

399 **A Appendix**

400 **A.1 Message Passing in SyncTREE**

401 As the message passing process in Figure 8, information from leaves, sub-branches, and the whole
 402 global structure is first collected following the bottom-up propagation by GAT_{bu} . Then, the final node
 403 representations of GAT_{bu} are applied to each layer of GAT_{td} to jointly update the node attributes of
 404 the corresponding top-down tree with the node embeddings of its previous layer. As the example
 405 shown in Figure 8, by designing this two-pass message-passing mechanism, the node features will
 406 incorporate the information from different levels and become more expressive. Furthermore, in the
 407 top-down tree, the root node can only be updated with synchronized h_{bu}^L since it doesn't have any
 408 incoming connection, it ensures that information injection at the source of the top-down tree is fixed
 which can help to maintain differentiable feature embeddings without over-smoothing.

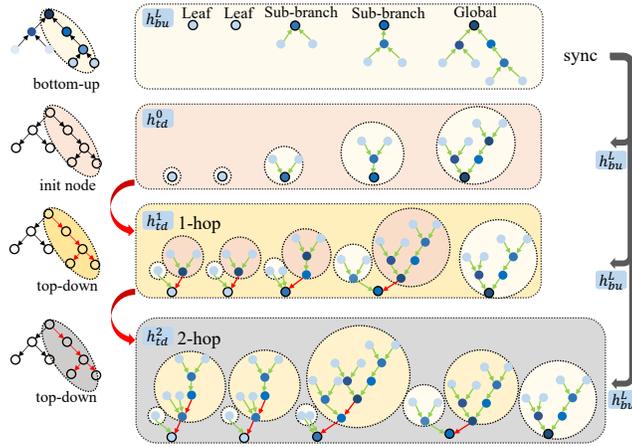


Figure 8: Illustration of our two-pass message-passing mechanism.

409

410 **A.2 Synthetic and RISC-V Dataset Preparation**

411 Our dataset is composed of artificially generated and practical RC trees and the golden timing results
 412 at sinks (leaf nodes of each RC tree) obtained by SPICE simulation. On the one hand, we follow the
 413 pipeline in Figure 9 to generate the synthetic dataset. To be specific, we first adopt Algorithm 1 to
 414 generate RC-trees with random typologies and then convert them to artificial IC interconnects for
 415 further SPICE timing measurement. On the other hand, we directly extract RC trees from practical
 416 RISC-V circuit designs to compose the RISC-V dataset. The statistics of both datasets are shown in
 417 Figure 10 and Figure 11.

Algorithm 1 Generate artificial RC-trees

Require: $v_d \in [v_{min}, v_{max}]$, $R \in [R_{min}, R_{max}]$, $C \in [C_{min}, C_{max}]$
 Initialize voltage v_d of driving cell, edge type (rising or falling), depth D of RC tree
 $parent\ set = list[drivingcell]$
 $parent \leftarrow$ randomly pick one element from $parent\ set$
while $depth \leq D$ **do**
 randomly choose R, C
 generate $child$, add $child$ into $parent\ set$
 the R_{child} of the edge from $child$ to the $parent \leftarrow R$
 the C_{child} of $child$ to the ground $\leftarrow C$
 $parent \leftarrow$ randomly pick one element from $parent\ set$
 $D = D + 1$
end while

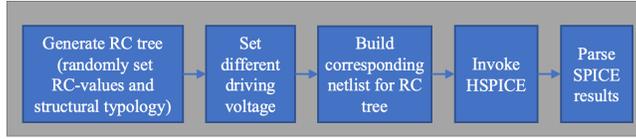


Figure 9: Pipeline for the synthetic dataset generation.

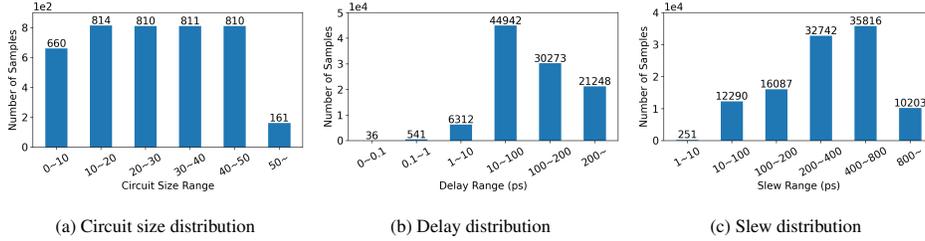


Figure 10: Synthetic dataset statistics.

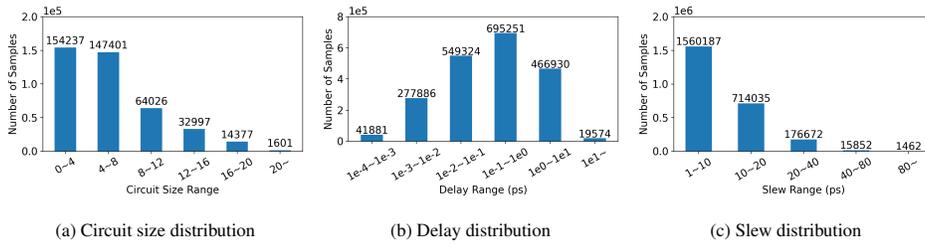


Figure 11: RISC-V dataset statistics.

418 A.3 Baselines's Implementation

419 The GNNs of all the baseline models are set with 32/128 hidden dimensions separately for the
 420 synthetic dataset/RISC-V dataset. For GraphTrans, the dimension of the feedforward full-connection
 421 layers in the Transformer of GraphTrans is set to 256 with 0.1 dropout probability between layers, the
 422 number of attention heads is set to 4, and the max input sequence length is set to the maximum circuit
 423 size. It should be noted that we only made a little modification to the GraphTrans model. GraphTrans
 424 is originally designed for node classification tasks, it takes CLS token from Transformer output as the
 425 representation of the whole graph and applies a linear module followed by softmax to implement
 426 prediction. In order to incorporate global information into node features, in our experiments, we
 427 concatenate the CLS token with node embeddings and then feed it into MLP to get the final output.
 428 For NTREE, we set GAT as its basic block with a 0.2 dropout probability between layers. We follow
 429 the original junction-tree-based algorithm in [10] to compose H-trees from our RC circuits with the
 430 same radius setting for extracting subgraphs in the paper.

431 A.4 Analysis of TContrast Loss

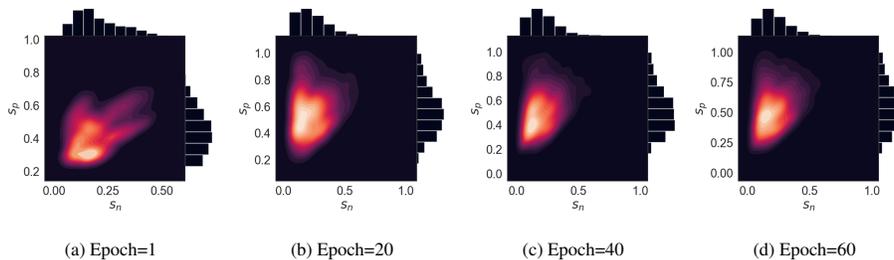


Figure 12: The distribution of similarity pairs with training epochs.

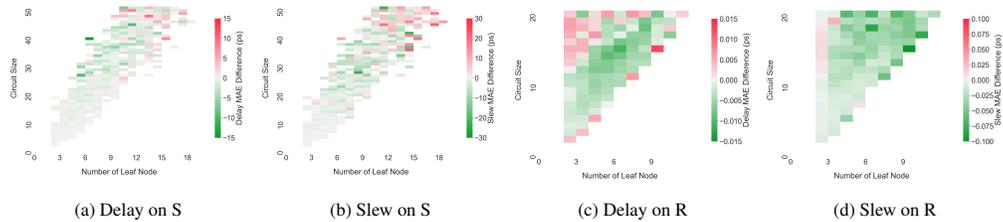


Figure 13: Mean Average Error difference after applying TContrast loss on the synthetic dataset (S) and RISC-V dataset (R). (Negative values indicate that TC-loss-guided SyncTREE has a lower MAE error than vanilla SyncTREE)

432 To visualize the converging process during training, we plot the distribution of similarity pairs in
 433 space at different epochs in Figure 12. It obviously shows that our model approaches the optimization
 434 goal with a more concentrated similarity distribution after enough training with the guidance of
 435 TContrast loss. In Figure 13, we show the MAE difference of timing results obtained by vanilla
 436 SyncTREE and TC-loss guided SyncTREE. As shown in the results, after being combined with TC
 437 loss, our SyncTREE model has smaller errors for most types of RC trees which can effectively prove
 438 the validity of TC loss.