.1207	0.0959	1.26X
circuit3	152643	0.3580	0.1681	2.13X
circuit4	335698	0.9871	0.2812	3.51X
circuit5	2574826	48.6445	4.6503	10.46X
Average	-	-	-	3.672X

C. Order Reduction Accuracy

Next, to verify the reduction accuracy, we conduct the transient analysis (TA) using the Ngspice for the reduced RC networks, which are obtained by the TICER, GCN-based TICER and the proposed TSA-TICER. Moreover, in order to test the effectiveness of the GAT, the betweenness centrality

TABLE I: Order reduction efficiency comparisons with different MOR methods for RC networks

Circuits	Number of original nodes	Number of eliminated nodes	Running time of MOR (s)			Speedups	
			TICER	GCN-based TICER	TSA-TICER (Ours)	vs TICER	vs GCN-based TICER
circuit1	2094	1569	0.035	0.0041	0.0040	8.75X	1.03X
circuit2	77214	56910	7.585	0.2799	0.2739	27.69X	1.02X
circuit3	152643	110849	20.022	0.5012	0.4798	41.72X	1.04X
circuit4	335698	247778	45.360	0.7786	0.7848	57.80X	0.99X
circuit5	2574826	1919275	8034.697	10.5660	10.0912	796.21X	1.05X
Average	-	-	-	-	-	186.434X	1.03X

metric (BCM), bi-level aggregation based topology updating (BiTu) and adaptive merging strategy (AMS) for the MOR accuracy, the ablation experiments are added. The proposed TSA-TICER only with GAT (ours without BCTu), TSA-TICER without BCM (ours without BCM), without BiTu (ours without BiTu) and without adaptive merging strategy (ours without AMS) are tested. The maximum relative errors in transient analysis using above-mentioned MOR methods are presented in Table III.

TABLE III: Maximum relative error of TA with different MOR methods

Number of nodes	TICER	GCN-based TICER	Ours	Ours w/o BCTu	Ours w/o BCM	Ours w/o BiTu	Ours w/o AMS
2094	0.065%	0.584%	**0.186**%	0.391%	0.374%	0.358%	0.184%
77214	0.087%	0.613%	**0.206**%	0.398%	0.390%	0.378%	0.201%
152643	0.102%	0.718%	**0.242**%	0.461%	0.455%	0.431%	0.230%
335698	0.117%	1.084%	**0.499**%	0.862%	0.711%	0.660%	0.481%
2574826	0.192%	1.475%	**0.574**%	1.020%	0.954%	0.921%	0.543%

From this table, the maximum relative errors of those GNN based TICER are higher than that of the TICER, but our TSA-TICER transient analysis error are much lower than the GCN-baded TICER. Moreover, by comparing the results among the proposed method (Ours), Ours without BCTu, Ours without BCM, Ours without BiTu and GCN-based TICER, it is clear that both the BCM-based sample selection strategy, attention mechanism and the bi-level aggregation based topology updating scheme can enhance the node classification accuracy and further the order reduction accuracy. In addition, by comparing Ours with Ours without AMS, it can be obtained that the proposed adaptive merging strategy (AMS) can largely increase the fill-in efficiency, but with just slight accuracy loss.

Finally, a transient solution curve of example node in circuit5 is shown in Fig. 3. It can be seen that all the MOR methods can achieve high transient solution accuracy and the proposed TSA-TICER can achieve higher solution accuracy than GCN-based TICER.

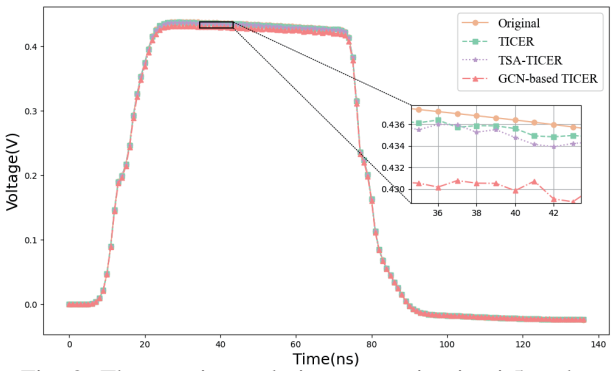

Fig. 3: The transient solution curves in circuit5 under different MOR methods

From the experimental results, it is demonstrated that the proposed two-stage order reduction acceleration framework (TSA-TICER) using BCTu-GAT based node classification and adaptive merging strategy for fill-in capacitors can significantly enhance the reduction efficiency than the TICER while ensuring high reduction accuracy.

V. Conclusion

In this paper, we propose a two-stage TICER acceleration framework (TSA-TICER). First, an improved graph attention network named BCTu-GAT equipped with betweenness centrality metric (BC)-based sample selection strategy and bi-level aggregation-based topology updating (BiTu) scheme is proposed to speed up the eliminated-node classification stage in TICER. Moreover, an adaptive merging strategy for small new capacitor is designed to decrease the fill-in amount and further improve the insertion efficiency. Test results demonstrate that the proposed TSA-TICER can significantly enhance the order reduction efficiency (up to 796.21X) while keeping high simulation accuracy.

References

[1] S. Chellappa, L. Feng, V. d. l. Rubia and P. Benner, "Inf-Sup-Constant-Free state error estimator for model order reduction of parametric systems in electromagnetics," *IEEE Transactions on Microwave Theory and Techniques*, pp. 1–16, 2023.

[2] L. Hao and G. Shi, "High-Dimensional extension of the TICER algorithm," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 17, pp. 4722–4734, 2021.

[3] L. Hao and G. Shi, "Realizable reduction of multi-Port RCL networks by block elimination," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 17, pp. 399-412, 2023.

[4] A. Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, pp. 645-654, 1998.

[5] Q. Huang, X. Li, C. Fang, F. Yang, Y. Su, and X. Zeng, "An aggregating based model order reduction method for power grids," *Integration*, vol. 55, pp. 449–454, 2016.

[6] C. Antoniadis, N. Evmorfopoulos and G. Stamoulis, "A rigorous approach for the sparsification of dense matrices in model order reduction of RLC circuits," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pp. 1-6, 2019.

[7] B. N. Sheehan, "TICER: realizable reduction of extracted RC circuits," in *1999 IEEE/ACM International Conference on Computer-Aided Design(ICCAD)*, pp. 200-203, 1999.

[8] Y. Su, F. Yang and X. Zeng, "AMOR: An efficient aggregating based model order reduction method for many-terminal interconnect circuits," in *Design Automation Conference(DAC)*, pp. 295-300, 2012.

[9] C. S. Amin, M. H. Chowdhury and Y. I. Ismail, "Realizable reduction of interconnect circuits including self and mutual inductances," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, pp. 271-277, 2005.

[10] Z. Ye, D. Vasilyev, Z. Zhu, and J. R. Phillips, "Sparse Implicit Projection (SIP) for reduction of general many-terminal networks," in *2008 IEEE/ACM International Conference on Computer-Aided Design(ICCAD)*, pp. 736-743, 2008.

[11] D. Sánchez, L. Servadei, G. N. Kiprit, R. Wille, and W. Ecker, "A comprehensive survey on electronic design automation and graph neural networks: theory and applications," *ACM Transactions on Design Automation of Electronic Systems*, pp. 1–27, 2023.

[12] Z. Jin, H. Pei, Y. Dong, X. Jin, X. Wu, W. W. Xing, and D. Niu, "Accelerating nonlinear dc circuit simulation with reinforcement learning," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pp. 619–624, 2022.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR) 2018*, 2018.

[14] M. Barthelemy, "Betweenness centrality in large complex networks," *The European Physical Journal B - Condensed Matter*, vol. 38, pp. 163–168, 2004.

[15] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, p. P10008, 2008.

[16] S. Ji, S. Pan, G. Long, X. Li, J. Jiang and Z. Huang, "Learning Private Neural Language Modeling with Attentive Aggregation," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, 2019.