

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 SI-GT: FAST INTERCONNECT SIGNAL INTEGRITY ANALYSIS FOR INTEGRATED CIRCUIT DESIGN VIA GRAPH TRANSFORMERS

006
007 **Anonymous authors**
008 Paper under double-blind review

011 ABSTRACT

013 Signal integrity issues present significant challenges in modern integrated circuit
014 (IC) design, as crosstalk-induced delay variation and transient glitches caused by
015 capacitive coupling among interconnects can severely impact IC functional cor-
016 rectness. Although circuit simulators like SPICE can deliver accurate signal in-
017 tegrity analysis, their computational cost becomes prohibitive for large-scale de-
018 signs. In this paper, we propose Si-GT, a novel transformer-based model for fast
019 and accurate signal integrity analysis in IC interconnects. Our model elaborates
020 three key designs: (1) virtual NET token to encode net-specific signal char-
021 acteristics and serve as net-wise representation, (2) mesh pattern encoding to embed
022 high-order mesh structures at each node while distinguishing uncoupled wire seg-
023 ments, and (3) intra-inter net (IIN) attention mechanism to capture structures of
024 signal propagation path and coupling connections. To support model training and
025 evaluation, we construct the first interconnect signal integrity dataset comprising
026 200k delay examples and 187k glitch examples using SPICE simulations as the
027 golden reference. Our experiments show that our Si-GT surpasses state-of-the-art
028 graph neural network and graph transformer baselines with substantially reduced
029 computation compared to SPICE, offering a scalable and effective solution for
030 interconnect signal integrity analysis in IC design verification.

031 1 INTRODUCTION

032
033 Signal integrity (SI) analysis is essential in integrated circuit (IC) design to ensure reliable signal
034 transmission and correct timing behavior (Caiguet et al., 2001). Among signal integrity problems,
035 crosstalk is the primary culprit. Dense interconnect layouts and high-speed signaling in modern
036 ICs exacerbate crosstalk-induced noise and delay variations, leading to potential functional errors,
037 performance degradation, and even chip failure Li et al. (2022); Song et al. (2015). Engineers have to
038 run SPICE simulations (Quarles et al., 1994) repeatedly throughout IC design flow to identify circuit
039 behavior and crosstalk-induced noise and delay violations, allowing careful crosstalk mitigations
040 (Vittal & Marek-Sadowska, 1997; Stöhr et al., 1998; Duan et al., 2010; Gao & Liu, 1996), which is
041 computationally prohibitive for very-large-scale integration (VLSI) (Achar & Nakhla, 2001).

042 Recently, machine learning (ML) has emerged as a computationally efficient surrogate for signal
043 integrity analysis in IC design (Kahng et al., 2015; Lu & Lim, 2022; Swaminathan et al., 2020;
044 Wang & Luo, 2019; Cheng et al., 2020; Liang et al., 2022; Liu et al., 2025). However, most prior
045 efforts primarily concentrate on timing prediction, aiming to “unravel the mystery” of black-box
046 timing estimation formulas in sign-off timers. These works generally do not model crosstalk effects
047 explicitly with aggressor–victim switching interactions and signal pattern-dependent analysis.

048 Advances in graph neural networks (GNNs) (Wu et al., 2020) and graph transformer (GT) (Dwivedi
049 & Bresson, 2020) have revolutionized machine learning capabilities for graph-structured data, en-
050 abling breakthrough applications in electronic design automation (EDA) from precise timing (Guo
051 et al., 2022; Hu et al., 2023; Lin et al., 2025; Zhong et al., 2024; Guo et al., 2025) and parasitics
052 prediction (Ren et al., 2020; Shahane et al., 2023; Yoon et al., 2025; Liu et al., 2023) to complex
053 optimization tasks like placement (Lu et al., 2020; Ding et al., 2024; Hou et al., 2025) and routing
(Cheng & Yan, 2021; Liao et al., 2020; Wang et al., 2024). However, developing a graph learn-

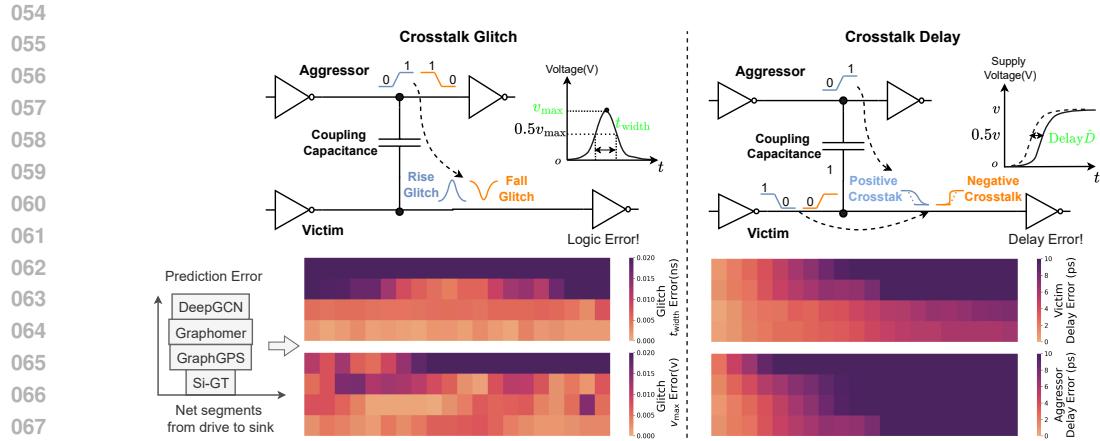


Figure 1: Prediction tasks illustration and Si-GT model performance.

ing model for signal integrity analysis is challenging due to the complex crosstalk effect (Aragones & Rubio, 2003). In IC interconnects, the crosstalk effect arises from electromagnetic interference between signals propagating on adjacent wires. On the one hand, the severity and nature of this interference depend on multiple factors, including switching directions, active/quiet net states, slew rate, coupling capacitance, and wire characteristics (Wong et al., 2000; You & Soma, 1990). On the other hand, the crosstalk effect exhibits both long-range dependencies (i.e., signal propagating from drive to distant load) and adjacent net-wise dependencies (i.e., energy transfer between coupled nets). The successful application of GNNs to EDA tasks relies on incorporating domain physics into the graph’s inductive bias (Haoxiang et al., 2022). To design an effective graph learning model for signal integrity analysis, it’s important to encode both signal switching patterns and structural features into the graph inductive bias while accounting for the unique circuit behaviors under crosstalk effect.

Graph transformers are excellent at capturing long-range dependencies through self-attention mechanisms. To this end, we propose Si-GT, a novel graph transformer model for IC interconnect signal integrity analysis. Si-GT incorporates three key designs: (1) Mesh pattern encoding, which embeds local mesh structures at each node to enrich node features and separate uncoupled nets; (2) Virtual `<NET>` tokens, which encode net-specific signal characteristics (e.g., switching direction and slew rate) and serve as net-level representations, with their receptive fields restricted to the corresponding nets via attention masks; (3) Intra–Inter Net (IIN) attention, which explicitly models both the spatial relationships among nodes within a net and the coupling effects from adjacent nets connected by coupling capacitors. Our contributions are summarized as follows:

- We propose Si-GT, a Transformer-based model for fast interconnect signal integrity analysis. To enhance graph inductive bias, Si-GT leverages virtual NET tokens for net-level signal encoding, mesh pattern encoding for local coupling structures, and intra-inter net attention to capture signal propagation and coupling effects.
- We construct a dataset for ML-based signal integrity analysis of IC circuits, comprising 200,200 crosstalk delay examples and 187,309 crosstalk glitch examples referring to golden SPICE simulations. To the best of our knowledge, this is the first large-scale dataset dedicated to IC interconnect signal integrity analysis.
- Experiments highlight the superior performance of Si-GT over advanced GNNs and graph transformers, as well as in computational efficiency compared to SPICE simulation. We validate the effectiveness of each design in Si-GT through ablation studies.

2 RELATED WORK

Crosstalk Effect. Crosstalk is a severe signal interference that degrades signal integrity in circuits (Hall & Heck, 2011). As illustrated in Figure 1, when a signal transitions on the interconnect (aggressor), it induces a voltage disturbance on the adjacent interconnect (victim). This interference can

manifest either as a crosstalk glitch on the victim, leading to a logic error, or as crosstalk-induced delays in signal propagation, causing timing failures (Vittal et al., 1999). Different switching patterns of aggressor and victim can create distinct delay scenarios. When aggressor and victim switch in the same direction, constructive interference occurs, accelerating the victim’s transition and potentially causing timing violations. When they switch in opposite directions, destructive interference occurs, slowing the victim’s transition and increasing delay (Wong et al., 2000; You & Soma, 1990).

ML for SI. ML has been applied to reduce the cost of SI analysis in circuit design cycles (Lu & Lim, 2022). Related studies mainly fall into three categories: (1) early-stage crosstalk mitigation at the global routing stage, including critical net classification (Liang et al., 2020; 2022), crosstalk-aware placement (Gao et al., 2022; Yu et al., 2025), gate sizing (Zhou et al., 2022; Lu et al., 2021), and buffer insertion (Ding et al., 2024); (2) pre-routing timing estimation (Jin et al., 2024); (3) post-routing timing estimation (Kahng et al., 2015; Cheng et al., 2020; Liu et al., 2025; Ye et al., 2023). These works for SI only serve for timing prediction and share a key limitation that none of them consider signal pattern variability in both their dataset and model design, which is central to accurate and practical signal integrity analysis.

Graph Transformer. Graph transformer (Kreuzer et al., 2021; Yuan et al., 2025; Ying et al., 2021) encodes structural information into the graph inductive bias and leverages the graph attention mechanism to capture the long-range dependencies, breaking the limitation of message-passing GNNs in capturing global context due to its inherent over-smoothing and over-squashing issues (Pei et al.). A graph transformer layer is composed of a self-attention module followed by a feed-forward neural network (FFN). Given a graph \mathcal{G} having n nodes with node feature matrix $X \in \mathbb{R}^{n \times d}$ where d is node feature dimension, self-attention module will project X into query, key, and value matrices: $Q = XW_Q$, $K = XW_K$, and $V = XW_V$ with three trainable weight matrices $W_Q, W_K \in \mathbb{R}^{d \times d_K}$, $W_V \in \mathbb{R}^{d \times d_V}$ respectively. Then global attention is calculated with self-attention module: $\text{Attn}(X) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$.

3 FORMULATION AND BACKGROUND

3.1 INTERCONNECT FEATURIZATION

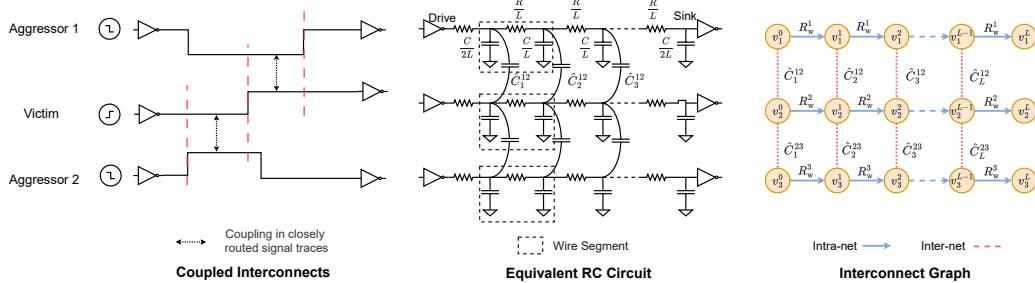


Figure 2: Graph representation for coupled IC interconnects.

IC interconnects are typically modeled as distributed RC circuits using wire-load models (Jin et al., 1999). As shown in Figure 2, for three coupled interconnects (two aggressors and one victim) driven and loaded by inverters, its equivalent RC circuit can be derived by breaking the wire into L equal-length segments with II model (Chu & Wong, 2001) and parameterized with wire parasitics extracted from the physical layout. For each interconnect net^i , we define the wire resistance as R_w^i and wire capacitance as C_w^i for every segment, while \hat{C}_s^{ij} denotes the coupling capacitance between net^i and net^j at segment s .

Given a configuration of M coupled interconnects $\{net^i\}_{i=1}^M$, as shown in Figure 2, we represent its equivalent RC circuit as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Here, \mathcal{V} denotes the vertex set and \mathcal{E} denotes the edge set. Each net net^i is a node subset $\mathcal{V}_S^i = \{v_i^0, v_i^1, v_i^2, \dots, v_i^L\} \subset \mathcal{V}$, where L is the number of length-equivalent wire segments. Nodes in \mathcal{V}_S^i are connected in their physical sequence along net^i and

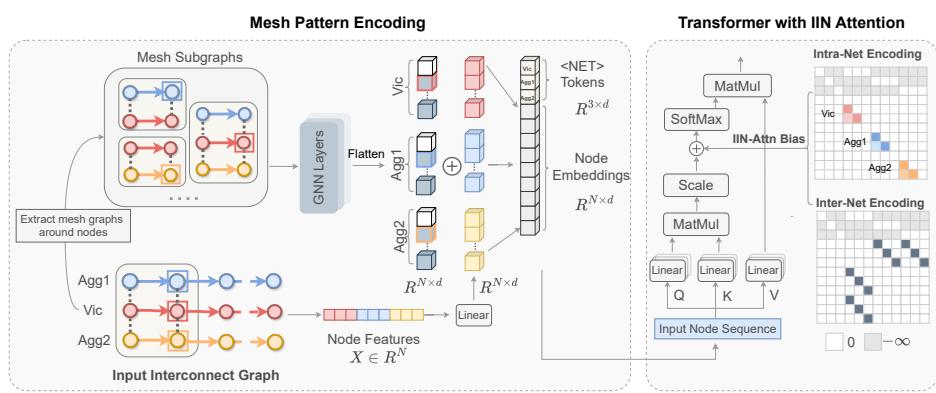


Figure 3: Overview of Si-GT.

characterized by the wire capacitance C_w , and each edge is associated with the wire resistance R_w^i . Edges in the graph represent two types of connections: intra-net connections between nodes within the individual net, and inter-net connections between nodes of different nets connected with coupling capacitors. For intra-net connections, we assign $[R_w, 0]$ as the edge feature of wire connection, while for inter-net connections, we assign $[0, \hat{C}]$ as the edge feature of coupling connection. More graph construction methods are discussed in Appendix B.

3.2 PROBLEM DEFINITION

Crosstalk problems concerning the signal integrity in integrated circuits manifests through three fundamental scenarios: (1) victim net quiet; (2) victim net active and switching in the opposite direction to the aggressor; (3) victim net active and switching in the same direction to the aggressor. Given the above critical scenarios for signal integrity analysis, we address two prediction tasks:

Task1: Crosstalk Glitch Prediction. As shown in Figure 1, when the aggressor is switching and the victim is quiet, the potential difference across the coupling capacitor will generate a leakage current flowing to the victim, resulting an undesirable raising or falling glitch on the victim. The height and width of a glitch indicate the potential severity of a crosstalk event. Therefore, for the quiet victim case (1), we estimate two key parameters of crosstalk noise at each interconnect segment s on the victim: the peak voltage v_{\max}^s and noise width t_{width}^s . Here, noise width is defined as the time interval between the glitch's rising and falling edges at the 50% of its peak voltage.

Task2: Crosstalk Delay Prediction. As shown in Figure 1, when both aggressor and victim are switching, the coupling capacitor will affect the signal transition on victim, leading to slower or faster transition time by Δt . Net delay D is defined as the time delay of a voltage waveform propagating through the net measured at the 50% of the waveform's voltage level. With crosstalk-induced variation, the net delay becomes $\hat{D} = D \pm \Delta t$. In scenarios (2) and (3) where both aggressor and victim nets are actively switching, we predict the net delay \hat{D}_i^s of the signal propagating through each segment s along the net ^{i} .

4 SI-GT

In this section, we present Si-GT framework. As shown in Figure 3, Si-GT incorporates several key designs, including mesh pattern encoding, virtual NET tokens, and intra-inter net (IIN) attention. To integrate the structural information of interconnect graphs into the Transformer model and account for circuit-specific behaviors under crosstalk effects, we first decompose the interconnect graph into local mesh structures at each node and encode these structures using GNN layers for absolute positional encoding. Since the driving signals propagating along nets present different characteristics, we introduce virtual NET tokens to represent individual nets and encode the net-level information such as net state, slew rate, and switching direction. Additionally, to structurally differentiate nodes within a specific net and coupled from different nets, we introduce IIN attention bias into the self-attention mechanism of Transformer to further improve the graph inductive bias.

216 4.1 MESH PATTERN ENCODING
217

218 In the equivalent RC circuit, coupling capacitors connect pairs of wire segments, forming a mesh
219 unit defined as follows: [Couple Mesh Unit.] A couple mesh unit at a node v_i^s on net^i is defined
220 as a subgraph of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with a set of nodes $\{v_i^{s-1}, v_i^s, v_j^{s-1}, v_j^s\}$ in direction from source to sink,
221 assuming net^i is coupled with an adjacent wire net^j with coupling capacitance \hat{C}_s^{ij} . In coupled
222 interconnects, we define mesh units to represent the coupling interactions between aligned segment
223 pairs on aggressor and victim nets. As illustrated in Figure 3, for a node v_i^s located at the end of
224 the s -th segment on net^i , the number of mesh units constructed at v_i^s is determined by the number
225 of couplings with other nets. For example, if net^i is coupled with two nets net^j and net^k at the
226 s -th segment, we construct a subgraph $mesh(v_i^s)$ to capture its local structural information with
227 two mesh units: $\{v_i^{s-1}, v_i^s, v_j^{s-1}, v_j^s\}$ and $\{v_i^{s-1}, v_i^s, v_k^{s-1}, v_k^s\}$. As shown in Figure 2, an aggressor
228 net is typically coupled with a single victim net, while a victim net may be coupled with multiple
229 aggressor nets. Therefore, we decompose the interconnect graph into local mesh subgraphs at each
230 node, enabling the model to capture each node’s local neighborhood while preserving the separation
231 of uncoupled nets. Since mesh subgraphs are small, we then employ a shallow GNN model with l
232 layers GNN l to aggregate the local mesh structure information as the embedding of v_i^s , effectively
233 encoding high-order mesh structural information into the node features. Finally, we add the GNN l
234 embeddings to the linear projected node feature as the input to Transformer encoder:

$$235 h^{(0)}(v_i^s) = \text{GNN}^l(mesh(v_i^s)) + en(x(v_i^s)) \in \mathbb{R}^d \quad (1)$$

236 Since the interconnect is decomposed at the end node of each wire segment, we initialize the em-
237 beddings of driving nodes (i.e., the starting nodes of each net) to a zero vector: $h^{(0)}(v_i^0) = \mathbf{0} \in \mathbb{R}^d$.
238

239 4.2 INTRA-INTER NET ATTENTION MECHANISM
240

241 Connections between nodes on each individual net are termed intra-net connections, while those
242 linking a pair of coupled nets are referred to as inter-net connections. Both types play a critical
243 role in the crosstalk effect. Intra-net connections capture incremental signal distortions and noise
244 transformations along the net, whereas inter-net connections provide pathways for signal energy
245 to transfer between nets. To capture this structural information, we introduce IIN-Attn, a novel
246 attention mechanism that incorporates both intra-net and inter-net connections through specialized
247 attention biases. First, as the signal is propagating forward on a single net, both the net delay and
248 crosstalk noise attributes at any specific node are highly dependent on its former nodes that the signal
249 has passed through. To this end, we design an intra-net encoding $\phi_{Intra}(v_i^u, v_i^v) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ to
capture the net structural feature and relative position between intra-nodes:

$$250 \phi_{Intra}(v_i^u, v_i^v) = \begin{cases} \frac{1}{d_{uv} \cdot R_w^i}, & \text{if } \{v_i^u, v_i^{u+1}, \dots, v_i^v\} \subseteq \mathcal{V}_S^i, \\ 251 0, & \text{otherwise.} \end{cases} \quad (2)$$

252 here, $d_{uv} = |v - u|$ denotes the distance from v_i^u to v_i^v along the net. $\phi_{Intra}(v_i^u, v_i^v)$ aggregates
253 the wire resistance along the path from u to v on a net. If u and v are not from the same net, we
254 set the value to be 0. The intra-net encoding explicitly captures the relative positional information
255 of nodes connected in a net. Second, considering the interconnections between coupled nets, for
256 net^i corresponding to node set \mathcal{V}_S^i and net^j corresponding to node set \mathcal{V}_S^j , we define function
257 $\phi_{Inter}(v_i^u, v_j^u) : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$:

$$258 \phi_{Inter}(v_i^u, v_j^u) := \begin{cases} \hat{C}_{u+1}^{ij}, & \text{if } net^i, net^j \text{ are coupled at } (u+1)\text{-th segment,} \\ 259 0, & \text{otherwise.} \end{cases} \quad (3)$$

260 $\phi_{Inter}(v_i^u, v_j^u)$ measures the coupling capacitance between v_i^u and v_j^u when the $(u+1)$ -th net seg-
261 ment of net^i and net^j are coupled; otherwise, this value is set to 0. Intuitively, the inter-net encoding
262 captures the connections between coupled net segments.

263 To encode structural information of coupled interconnects into attention layers, we directly incorpo-
264 rate the intra-net and inter-net biases into the attention logits:

$$265 \text{Attn-IIN}(X) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_K}} + \tilde{\Phi}_{IIN} + \tilde{\Phi}_d + \tilde{\Phi}_{sp} \right) V, \quad (4)$$

270 where the bias matrix $\tilde{\Phi}_{\text{IIN}}$ has entries given by: $\tilde{\phi}_{\text{IIN}} = \tilde{\phi}_{\text{Intra}} + \tilde{\phi}_{\text{Inter}}$. Here, $\tilde{\phi}_{\text{Intra}}$ and $\tilde{\phi}_{\text{Inter}}$ are
 271 obtained by applying learnable linear transformations to ϕ_{Intra} and ϕ_{Inter} respectively. **Additionally, to capture global typological features, we use spatial encoding and edge encoding as extra attention**
 272 **bias terms to the attention module.** Specifically, $\tilde{\Phi}_{\text{d}}$ encodes the distance of the shortest path (SP)
 273 between two connected nodes using learnable embedding table indexed by the distance scalar, while
 274 $\tilde{\Phi}_{\text{sp}}$ encodes the edge features along the path $\text{SP}_{ij} = (e_1, e_2, \dots, e_n)$ from node i to j via $\phi_{\text{sp}}(i, j) =$
 275 $\frac{1}{n} \sum_{k=1}^n e_k w_k^T$. Here, e is the edge feature, and $w \in \mathbb{R}^e$ is the weight embedding with edge feature
 276 dimension \mathbb{R}^e (Ying et al., 2021).

279 4.3 VIRTUAL NET TOKEN

280 As signals propagate along interconnects from source to sink, local electromagnetic interference
 281 between adjacent wire segments not only affects signal integrity at individual segments but also
 282 accumulates, leading to significant distortions at the sinks of all nets. To capture this global net-level
 283 interaction, we introduce virtual $\langle \text{NET} \rangle$ tokens that represent individual nets and attend to all nodes
 284 in the self-attention mechanism. Besides, for net-level attributes such as the switching direction
 285 and slew rate of signals propagating on each net, Si-GT encodes these features into learnable
 286 embeddings of $\langle \text{NET} \rangle$ tokens. Specifically, for each distinct net, we assign a learnable embedding
 287 vector $h_{\langle \text{NET} \rangle}^{(0)} \in \mathbb{R}^d$ as the input embedding for the special $\langle \text{NET} \rangle$ node. These embeddings are
 288 then processed alongside other node features within the transformer architecture. To restrict the
 289 receptive field of $\langle \text{NET} \rangle$ token to its corresponding net, as illustrated in Figure 3, we define an
 290 attention mask $\mathbf{M}_{\text{NET}} \in \mathbb{R}^{|V| \times |V|}$ applied to the softmax logits of the IIN attention:
 291

$$\mathbf{M}_{\text{NET}}(i, j) := \begin{cases} -\infty, & \text{if } i \text{ represents net}^i \text{ and } j \notin \mathcal{V}_S^i, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

295 \mathbf{M}_{NET} ensures that each $\langle \text{NET} \rangle$ node aggregates information exclusively from nodes within its re-
 296 spective net while remaining visible to all other nodes.

297 5 EXPERIMENTS

300 5.1 SIGNAL INTEGRITY DATASET

301 To benchmark graph learning-based models for signal integrity analysis, we construct a dataset
 302 elaborating on crosstalk delay and glitch prediction tasks that covers various net lengths and signal
 303 characteristics. Our dataset is based on the circuits of two aggressors and one victim. To simulate
 304 the circuit behavior of coupled interconnects, we construct RC circuits of the interconnect wires for
 305 SPICE simulation. In practice, circuit simulation begins by converting a physical layout into an RC
 306 model through parasitic extraction, where wires are divided into multiple equal-length segments
 307 and replaced with RC networks. To model varying interconnect lengths, we sweep the number
 308 of segments and follow Intel’s 14 FinFET (Fischer et al., 2015) to set the wire capacitance and
 309 resistance for every segment. Additionally, we sweep other key parameters such as wire separation,
 310 input slew rate, and signal switching direction to create diverse signal and coupling configurations,
 311 as summarized in Table 1. More details on coupling capacitance calculation, circuit simulation, and
 312 dataset construction pipeline are provided in the Appendix A.1. For each setup of RC circuit and its
 313 driving signals, we use Synopsys HSPICE simulator to measure the crosstalk delay and glitch along
 314 the nets, which results in 200,200 delay examples and 187,309 glitch examples in total.

315 316 Table 1: Circuit parameters of signal integrity dataset.

317 Fixed Parameters			318 Sweeping Parameters					
319 Segment Length	320 Wire Resistance	321 Wire Capacitance	322 Net Length	323 Wire Separation	324 Coupling Capacitance	325 Victim State	326 Switching Direction	327 Slew Rate
328 5 μm	329 2.7 $\Omega/\mu\text{m}$	330 0.15fF/ μm	331 10-100 μm	332 1-20 μm	333 0.2214-7.908fF	334 Active/Quiet	335 Low-To-High/High-To-Low	336 40-60ps

321 5.2 EXPERIMENTAL SETTINGS

322 **Baselines.** We compare our Si-GT model with following baselines: 1) GNNs, including standard
 323 GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), GIN (Xu et al., 2018), GraphSAGE

(Hamilton et al., 2017), and advanced DeepGCN (Li et al., 2019) with residual connections; 2) recent SOTA graph transformers, including Graphomer (Ying et al., 2021), GraphGPS (Rampášek et al., 2022), and SGFormer (Wu et al., 2023); and 3) variations of Si-GT using different standard GNN backbones for mesh pattern encoding. We also evaluate Si-GT against baseline models using different position/ structural encodings (PE/SE), such as RWSE and LapPE (Dwivedi et al., 2021), as detailed in Appendix D.1.

Settings. In the main results, we use 5 convolutional layers for standard GNNs and 20 layers for DeepGCN. For Graphomer, we follow its original configuration with a 5-step limit for shortest path encoding, while GraphGPS is implemented with RWSE using 16 walk length. Full implementation details and experimental tests for hyperparameter setting of all baseline models are provided in Appendix C.3. For Si-GT, we use $l = 2$ GNN layers with a hidden dimension of 64 to encode the mesh patterns. We use 6 encoder layers with 4 attention heads and set the embedding size to 64 for the self-attention module and 128 for the feed-forward network. We train our Si-GT for 60 epochs with 256 batch size using the AdamW optimizer with polynomial learning rate decay and linear warmup, where the learning rate decays to 1e-9 over the total training steps, with weight decay set to 1e-4. All experiments in this paper are implemented with PyTorch 2.2.2, DGL 2.4.0, and Pytorch-geometric 2.6.1. Models are trained with $2 \times$ NVIDIA A100 80GB GPUs. Detailed training setup of baseline models are in Appendix C.1 due to space constraints.

5.3 EXPERIMENTAL RESULTS

Table 2: Mean relative accuracy (%) of crosstalk delay prediction results.

Experiment	Metric	GNNs					Graph Transformer						
		GCN	GAT	GIN	SAGE	DeepGCN	SGFormer	Graphomer	GraphGPS	Si-GT GCN	Si-GT GAT	Si-GT GIN	Si-GT SAGE
AV Segment	\hat{D}_{vic}	65.21	58.68	60.02	58.50	85.49	64.64	88.23	88.23	88.32	88.28	88.27	88.28
	\hat{D}_{agg}	57.38	43.27	54.17	62.41	71.64	52.60	72.58	72.65	73.67	73.18	73.30	73.81
AV Sink	\hat{D}_{vic}	51.14	45.96	51.12	46.67	50.17	53.63	86.52	87.36	87.38	87.39	87.33	87.31
	\hat{D}_{agg}	39.72	35.34	45.12	47.58	35.11	44.58	71.02	70.65	71.17	71.82	71.60	71.05
V Segment	\hat{D}_{vic}	64.03	60.90	62.64	64.09	86.90	59.51	88.15	88.26	88.31	88.30	88.27	88.34
	\hat{D}_{agg}	51.97	42.77	43.65	43.59	43.89	55.53	87.11	87.18	87.19	87.21	87.38	87.20

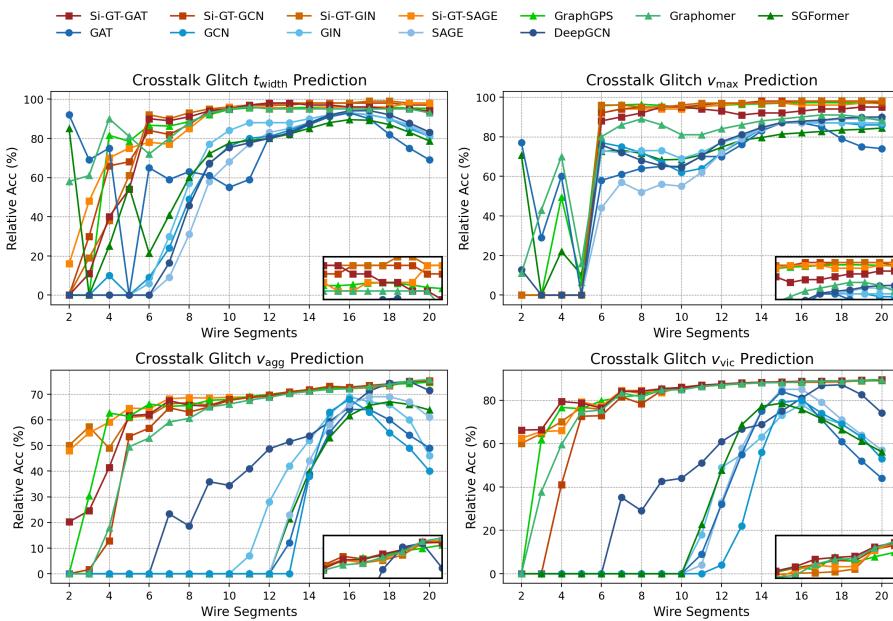
Table 3: Mean relative accuracy (%) of crosstalk glitch prediction results.

Experiment	Metric	GNNs					Graph Transformer						
		GCN	GAT	GIN	SAGE	DeepGCN	SGFormer	Graphomer	GraphGPS	Si-GT GCN	Si-GT GAT	Si-GT GIN	Si-GT SAGE
V Segment	t_{width}	87.86	88.38	87.01	88.23	87.94	84.17	94.97	96.61	97.71	97.08	98.36	97.47
	v_{max}	85.44	85.79	85.45	85.65	85.20	82.84	93.38	97.99	97.89	96.62	97.78	96.89
V Sink	t_{width}	83.97	84.05	83.85	84.01	83.99	83.72	95.46	98.29	98.53	97.83	98.13	98.19
	v_{max}	82.61	83.10	82.42	82.68	82.56	79.08	94.17	97.94	98.63	97.16	97.62	97.96

Main Results. We first report the main experimental results for crosstalk delay prediction in Table 2 and crosstalk glitch prediction in Table 3. Models are separately trained to predict delay and glitch metrics at each segment (Segment) along individual nets and specifically at their sinks (Sink). The sink-level results can provide an overview of model performance in predicting pin-to-pin delay and glitch. For delay prediction, we report accuracy for both aggressor and victim (AV) cases, as well as for victim-only (V) cases, since the victim is of greater concern in ensuring signal integrity. Predictions are evaluated against SPICE ground truth using mean relative accuracy.

The results show that: (1) Graph transformer models, particularly Graphomer, GraphGPS, and our Si-GT, consistently outperform traditional GNNs in signal integrity analysis, achieving significantly higher accuracy across both delay and glitch prediction tasks; and (2) for the more challenging delay prediction task, our proposed Si-GT model outperforms all baselines across all experiments. While GraphGPS also demonstrates strong performance, Si-GT variants consistently achieve the highest mean relative accuracy in nearly every case.

Accuracy with Interconnect Length. To illustrate the performance of our model across different interconnect scales, Figure 4 presents the model prediction accuracy with the number of wire segments of RC circuits for all prediction tasks. From the results, we can observe that: (1) Traditional GNNs exhibit notably lower accuracy for all tasks, with performance degrading on long interconnects, highlighting their inherent limitations in capturing long-range interactions critical for signal

398
399 Figure 4: Comparison of models in signal integrity analysis under various IC interconnect lengths.
400

401 integrity analysis. (2) While DeepGCN shows improved accuracy on longer interconnects, its
402 generalization across varying interconnect lengths remains limited. (3) Transformer-based models,
403 particularly our Si-GT, achieve consistently higher accuracy and demonstrate robust performance even
404 on longer interconnects. (4) All models struggle to generalize effectively to small interconnects. We
405 analyze that short interconnects present less coupling variety, reflected in fewer data examples in
406 dataset (detailed in Appendix A.2), resulting in poor generalization to those sparse examples.

407 Segment Models in Sink-level Prediction.

408 Table 4 compares the performance of segment models (trained with segment data) and
409 sink models (trained with sink data) on sink-level prediction tasks by evaluating their
410 differences in prediction accuracy. The results show
411 that: (1) for victim delay prediction, training
412 Transformer-based models with segment data
413 can improve sink-level predictions compared
414 to solely with sink data, while for other tasks,
415 training with sink data can yield better results; and (2) compared to baselines, the segment-trained
416 Si-GT exhibits the smallest performance variation, demonstrating its robustness and adaptability to
417 sink-level prediction, highlighting the robustness of Si-GT in capturing complex crosstalk behaviors
418 even with limited structural context.

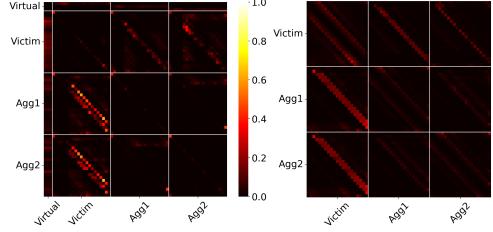
419
420 **Ablation Study.** We evaluate the impact of core design components in Si-GT through ablation
421 studies, with results summarized in Table 5. The components under investigation include the intro-
422 duction of virtual `<Net>` tokens (NET), mesh pattern encoding (MPE), and intra-inter net attention
423 (IIN). When MPE is removed, we use centrality encoding of Graphomer to construct the input node
424 features. In the absence of IIN, we only adopt the spatial and edge encoding of Graphomer for the
425 attention bias in equation 1. [More implementation details and fine-grained ablation experiments](#)
426 [are provided in Appendix D.3](#). Our ablation study shows the critical importance of virtual `<Net>`
427 nodes to Si-GT in all prediction tasks, as it yields a large margin performance boost in comparison
428 with other modules. For crosstalk delay prediction, MPE particularly shows the impact to aggres-
429 sor delay prediction. Additionally, IIN attention mechanism combining both intra- and inter-net
430 attention, consistently enhances accuracy across most tasks, which indicates that incorporating IIN
431 encoding as an additional attention bias effectively enables the Transformer to capture the structural
432 characteristics of coupled interconnects.

433 Table 4: Accuracy comparison of segment and
434 sink models in sink-level prediction tasks.

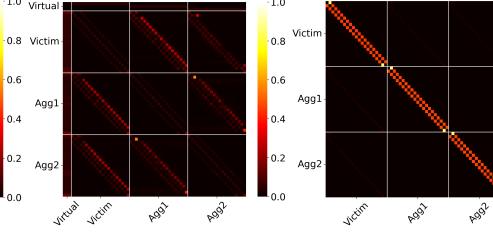
Model	Sink Delay		Sink Glitch	
	$\Delta \hat{D}_{\text{vic}}$	$\Delta \hat{D}_{\text{agg}}$	Δt_{width}	Δv_{max}
DeepGCN	+6.36	+11.48	-2.32	+0.51
SGFormer	-2.30	-12.86	-3.13	-6.08
Graphomer	+0.81	-1.02	-1.73	-5.83
GraphGPS	+0.48	+0.89	-0.34	-1.35
Si-GT-GCN	+0.08	-1.42	-0.12	-0.18

Table 5: Ablation study results on crosstalk prediction with different designs.

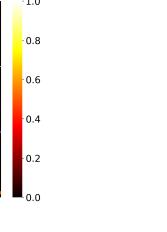
434	Module				Delay Prediction				Glitch Prediction				
	435	NET	MPE	IIN	IIN	Segment		Sink		Segment		Sink	
				ϕ_{Intra}	ϕ_{Inter}	\hat{D}_{vic}	\hat{D}_{agg}	\hat{D}_{vic}	\hat{D}_{agg}	t_{width}	v_{max}	t_{width}	v_{max}
437		x	x	x	x	88.23	72.58	86.52	71.02	94.97	89.49	95.46	94.17
438		✓	x	x	x	88.28	73.30	87.34	71.04	98.12	97.70	97.92	97.57
439		✓	✓	x	x	88.25	73.48	87.34	71.93	98.18	97.85	98.44	97.90
440		✓	✓	✓	x	88.22	73.40	87.29	71.06	98.12	97.83	97.98	97.44
441		✓	✓	x	✓	88.27	73.66	87.26	70.87	97.97	97.39	97.99	97.68
442		✓	✓	✓	✓	88.32	73.67	87.39	71.82	98.36	97.78	98.53	98.63



(a) Delay Si-GT



(b) Delay Graphomer



(c) Glitch Si-GT



(d) Glitch Graphomer

Figure 5: Comparison of attention maps between Si-GT and Graphomer.

Attention Visualization. GraphGPS applies global attention after local message passing updates, while Graphomer and our Si-GT directly integrate structural information into attention. We compare the learned attention maps of Si-GT and Graphomer in Figure 5. For delay prediction, without explicitly encoding the coupling patterns into attention bias, Graphomer (Figure 5b) shows strong attention among coupled segments, highlighting the structural importance of coupling in signal integrity analysis concerning crosstalk effect. Compared with Graphomer, Si-GT (Figure 5a) further enables the isolation of two aggressors, aligning with the fact that aggressors are not coupled with coupling capacitors in physical layout. For glitch prediction, the attention map of Si-GT (Figure 5c) shows clear coupling pattern, while Graphomer (Figure 5d) only concentrates on the neighbor nodes of the same net, limiting its ability to model noise propagation across coupled nets.

Computation Efficiency. We compare the computational efficiency of promising Transformer-based model Graphomer, GraphGPS, and our Si-GT model against SPICE simulation in this section. All reported runtimes are measured on CPU. Details of the computing environment and additional runtime benchmarks across different hardware platforms are provided in Appendix C.2. As shown in Figure 6, the computational cost of SPICE increases substantially with interconnect length, while Transformer-based models maintain consistently low inference times. On average, Graphomer, GraphGPS, and Si-GT achieve inference times of 2.4 ms, 6.8 ms, and 4.0 ms, respectively, compared to over 100 ms required by SPICE even for short interconnects, highlighting the practicality of transformer-based models as efficient and scalable alternatives for signal integrity analysis in large-scale IC designs.

6 CONCLUSION

In this paper, we propose Si-GT, a Transformer-based model for signal integrity analysis of IC interconnects, and construct the first large-scale benchmark dataset comprising crosstalk prediction tasks relevant to practical SI challenges. We demonstrate that Si-GT consistently outperforms state-of-the-art GNN and GT baselines across nearly all tasks, while significantly reducing runtime compared to SPICE. These results highlight the strong potential of Si-GT as an efficient surrogate for interconnect signal integrity analysis to accelerate IC design verification.

486 REFERENCES
487

488 Ramachandra Achar and Michel S Nakhla. Simulation of high-speed interconnects. *Proceedings of*
489 *the IEEE*, 89(5):693–728, 2001.

490 Xavier Aragones and Antonio Rubio. Challenges for signal integrity prediction in the next decade.
491 *Materials Science in Semiconductor Processing*, 6(1-3):107–117, 2003.

492

493 Fabrice Caignet, Sonia Delmas-Bendhia, and Etienne Sicard. The challenge of signal integrity in
494 deep-submicrometer cmos technology. *Proceedings of the IEEE*, 89(4):556–573, 2001.

495

496 Hsien-Han Cheng, Iris Hui-Ru Jiang, and Oscar Ou. Fast and accurate wire timing estimation on
497 tree and non-tree net structures. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*,
498 pp. 1–6. IEEE, 2020.

499

500 Ruoyu Cheng and Junchi Yan. On joint learning for solving placement and routing in chip design.
501 *Advances in Neural Information Processing Systems*, 34:16508–16519, 2021.

502

503 Chris Chu and DF Wong. Vlsi circuit performance optimization by geometric programming. *Annals*
504 *of Operations Research*, 105:37–60, 2001.

505

506 Wenjie Ding, Zhanhua Zhang, Guoqing He, and Peng Cao. A physical and timing aware placement
507 optimization framework based on graph neural network. In *Proceedings of the 43rd IEEE/ACM*
508 *International Conference on Computer-Aided Design*, pp. 1–9, 2024.

509

510 Chunjie Duan, Brock J LaMeres, and Sunil P Khatri. *On and off-chip crosstalk avoidance in VLSI*
511 *design*. Springer, 2010.

512

513 Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs.
514 *arXiv preprint arXiv:2012.09699*, 2020.

515

516 Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson.
517 Graph neural networks with learnable structural and positional representations. *arXiv preprint*
518 *arXiv:2110.07875*, 2021.

519

520 K. Fischer, M. Agostinelli, C. Allen, D. Bahr, M. Bost, P. Charvat, V. Chikarmane, Q. Fu, C. Gan-
521 pule, M. Haran, M. Heckscher, H. Hiramatsu, E. Hwang, P. Jain, I. Jin, R. Kasim, S. Kosaraju,
522 K. S. Lee, H. Liu, R. McFadden, S. Nigam, R. Patel, C. Pelto, P. Plekhanov, M. Prince, C. Puls,
523 S. Rajamani, D. Rao, P. Reese, A. Rosenbaum, S. Sivakumar, B. Song, M. Uncuer, S. Williams,
524 M. Yang, P. Yashar, and S. Natarajan. Low-k interconnect stack with multi-layer air gap and
525 tri-metal-insulator-metal capacitors for 14nm high volume manufacturing. In *2015 IEEE Interna-
526 tional Interconnect Technology Conference and 2015 IEEE Materials for Advanced Metallization*
527 *Conference (IITC/MAM)*, pp. 5–8, 2015. doi: 10.1109/IITC-MAM.2015.7325600.

528

529 Tong Gao and CL Liu. Minimum crosstalk channel routing. *IEEE Transactions on Computer-Aided*
530 *Design of Integrated Circuits and Systems*, 15(5):465–474, 1996.

531

532 Xiang Gao, Yi-Min Jiang, Lixin Shao, Pedja Raspopovic, Menno E Verbeek, Manish Sharma, Vi-
533 neet Rashingkar, and Amit Jalota. Congestion and timing aware macro placement using machine
534 learning predictions from different data sources: Cross-design model applicability and the dis-
535 cerning ensemble. In *Proceedings of the 2022 International Symposium on Physical Design*, pp.
536 195–202, 2022.

537

538 Jingjing Guo, Xuejie Ning, Chenfei Hua, Jun Yang, and Zhikuang Cai. A path statistical delay
539 prediction framework based on global graph neural network. *IEEE Transactions on Circuits and*
540 *Systems I: Regular Papers*, 2025.

541

542 Zizheng Guo, Mingjie Liu, Jiaqi Gu, Shuhan Zhang, David Z Pan, and Yibo Lin. A timing engine
543 inspired graph neural network model for pre-routing slack prediction. In *Proceedings of the 59th*
544 *ACM/IEEE Design Automation Conference*, pp. 1207–1212, 2022.

545

546 Stephen H Hall and Howard L Heck. *Advanced signal integrity for high-speed digital designs*. John
547 Wiley & Sons, 2011.

540 Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs.
 541 *Advances in neural information processing systems*, 30, 2017.
 542

543 REN Haoxiang, S NATH, ZHANG Yanqing, et al. Why are graph neural networks effective for eda
 544 problems. 2022.

545 Yunbo Hou, Haoran Ye, Shuwen Yang, Yingxue Zhang, Siyuan Xu, and Guojie Song. Transplace:
 546 Transferable circuit global placement via graph neural network. *arXiv preprint arXiv:2501.05667*,
 547 2025.

548 Yuting Hu, Jiajie Li, Florian Klemme, Gi-Joon Nam, Tengfei Ma, Hussam Amrouch, and Jinjun
 549 Xiong. SyncTree: fast timing analysis for integrated circuit design through a physics-informed
 550 tree-based graph neural network. *Advances in Neural Information Processing Systems*, 36:21415–
 551 21428, 2023.

552 Leilei Jin, Jiajie Xu, Wenjie Fu, Hao Yan, and Longxing Shi. A crosstalk-aware timing prediction
 553 method in routing. *arXiv preprint arXiv:2403.04145*, 2024.

554

555 Zhong-Fang Jin, J-J Laurin, Yvon Savaria, and Pierre Garon. A new approach to analyze intercon-
 556 nect delays in rc wire models. In *1999 IEEE International Symposium on Circuits and Systems*
 557 (ISCAS), volume 6, pp. 246–249. IEEE, 1999.

558

559 Andrew B Kahng, Mulong Luo, and Siddhartha Nath. Si for free: machine learning of interconnect
 560 coupling delay and transition effects. In *2015 ACM/IEEE International Workshop on System Level*
 561 *Interconnect Prediction (SLIP)*, pp. 1–8. IEEE, 2015.

562

563 Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional net-
 564 works. *arXiv preprint arXiv:1609.02907*, 2016.

565

566 Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Re-
 567 thinking graph transformers with spectral attention. *Advances in Neural Information Processing*
 568 *Systems*, 34:21618–21629, 2021.

569

570 Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep
 571 as cnns? In *Proceedings of the IEEE/CVF international conference on computer vision*, pp.
 572 9267–9276, 2019.

573

574 Shenggao Li, Mu-Shan Lin, Wei-Chih Chen, and Chien-Chun Tsai. Interconnect in the era of 3dic.
 575 In *2022 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–5. IEEE, 2022.

576

577 Rongjian Liang, Zhiyao Xie, Jinwook Jung, Vishnavi Chauha, Yiran Chen, Jiang Hu, Hua Xiang,
 578 and Gi-Joon Nam. Routing-free crosstalk prediction. In *Proceedings of the 39th International*
579 Conference on Computer-Aided Design, pp. 1–9, 2020.

580

581 Rongjian Liang, Zhiyao Xie, Erick Carvajal Barboza, and Jiang Hu. Net-based machine learning-
 582 aided approaches for timing and crosstalk prediction. *Machine Learning Applications in Elec-
 583 tronic Design Automation*, pp. 63–84, 2022.

584

585 Haiguang Liao, Wentai Zhang, Xuliang Dong, Barnabas Poczos, Kenji Shimada, and Levent Bu-
 586 rak Kara. A deep reinforcement learning approach for global routing. *Journal of Mechanical*
587 Design, 142(6):061701, 2020.

588

589 Zihao Lin, Haisen Zhang, Peng Gao, Fei Yu, Tingting Wu, Xiaoming Xiong, and Shuting Cai. Gnn-
 590 based timing prediction in pre-routing stage with multi-task learning strategy. *IEEE Transactions*
591 on Computer-Aided Design of Integrated Circuits and Systems, 2025.

592

593 Fangzhou Liu, Guannan Guo, Yuyang Ye, Ziyi Wang, Wenjie Fu, Weihua Sheng, and Bei Yu.
 594 Graphcad: Leveraging graph neural networks for accuracy prediction handling crosstalk-affected
 595 delays. In *Proceedings of the 2025 International Symposium on Physical Design*, ISPD ’25,
 596 pp. 125–133, New York, NY, USA, 2025. Association for Computing Machinery. ISBN
 9798400712937. doi: 10.1145/3698364.3705345. URL <https://doi.org/10.1145/3698364.3705345>.

594 Lihao Liu, Fan Yang, Li Shang, and Xuan Zeng. Gnn-cap: Chip-scale interconnect capacitance ex-
 595 traction using graph neural network. *IEEE Transactions on Computer-Aided Design of Integrated*
 596 *Circuits and Systems*, 43(4):1206–1217, 2023.

597

598 Yi-Chen Lu and Sung Kyu Lim. On advancing physical design using graph neural networks. In
 599 *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pp.
 600 1–7, 2022.

601

602 Yi-Chen Lu, Sai Pentapati, and Sung Kyu Lim. Vlsi placement optimization using graph neural net-
 603 works. In *Proceedings of the 34th Advances in Neural Information Processing Systems (NeurIPS)*
 604 *Workshop on ML for Systems, Virtual*, pp. 6–12, 2020.

605

606 Yi-Chen Lu, Siddhartha Nath, Vishal Khandelwal, and Sung Kyu Lim. Rl-sizer: Vlsi gate sizing
 607 for timing optimization using deep reinforcement learning. In *2021 58th ACM/IEEE Design*
 608 *Automation Conference (DAC)*, pp. 733–738. IEEE, 2021.

609

610 Hongbin Pei, Yu Li, Huiqi Deng, Jingxin Hai, Pinghui Wang, Jie Ma, Jing Tao, Yuheng Xiong,
 611 and Xiaohong Guan. Multi-track message passing: Tackling oversmoothing and oversquashing
 612 in graph learning via preventing heterophily mixing. In *Forty-first International Conference on*
 613 *Machine Learning*.

614

615 Thomas Quarles, AR Newton, DO Pederson, and A Sangiovanni-Vincentelli. Spice 3 version 3f5
 616 user’s manual, 1994.

617

618 Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Do-
 619 minique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural*
 620 *Information Processing Systems*, 35:14501–14515, 2022.

621

622 Haoxing Ren, George F Kokai, Walker J Turner, and Ting-Sheng Ku. Paragraph: Layout parasitics
 623 and device parameter prediction using graph neural networks. In *2020 57th ACM/IEEE Design*
 624 *Automation Conference (DAC)*, pp. 1–6. IEEE, 2020.

625

626 T. Sakurai and K. Tamaru. Simple formulas for two- and three-dimensional capacitances. *IEEE*
 627 *Transactions on Electron Devices*, 30(2):183–185, 1983. doi: 10.1109/T-ED.1983.21093.

628

629 Aditya Shahane, Saripilli Swapna Manjiri, Ankesh Jain, and Sandeep Kumar. Graph of circuits with
 630 gnn for exploring the optimal design space. *Advances in Neural Information Processing Systems*,
 631 36:6014–6025, 2023.

632

633 Taigon Song, Chang Liu, Yarui Peng, and Sung Kyu Lim. Full-chip signal integrity analysis and
 634 optimization of 3-d ics. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24
 635 (5):1636–1648, 2015.

636

637 Tilmann Stöhr, Markus Alt, Asmus Hetzel, and Jürgen Koehl. Analysis, reduction and avoidance of
 638 crosstalk on vlsi chips. In *Proceedings of the 1998 international symposium on Physical design*,
 639 pp. 211–218, 1998.

640

641 Madhavan Swaminathan, Hakki Mert Torun, Huan Yu, Jose Ale Hejase, and Wiren Dale Becker. De-
 642 mystifying machine learning for signal and power integrity problems in packaging. *IEEE Trans-*
 643 *actions on Components, Packaging and Manufacturing Technology*, 10(8):1276–1295, 2020.

644

645 Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua
 646 Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

647

648 Ashok Vittal and Małgorzata Marek-Sadowska. Crosstalk reduction for vlsi. *IEEE transactions on*
 649 *computer-aided design of integrated circuits and systems*, 16(3):290–298, 1997.

650

651 Ashok Vittal, Lauren Hui Chen, Małgorzata Marek-Sadowska, Kai-Ping Wang, and Sherry Yang.
 652 Crosstalk in vlsi interconnections. *IEEE transactions on computer-aided design of integrated*
 653 *circuits and systems*, 18(12):1817–1824, 1999.

654

655 Hao Wang, Jun Tu, Shenglong Bai, Jie Zheng, Weikang Qian, and Jienan Chen. Routing genera-
 656 tive pre-trained transformers for printed circuit board. In *2024 2nd International Symposium of*
 657 *Electronics Design Automation (ISEDA)*, pp. 160–165. IEEE, 2024.

648 Laura Wang and Matt Luo. Machine learning applications and opportunities in ic design flow. In
 649 *2019 international symposium on VLSI design, automation and test (VLSI-DAT)*, pp. 1–3. IEEE,
 650 2019.

651 Shyh-Chyi Wong, Gwo-Yann Lee, and Dye-Jyun Ma. Modeling of interconnect capacitance, delay,
 652 and crosstalk in vlsi. *IEEE Transactions on semiconductor manufacturing*, 13(1):108–111, 2000.

653

654 Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and
 655 Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations.
 656 *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023.

657

658 Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A
 659 comprehensive survey on graph neural networks. *IEEE transactions on neural networks and*
 660 *learning systems*, 32(1):4–24, 2020.

661 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
 662 networks? *arXiv preprint arXiv:1810.00826*, 2018.

663

664 Yuyang Ye, Tinghuan Chen, Yifei Gao, Hao Yan, Bei Yu, and Longxing Shi. Fast and accurate
 665 wire timing estimation based on graph learning. In *2023 Design, Automation Test in Europe*
 666 *Conference Exhibition (DATE)*, pp. 1–6, 2023. doi: 10.23919/DATEn56975.2023.10137233.

667

668 Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and
 669 Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural*
 670 *information processing systems*, 34:28877–28888, 2021.

671

672 Jongho Yoon, Jakang Lee, Donggyu Kim, Junseok Hur, and Seokhyeong Kang. Paraformer: A
 673 hybrid graph neural network and transformer approach for pre-routing parasitic rc prediction. In
 674 *Proceedings of the 30th Asia and South Pacific Design Automation Conference*, pp. 513–519,
 675 2025.

676

677 Hong You and Mani Soma. Crosstalk analysis of interconnection lines and packages in high-speed
 678 integrated circuits. *IEEE transactions on circuits and systems*, 37(8):1019–1026, 1990.

679

680 Tao Yu, Peng Gao, Fei Wang, and Ru-Yue Yuan. Non-overlapping placement of macro cells based on
 681 reinforcement learning in chip design. *International Journal of Circuit Theory and Applications*,
 682 53(2):1159–1170, 2025.

683

684 Chaohao Yuan, Kangfei Zhao, Ercan Engin Kuruoglu, Liang Wang, Tingyang Xu, Wenbing Huang,
 685 Deli Zhao, Hong Cheng, and Yu Rong. A survey of graph transformers: Architectures, theories
 686 and applications. *arXiv preprint arXiv:2502.16533*, 2025.

687

688 Ruizhe Zhong, Junjie Ye, Zhentao Tang, Shixiong Kai, Mingxuan Yuan, Jianye Hao, and Junchi
 689 Yan. Preroutgnn for timing prediction with order preserving partition: Global circuit pre-training,
 690 local delay learning and attentional cell modeling. In *Proceedings of the AAAI Conference on*
 691 *Artificial Intelligence*, volume 38, pp. 17087–17095, 2024.

692

693

694 Xinyi Zhou, Junjie Ye, Chak-Wa Pui, Kun Shao, Guangliang Zhang, Bin Wang, Jianye Hao,
 695 Guangyong Chen, and Pheng Ann Heng. Heterogeneous graph neural network-based imitation
 696 learning for gate sizing acceleration. In *Proceedings of the 41st IEEE/ACM International Confer-
 697 ence on Computer-Aided Design*, pp. 1–9, 2022.

698

699

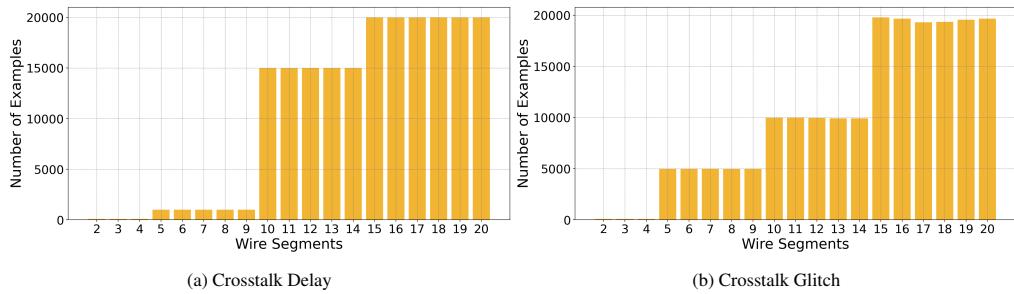
700

701

702 **A DATASET DETAILS**
703704 **A.1 DATASET PREPARATION**
705706 Given the separation between adjacent wires, as the length, width, and the relative electric permittiv-
707 ity ϵr of the substrate are constant per wire segment, we calculate the coupling capacitance according
708 to Eq.6, based on the two-dimensional capacitance formula from (Sakurai & Tamaru, 1983).

709
710
$$C_c = \epsilon r * \frac{\text{Segment Length of Wire} * \text{Wire Width}}{\text{Wire Separation}} \quad (6)$$

711

712 With the sweeping parameter setting in Table 1, we use Algorithm 1 to create the netlist of various
713 RC circuits modeling coupled interconnects. After the netlist is created, we supply a pulse input
714 with 0.7 magnitude to every active interconnect wire. SPICE simulation is carried out to measure
715 the voltage waveforms at each segment along individual nets. Only examples that successfully
716 complete SPICE simulations without failures are retained in the dataset.
717718 **Algorithm 1** Generate RC netlist for coupled interconnects
719720 **Require:** wr : wire resistance per micron, wc : wire capacitance per micron, $\{C_c\}$: set of coupling
721 capacitance values; $\{l\}$: wire segment length.
722 Sample number of segments $N \sim \mathcal{U}(2, 20)$
723 Initialize segment index $s \leftarrow 1$
724 Set wire segment length: $l \leftarrow 5\mu\text{m}$
725 **while** $s \leq N$ **do**
726 Set wire resistance: $R_w \leftarrow l \cdot wr$
727 Set wire capacitance: $C_w \leftarrow l \cdot wc$
728 Sample coupling capacitance between victim and aggressor 1: $\hat{C}_s \sim \text{Random Select}(\{C_c\})$
729 Sample coupling capacitance between victim and aggressor 2: $\hat{C}_s \sim \text{Random Select}(\{C_c\})$
730 $s \leftarrow s + 1$
731 **end while**
732
733734 **A.2 DATASET DISTRIBUTION**
735736 To show the composition of our signal integrity dataset, we analyze the distribution of examples
737 across different wire segments. Figure 7 illustrates the number of examples for both the crosstalk
738 delay (Figure 7a) and crosstalk glitch (Figure 7b) prediction tasks.750 **Figure 7: Distribution of the signal integrity dataset across wire segments.**
751752 **B GRAPH FEATURIZATION.**
753754 **Directed or Undirected.** As illustrated in Figure 2, we construct the interconnect graph by mod-
755 eling edges within each individual net as directed from source to sink. In this section, we evaluate

model performance using undirected interconnect graphs as input, aiming to assess the impact of edge directionality on prediction accuracy. We primarily report the prediction accuracy difference between undirected and directed settings on segment-level prediction tasks for both aggressor and victim nets for comparison. The results are reported in Table 6. A positive difference indicates improved performance with undirected graphs, while a negative difference suggests that maintaining directionality is beneficial for capturing the underlying signal behavior in the circuit.

Table 6: Comparison of segment-level prediction accuracy using directed vs. undirected interconnect graphs.

Model	Undirected			
	Segment Delay		Segment Glitch	
	$\Delta\hat{D}_{\text{vic}}$	$\Delta\hat{D}_{\text{agg}}$	Δt_{width}	Δv_{max}
GCN	+3.80	+3.62	+1.05	-0.10
DeepGCN	+7.98	+11.48	-0.10	-1.08
Graphomer	+1.89	+1.26	-0.82	-3.28
GraphGPS	-1.35	-0.28	+0.39	-1.53
Si-GT-GCN	-0.13	-0.20	+2.30	-0.16

C EXPERIMENT DETAILS

C.1 DETAILED EXPERIMENTAL SETTINGS

For the model training, we use different training schemes for the models included in our experiments:

Standard GNN models. We train the model using the Adam optimizer with a learning rate of 2e-3 and a weight decay of 6e-4. Models are trained with 256 batch size for 100 epochs.

DeepGCN. We train the model using the Adam optimizer with a learning rate of 1e-3 and a weight decay of 6e-4. Models are trained with 256 batch size for 100 epochs.

GraphGPS. We train the model using the Adam optimizer with a learning rate of 5e-4 and a weight decay of 1e-5. Models are trained with 256 batch size for 100 epochs.

SGFormer. We train the model using the Adam optimizer with a learning rate of 5e-5 and a weight decay of 1e-5. Models are trained with 256 batch size for 200 epochs.

Graphomer and Si-GT. We train the model using the AdamW optimizer with an initial learning rate of 1e-4, polynomial learning rate decay, and linear warmup, where the learning rate decays to 1e-9 over the total training steps, with weight decay set to 1e-4. Models are trained with 256 batch size for 60 epochs.

All experiments in this paper are implemented with PyTorch 2.2.2, DGL 2.4.0, and Pytorch-geometric 2.7.0.

C.2 COMPUTING ENVIRONMENT

All models are trained with 2× NVIDIA A100 80GB GPUs. SPICE simulations are carried out with the commercial Synopsys HSPICE simulator on an Intel Core i7-11700K Processor. In Figure 6, we report the running time of Transformer-based models executed on an Intel Xeon Gold 6448Y Processor. Additionally, we compare the running time of Graphomer, GraphGPS, and Si-GT on the A100 GPU in Figure 8. As our graph sizes are relatively small, GPU inference may exhibit higher latency due to kernel launch overhead and underutilization of GPU parallelism.

C.3 BASELINE IMPLEMENTATION.

For crosstalk delay prediction, the model outputs a single delay value, so the output dimension is set to 1. For crosstalk glitch prediction, we predict both the peak voltage and noise width, so the output dimension is set to 2. The model architectures of the baselines are summarized as follows:

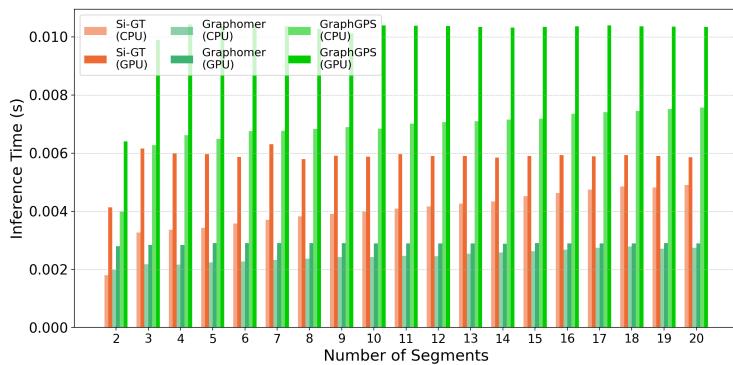


Figure 8: Model inference time.

Standard GNN models. We set convolutional layers to 5 and the hidden dimension to 64 for glitch prediction and 128 for delay prediction. Each convolutional layer is followed by ReLU activation and PairNorm normalization, except for the final layer.

DeepGCN. We set convolutional layers to 20 and the hidden dimension to 64 for glitch prediction and 128 for delay prediction. It stacks 20 DeepGCNLayer blocks, each composed of a GCNConv layer for message passing, LayerNorm for normalization, a ReLU activation, and 0.1 dropout. These blocks use a configurable residual connection strategy to enable stable training of deep GNNs (Li et al., 2019).

SGFormer. SGFormer integrates a GCN-based GraphModule and a Transformer-style SGModule. In our implementation, we set the hidden dimension to 64. The SGModule uses 2 transformer layers, each with 4 attention heads and 0.5 dropout, to model long-range interactions via dense attention. In parallel, the GraphModule applies 3 layers of GCNConv with dropout 0.5 and residual connections to extract localized features. The outputs from both modules are averaged to form the final node representation.

Graphomer. We use 6 encoder layers with a hidden dimension of 64 and 4 attention heads; each layer includes a feed-forward network with an embedding dimension of 128, and a dropout rate of 0.1 is applied after multi-head self-attention. Graphomer uses the shortest path between any pair of nodes for spatial and edge encoding. Follow (Ying et al., 2021), the length limit of the shortest path is set to 5 by default.

GraphGPS. For the PE/SE of GraphGPS, we use RSWE with a walk length of 16 by default. In our implementation, the input node features are projected to 64 dimensions, with 16 dimensions for PE/SE. The model stacks 10 GPSConv layers, each integrating a GINEConv-based local aggregator and multi-head attention mechanism with 4 heads to capture global interactions. A 3-layer MLP with decreasing dimensions is applied for the final prediction.

Si-GT. We use 6 encoder layers with a hidden dimension of 64 and 4 attention heads; each layer includes a feed-forward network with an embedding dimension of 128, and a dropout rate of 0.1 is applied after multi-head self-attention. For mesh pattern encoding, we use 2 convolutional layers (e.g., EGATConv, GraphConv, SAGEConv, GINConv in DGL.) with residual connection, and we set 0.2 dropout rate for node embeddings.

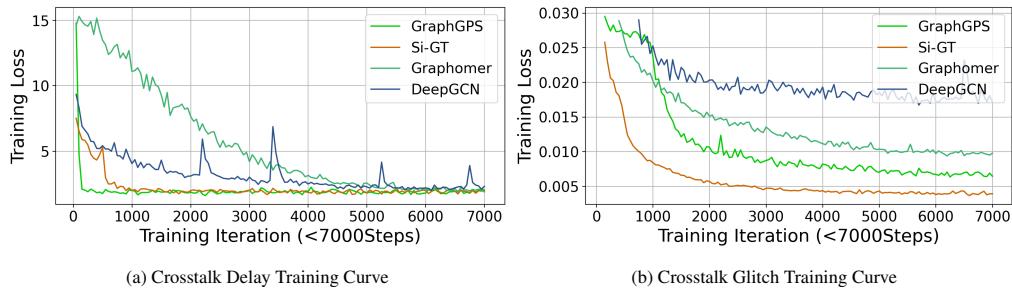
We report the trainable model parameters of all models in Table 7.

Table 7: Trainable parameter size of models.

Model	GCN	GAT	GIN	SAGE	DeepGCN	SGFormer	Graphomer	GraphGPS	Si-GT GCN	Si-GT GAT	Si-GT GIN	Si-GT SAGE
Parameter Size	49,921	52,486	115,971	99,329	85,953	210,561	273,261	422,417	282,029	306,733	282,029	290,221

864 C.4 TRAINING CURVE
865

866 We plot the training loss curves for crosstalk delay and glitch prediction tasks in Figure 9. Across
867 both tasks, Si-GT consistently achieves faster convergence and lower final training loss compared to
868 other models.

879
880 Figure 9: Training curve of models for signal integrity analysis tasks.
881
882883 D MORE EXPERIMENTS
884885 D.1 MORE EXPERIMENTS WITH DIFFERENT STRUCTURAL ENCODING
886

887 In this section, we conduct experiments with random walk structural encoding (RWSE) and
888 Laplacian-based positional encoding (LapPE) for GraphGPS and our Si-GT. Specifically, we replace
889 the mesh pattern encoding in Si-GT with RWSE and LapPE variants, and compare the performance
890 against GraphGPS. In the experiments, we vary the random walk length with 4, 8, 16 for RWSE-
891 based position encoding of GraphGPS model (e.g., RWSE16) and set 8 top eigenvectors of the
892 graph Laplacian for LapPE (e.g., LapPE8). Additionally, we compare Si-GT with GraphGPS using
893 composite position encoding (e.g., LapPE8+RWSE16), following the implementation in (Rampášek
894 et al., 2022), we concatenate the LapPE and RWSE vectors to form the final positional encoding.
895 Experimental results are reported in Table 8.

896 Table 8: Mean relative accuracy (%) of crosstalk delay and glitch prediction results.
897

Model	PE/SE	Segment Delay		Segment Glitch	
		\hat{D}_{vic}	\hat{D}_{agg}	t_{width}	v_{max}
GraphGPS	LapPE8	87.76	71.69	96.15	97.30
GraphGPS	RWSE4	88.12	72.19	95.67	97.21
GraphGPS	RWSE8	88.24	71.97	95.14	97.69
GraphGPS	RWSE16	88.23	72.65	96.61	97.99
GraphGPS	LapPE8+RWSE16	88.28	72.92	96.49	97.83
Si-GT	LapPE8	87.72	73.50	95.25	93.18
Si-GT	RWSE16	88.21	73.78	97.44	96.30
Si-GT	LapPE8+RWSE16	87.85	73.47	96.87	95.32
Si-GT	MPE	88.32	73.67	98.36	97.78

909 As shown in Table 8, Si-GT consistently achieves higher accuracy in delay and glitch width prediction
910 tasks compared to GraphGPS across various positional encoding configurations. These results
911 highlight the effectiveness of our mesh pattern encoding (MPE) and demonstrate the robustness of
912 Si-GT when combined with both LapPE and RWSE encodings.

913 D.2 INFERENCE EXAMPLES
914

916 We visualize the predicted values of key signal integrity metrics against the SPICE-measured ground
917 truth in Figure 10. For crosstalk glitch and crosstalk delay of victim prediction tasks, Si-GT con-
918 sistently provides the closest match to the ground truth across all metrics, demonstrating its ability

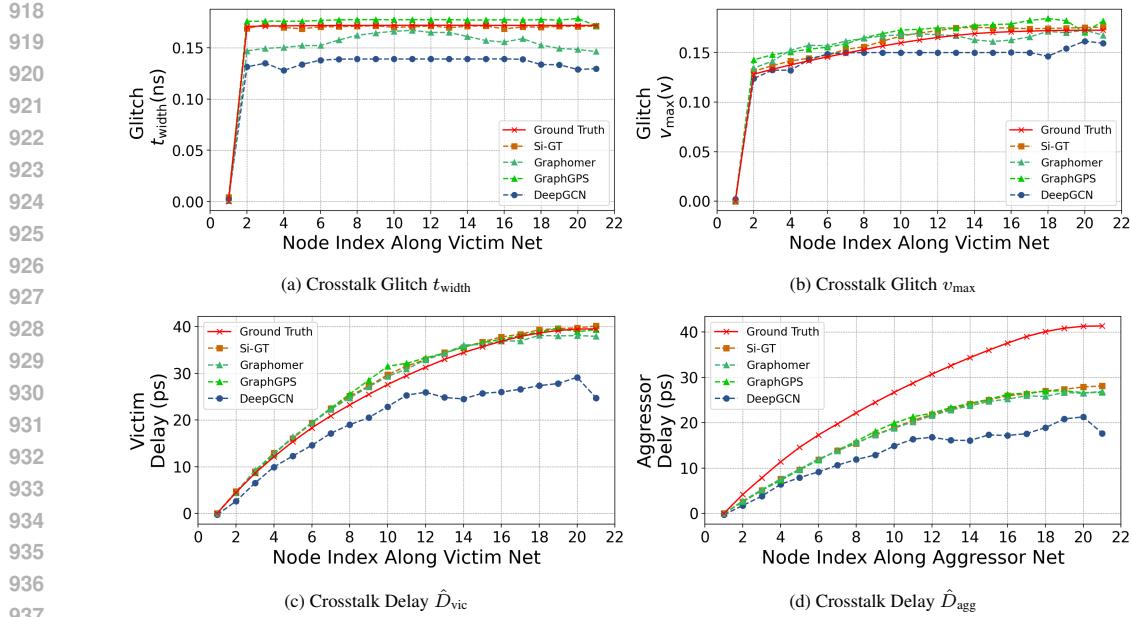


Figure 10: Comparison of crosstalk prediction accuracy with the number of wire segments.

to accurately model both local coupling effects and global signal dependencies. Additionally, we present an example of aggressor delay prediction. Since capacitive coupling primarily affects the victim net, signal integrity analysis mainly focuses on victim-side behavior.

D.3 FINE-GRAINED ABLATION EXPERIMENTS

Ablation experiment setup. Table 5 summarizes the ablation results for the core architectural components introduced in Si-GT. All ablations are performed using the strongest Si-GT backbone selected for each prediction task, e.g., GCN for delay-segment, GAT for delay-sink, GIN for glitch-segment, and GCN for glitch-sink, following the configurations reported in Table 2 and Table 3.

Additional analyses. To further understand the contribution of each component, we conduct more fine-grained ablation studies. In Equation 4, spatial encoding and edge encoding are incorporated as attention bias terms to better capture the global structural context of interconnect topology. In this section, we remove $\tilde{\Phi}_d$ and $\tilde{\Phi}_{sp}$ from the attention bias to isolate their effect on model performance.

Table 9: Ablation study on the removal of spatial and edge encoding biases.

Model	Delay Prediction				Glitch Prediction			
	Segment		Sink		Segment		Sink	
	\hat{D}_{vic}	\hat{D}_{agg}	\hat{D}_{vic}	\hat{D}_{agg}	t_{width}	v_{max}	t_{width}	v_{max}
Si-GT-without $\tilde{\Phi}_d$, $\tilde{\Phi}_{sp}$	87.25	72.84	87.03	71.50	97.74	97.13	97.59	97.57
Si-GT	88.32	73.67	87.39	71.82	98.36	97.78	98.53	98.63