Representation Learning for Digital Integrated Circuit Design Automation

Pratik Shrestha and Ioannis Savidis
Department of Electrical and Computer Engineering
Drexel University, Philadelphia, Pennsylvania 19104

***Abstract*—Representation learning has become an effective technique utilized by electronic design automation (EDA) algorithms. By addressing challenges related to the increasing complexity of circuits and the corresponding stringent power, performance, and area (PPA) requirements, representation learning facilitates the automatic extraction of meaningful features from complex data formats, including images, grids, and graphs. In this paper, the application of representation learning to digital IC design automation is explored with an analysis of prior work on foundational concepts and case studies on tasks that include timing prediction and routability analysis. Key techniques, including image-based methods, graph-based approaches, and hybrid multimodal solutions, are described that highlight the improvements provided in routing and timing prediction. The provided advancements demonstrate the potential of representation learning to enhance efficiency, accuracy, and scalability in current integrated circuit design flows.**

I. Introduction

The automated design flow of digital circuits, which is shown in Fig. 1., transforms hardware description language (HDL) representations into manufacturable integrated circuits while meeting power, performance, and area (PPA) specifications through design stages that include logical synthesis, floorplanning, placement, clock network synthesis, and routing. The stages rely on technology-specific constraints and generate circuit descriptors that include netlists, interconnect parasitics, and reports to validate the design against the target PPA objectives. However, increasing circuit complexity, technology scaling, and stringent PPA requirements challenge traditional electronic design automation (EDA) tools, often resulting in suboptimal designs and failed timing closure that delay fabrication and require manual modifications to the circuit. Machine learning (ML) is therefore currently being explored to improve efficiency across the design flow through predictive modeling, adaptive optimization, and automation.

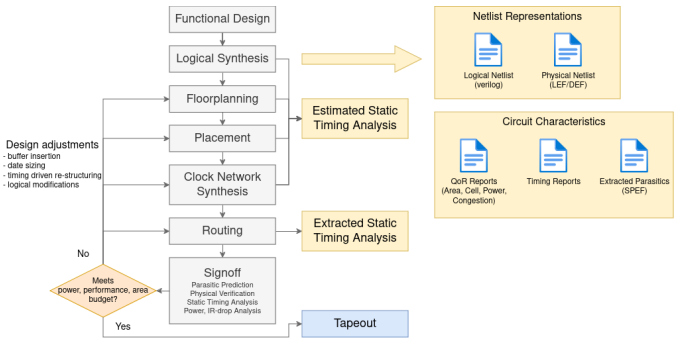

Fig. 1: Stages of the design automation flow from HDL to tapeout.

*For an extended version of this work, please refer to the arXiv preprint [1].

Artifacts, metrics, and data produced during the automated design flow are represented as images or graphs, which has resulted in an increased use of image- and graph-based learning techniques within EDA workflows. Representation learning [1] provides a promising direction for advancing EDA methodologies. By extracting meaningful features from raw data, deep representation learning effectively captures spatial, hierarchical, and relational patterns inherent in graph-based netlists and image-based layouts.

This paper provides an overview of representation learning algorithms and the application of such algorithms in EDA. Foundational concepts of representation learning and key methodologies relevant to EDA are introduced in Section II. Prior work that utilizes representation learning within the EDA flow and two case studies exploring timing analysis and routability prediction are described in Section III. Potential future directions for the use of representation learning to address challenges in digital IC design automation are presented in Section IV. Some concluding remarks are provided in Section V.

II. Background on Representation Learning

Representation learning, a subset of machine learning, targets the discovery of patterns or representations within complex data. High-dimensional raw data are transformed into low-dimensional vector embeddings that preserve essential features while minimizing irrelevant noise. The embeddings simplify the identification of complex relationships and support downstream tasks including classification, clustering, and regression. A *modality* refers to the type of data being processed, which include text, images, audio, video, and graphs, each requiring specialized algorithmic approaches for analysis and utilization.

a) Images: Image and grid-based data, represented as arrays of pixels or numerical grids, are well-suited for convolutional neural networks (CNNs) [2], which extract spatial patterns and hierarchical features. CNNs apply convolutional filters to produce feature maps that capture localized patterns such as edges, textures, and shapes. The operation of a convolutional layer is given by

$$\vec{y}_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \vec{x}_{i+m,j+n} \cdot \vec{w}_{m,n} + b, \tag{1}$$

where $\vec{x}$ is the input, $\vec{w}$ the filter kernel, b a bias term, and $\vec{y}$ the resulting feature map. Through successive layers, CNNs capture increasingly abstract features.

Residual networks (ResNets) [3] improve training with skip connections that mitigate vanishing gradients. Architectures including ResNet-18, ResNet-50, and ResNet-101 are widely used, with pre-trained models on ImageNet [4] enabling transfer learning while providing improved performance.

b) Graphs: Graphs naturally represent structured data, where entities are modelled as nodes and relationships as edges. Graph neural networks (GNNs) embed and analyze such data. Graph convolutional networks (GCNs) [5] update node embeddings by aggregating features of neighboring nodes as described by

$$\mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops and $\mathbf{W}^{(l)}$ is a learnable weight matrix.

Graph attention networks (GATs) [6] assign attention weights to neighbors, similar to transformers [7], as given by

$$\vec{h}_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\mathbf{W}^{(l)}\vec{h}_j^{(l)}\right), \tag{3}$$

where α_{ij} is the attention coefficient. GraphSAGE [8] further scales embeddings to large graphs by sampling a subset of neighbors, as given by

$$\vec{h}_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{AGG}\left(\{\vec{h}_j^{(l)} \mid j \in \mathcal{S}(i)\}\right)\right). \tag{4}$$

GCNs, GATs, and GraphSAGE differ in aggregation: GCNs treat all neighbors equally, GATs weigh neighbors adaptively at higher cost, and GraphSAGE samples neighbors for scalability. Together, the three types of neural networks highlight trade-offs between efficiency, scalability, and expressiveness in graph representation learning.

III. Use cases of Representation Learning in EDA

The practical applications of deep representation learning techniques in digital EDA are described in Section III. Prior work that utilizes ML for routability prediction and timing analysis, which are explored in detail in Sections III-A and III-B, respectively, are summarized in Table I. For each general use case, the table is categorized by the problem type (e.g., classification, regression, optimization, generative), data modality (e.g., images, graphs), representation techniques, and the neural network architectures employed. The body of work that falls under each of the described cases highlight targeted applications of representation learning techniques that address specific challenges in EDA workflows.

A. Case I: Routability Prediction

Routability is a measure of how effectively signal paths are routed between modules and devices within a circuit while adhering to design constraints. Congested areas within a circuit are susceptible to violating design rules and degrading performance due to high-density and high-activity hotspots. Predicting routing congestion is essential in order to identify high-risk zones within an integrated circuit (IC). Utilizing predicted routing congestion to optimize the placement of cells and modules by identifying overflow areas and improving routing efficiency ensures the circuit is both functional and manufacturable.

Image-based machine learning is generally used for such predictions, with RouteNet [9] being one of the earliest examples. RouteNet introduces a dual-model approach, utilizing a CNN (ResNet) to predict design rule violations (DRVs) and a U-network (U-net) to detect areas of a circuit considered as hotspots for design rule check (DRC) violations. Using image-like representations of layout data (e.g. pin density, heat maps, macro region inclusion, mapped RUDY [10] scores, etc.), RouteNet matches the accuracy of traditional global routing algorithms while being significantly faster and providing improved accuracy of detected DRC hotspots. The J-network (J-net) is introduced in [11], which is an extension of the U-net, to predict post-routing DRVs. J-nets utilize post-placement layout images as inputs and combine high-resolution pin configuration images with low-resolution tile-based feature maps. The feature maps include the utilized routing resources by IP block, congestion metrics that include local/global net features and RUDY scores, and measures of the density of logic gate pins, clock pins, logic cells, and filler cells. Generative adversarial networks (GANs) are implemented in [12] to predict routing congestion. Using images of placed circuits and cell and macro connectivity as input, the GAN generates heat maps that visually represent levels of routing congestion. The generator, also based on a U-Net model, creates heat maps while the discriminator ensures the generated heat maps are realistic by comparing the generated maps to actual data. The accuracy of the model is improved through adversarial learning. Another approach enhances U-Net skip connections with a spatial attention module that integrates average-pooling, max-pooling, and standard deviation operations to address any imbalance in the data [13]. In [14], a hyper-image based CNN is used to predict DRVs. The hyper-image is a multi-dimensional representation that combines placement and global routing features including pin counts, neighborhood pin density, and routing overflow (horizontal and vertical), across multiple metal layers. The multidimensional representation preserves spatial and layer-specific congestion characteristics, offering a more comprehensive input to the model than traditional 2-D congestion maps. DRVNet [15] detects DRV hotspots by utilizing early global routing (eGR)-derived features within a U-Net architecture to predict layer-wise instances of DRVs. Unlike traditional heuristic-based congestion models that rely on a single snapshot of a circuit design, DRVNet extracts layer-wise congestion features from multiple instances of eGR. Global cell (GCell) based congestion maps are used to capture the utilized routing resources, the pin density, and the via distribution across metal layers. Using the grid-structured predictions from DRVNet, BlkgComp [15], a classification-based blockage placement framework, employs ResNet-50 to identify congestion-prone regions of the circuit and to generate an initial set of placed routing blockages. An RL agent further optimizes the placement of the blockage by utilizing a policy function that modifies the location of a blockage based on a reward function that balances reductions in DRVs with routing feasibility.

Images are typically used for the prediction of routability, but there are also techniques that utilize graphs. In [16], heterogeneous graphs are used to encode layout objects and any corresponding relationships. The graph represents globally

routed cells/macros and nets as nodes, with edges capturing connections between the nodes. The technique utilizes a GCN to predict the routing demand and the congestion status (binary classification) of each GCell. Beyond utilizing images or graphs as a single modality in isolation, a hybrid model that combines a pin graph neural network (PGNN) and a U-Net is proposed in [17] to predict DRC hotspots. The approach integrates a pin proximity graph, which captures detailed spatial relationships and pin accessibility, with grid-based layout features including pin density, RUDY, routing capacity, and locations of macro blockage to contextually represent the circuit. The combined representation enables the model to accurately identify areas of the circuit considered hotspots for DRC violations, addressing challenges in advanced technology nodes where pin accessibility and routing congestion heavily affect manufacturability.

B. Case II: Timing Prediction

ML is increasingly used in timing analysis to improve the accuracy and efficiency of predicted values across various stages of the IC design flow. ML models predict timing metrics early in the design process, accounting for parasitics, process variations, and even aging effects if sufficient data is available.

GNNs excel at capturing dependencies and topological relationships, using message-passing methods to model cumulative delays and interdependencies, which provides a powerful tool for timing-based ML in IC design. In [18], timing paths defined in static timing analysis (STA) reports are represented as graphs and applied to a GCN to predict the arrival time of a signal to the gates of the corresponding timing path. The post-routed arrival time is predicted in early design stages (floorplan, placement, CTS). The nodes of a timing path graph are embedded with features that include the delay of the gate, the arrival time provided at the current design stage, the fan-out, the logic level, the estimated interconnect length, and the interconnect capacitance. A GNN-based method is introduced in [19] to predict the pre-routing arrival time and slack at timing endpoints. The circuit is modeled as a heterogeneous graph, where pins are represented as nodes with features that include capacitance, distance to die boundaries, and timing endpoint indicators, and nets are represented as edges with attributes that include the length and delay of the interconnects and the slew of the signals. The GNN utilizes a level-by-level message-passing technique to emulate the propagation of timing paths in static timing analysis. GNNTrans, a GraphSAGE-based network adapted for wire delay and slew estimation, is introduced in [20] to model interconnect RC networks as graphs. Unlike previous methods that target timing paths or heterogeneous circuit elements, GNNTrans embeds RC graphs by aggregating local node features (e.g., resistance, capacitance) and global path features (e.g., Elmore delay, input slew, drive strength and functionality of load and drive cells). Attention mechanisms assign weights to neighboring nodes and edges based on relative importance, allowing the model to prioritize critical nets and timing paths. In [21], the circuit netlist is modelled as a heterogeneous graph with nodes (cells, pins) and edges (cell arc, net arc) that capture intra-cell and inter-cell timing relationships. A heterogeneous graph attention network (HGAT) [26] processes the graph, using attention mechanisms that prioritize critical interactions (e.g., pin-to-cell effects) and residual connections to prevent over-smoothing. The HGAT enables effective learning of local patterns and global dependencies, which ensures an accurate pre-routed prediction of circuit timing characteristics.

While graph-based methods excel at leveraging the topological nature of circuits for timing prediction, image-based methodologies offer a complementary set of techniques by treating timing analysis as a spatial problem. In [22], a ResUNet is utilized to process circuit features and performance parameters mapped to image-like representations, which include transition time and timing maps (e.g., WNS and TNS), to predict post-routing timing distributions and identify critical timing hotspots. By utilizing spatial circuit characteristics, the image-based technique offers an alternative method to analyze and optimize circuit timing characteristics.

The combining of the graph and image modalities for timing analysis is explored in [23] and [24]. GNNs and CNNs are combined in [23] to improve the predicted timing of pre-routed paths. GNNs model circuit dependencies to predict net resistance and capacitance, while CNNs process spatially-organized features including congestion maps to refine predictions of the arc length. The multi-modal network proposed in [23] is integrated into a commercial EDA tool in [24], where the predicted timing characteristics replace default estimates provided by the tool. The predicted timing values improved WNS and TNS and reduced the number of timing violations reported by the commercial tool. In [25], an optimization-aware framework integrates heterogeneous graph attention networks, CNNs, and transformers to improve pre-routing timing prediction and guide placement optimization with better WNS and TNS outcomes.

IV. Future Directions

This section outlines future directions for advancing representation learning in EDA. The areas of interest include multimodal learning, the natural language modality and large language models (LLMs), and workflow standardization, which are discussed in Section IV-A, Section IV-B, and Section IV-C, respectively.

A. Multimodal Learning and Alignment

Multimodal learning [27] is gaining traction in EDA as combining diverse data types such as graphs and images enables richer circuit representations. Recent work leveraging the complementary strengths of both graph and image modalities improves prediction accuracy and optimization efficiency [16], [23], [24]. Future research must include systematic ablation studies [28] to quantify the contribution of each modality.

Another important application is modality alignment, where one modality guides or enhances a different modality [29], [30]. For example, images provide spatial context for graph analysis, or graphs inform image-based reasoning. Alignment improves feature extraction, anomaly detection, and design optimization by bridging gaps and reducing ambiguities. The commonality between graph and image/grid features is listed in Table I. As an example, the RUDY metric is used as both a grid-based feature [13] and a graph-based feature [17]. Such

TABLE I: Prior work utilizing representation learning in digital IC design automation.

Category	Ref	Objective	Problem Type	Modality	Representation/Feature Description	Network Architecture
Routability Prediction	[9]	Predict number of DRV and DRC hotspots during placement	Regression & Classification	Image	Pin density, macro regions, RUDY scores	ResNet for DRV and U-net for DRC hotspots
	[11]	Predict if a tile in the layout is a Design Rule Check (DRC) hotspot	Classification	Image	Pin configuration images and tile features (routing resources, congestion metrics, and density metrics)	Customized U-Net (J-Net)
	[12]	Predict routing congestion based on heatmaps generated from placement	Generative	Image	Post-placement layout images combined with connectivity images (maps netlist connections as edges based on placement locations)	GAN
	[13]	Predict routing congestion heatmaps during placement	Regression & Classification	Image	Grids of numerical features per tile: Macro Regions, RUDY, Pin RUDY, Cell Density, and Congestion Map.	U-Net enhanced with spatial attention
	[14]	Predict if a tile in the layout is DRV prone	Classification	Image	Hyper-images combining placement (pin counts, neighborhood pin density) and global routing (routing overflow by layer)	CNN
	[15]	Predict layer-wise DRV hotspots to optimize routing blockages	Classification & Optimization	Image	GCell-based congestion maps, multiple early global routing (eGR) runs, cell/macro density, pin density, routing track utilization, via counts	U-Net for DRV prediction, ResNet-50 for blockage ranking, Actor-Critic for blockage optimization
	[16]	Predict whether the routing demand exceeds the routing capacity of the GCell boundary	Classification	Graph	Heterogeneous Graph consisting of - global cells/nodes with routing capacity, net/pin overlap density, and macro overlap as features - net nodes with bounding box dimensions, net overlap density, and pin overlap count as features	Heterogenous GNN
	[17]	Predict DRC hotspots considering pin accessibility and routing congestion	Classification	Graph & Image	Pin proximity graph consisting of - pins as nodes with pin location, orientation, cell type of pins (standard cell, macro, etc.), and proximity features (distance to nearest pins, etc.) - proximity-based edges connecting pins that are within a defined distance - edges linking pins with shared access points or overlapping regions Grid-based features (e.g., Pin Density, RUDY, Routing Capacity, etc.)	Combined GNN (Pin Proximity Graph) and U-Net (Grid Features)
Timing Prediction	[18]	Predict post routing gate arrival time at post floorplan, placement, and CTS design stages	Regression	Graph	Timing path graphs with gates, inputs, and outputs as nodes with attributes including structural features (logic level, fan-out count, standard cell encoding, etc.), timing based features (stage delay, arrival time, etc.) and parasitics.	GNN
	[19]	Predict pre-routing arrival time and slack at timing endpoints	Regression	Graph	Heterogeneous graph with - pins as nodes (features: pin capacitance, position, type) - net/cellas edges (features: distances, delay, slew)	GNN
	[20]	Predict post routing wire slew and delay	Regression	Graph	*RC* network graph consisting of - interconnect capacitance as nodes - interconnect resistance as edges	GraphSAGE and tranformers
	[21]	Predict pre-routing endpoint arrival time considering routing-induced parasitics and timing optimizations	Regression	Graph	Heterogeneous graph - cells as nodes with standard cell encoding, drive strength, slew, and slack as features - pins as nodes with pin type, capacitance, and arrival time as feature - cell arc as edges with cell delay and slew as features - net arc as edges with net length and delay as features	GAT
	[22]	Predict instance-level timing hotspots and slack distributions at placement	Regression	Image	Mesh grid-based maps of transition time, instance density, and timing slack (worst slack, total slack, negative slack)	ResNet and U-Net
	[23] [24]	Predict net resistance, capacitance, arc delay, and slew at the pre-route stage for timing optimization	Regression	Graph & Image	Timing graph with - pins or cells as nodes with electrical properties (e.g., capacitance, resistance), timing slack, arc delay, and slew as features - timing arcs between nodes as edges Image based congestion maps (e.g., RUDY, density, pin capacitance location)	GNN and CNN (multimodal)
	[25]	Predict pre-routing endpoint arrival	Regression	Graph & Image	Heterogeneous netlist graph (cells, pins, arcs, nets) with timing attributes Grid-based layout images (congestion, density, placement features)	HGAT + U-Net (local) + Transformer (global)

commonality in features between modalities provides opportunities for alignment, which enables a unified representation that reduces redundancy and improves analysis.

B. Natural Language Modality and Large Language Models

In EDA, text-based data primarily comes from register transfer level (RTL) descriptions, which form the foundation of a circuit representation. Other sources, including textbooks, research papers, and technical documentation, also provide valuable insights. Yet, text-based data remains underutilized in EDA, representing untapped potential for automation, optimization, and generation.

The rise of LLMs has opened new ways to leverage text data in circuit design. LLMs are able to interpret, generate, and analyze text for tasks that include translating natural language requirements into formal specifications, extracting insights from documentation, and automating reporting [31]. Current applications include generating HDL code [32], [33], interpreting code and analyzing logs for debugging and verification [34] , and powering chatbots that support design queries and tool usage [35] . However, integration of LLMs with other modalities such as graphs and images is still limited. Combining language understanding with spatial and relational insights enables a more comprehensive and adaptive EDA flow.

C. Standardizing EDA through Open Data and Frameworks

Multimodal learning in EDA relies on unified data formats, workflows, and evaluation protocols to ensure comparability, reproducibility, and reuse. While features such as netlist graphs, layout images, and timing data are widely used, disparities in datasets, PDKs, and preprocessing hinder benchmarking and model transfer. Restrictions on publishing data generated with commercial tools further limit sharing. Standardized frameworks and representations address such challenges by enabling seamless integration of modalities, fostering alignment, and supporting scalable EDA pipelines.

Several initiatives are advancing ML for EDA research and development through standardized data formats and open datasets for benchmarking [36]–[38] , ML infrastructure [39], [40], and open-source tools and resources [41]. Such efforts provide a foundation for standardized, multimodal EDA workflows.

V. Conclusion

This paper explores the current landscape and potential of representation learning in EDA. The paper highlights the ability of representation learning to leverage diverse modalities including graphs, images, and hybrid approaches to address challenges in modern circuit design. By categorizing key applications listed in Table I and analyzing general use cases in routability prediction and timing analysis, the utilization of representation learning to improve prediction accuracy, efficiency, and scalability is demonstrated. Future directions on expanding multimodal learning, enhancing interoperability, and exploring new methods for aligning diverse data modalities are discussed, which enable scalable and robust solutions to meet the growing complexity of circuit design workflows.

REFERENCES

[1] Y. Bengio, A. Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1798–1828, Jun. 2013.

[2] O.S. Keiron and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, pp. 1–11, Nov. 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Jun. 2016.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, Vol. 115, pp. 211–252, Apr. 2015.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, pp. 1–14, Sept. 2016.

[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, pp. 1–12, Oct. 2017.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, NIPS'17, pp. 6000–6010, Dec. 2017.

[8] W. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1025–1035, Dec. 2017.

[9] Z. Xie, Y.-H. Huang, G.-Q. Fang, H. Ren, S.-Y. Fang, Y. Chen, and J. Hu, "RouteNet: Routability prediction for mixed-size designs using convolutional neural network," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 1–8, Nov. 2018.

[10] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, Apr. 2007.

[11] R. Liang, H. Xiang, D. Pandey, L. Reddy, S. Ramji, G.-J. Nam, and J. Hu, "DRC hotspot prediction at sub-10 nm process nodes using customized convolutional network," *Proceedings of the IEEE/ACM International Symposium on Physical Design*, pp. 135–142, Mar. 2020.

[12] C. Yu and Z. Zhang, "Painting on placement: Forecasting routing congestion using conditional generative adversarial nets," *Proceedings of the Annual Design Automation Conference (DAC)*, pp. 1–6, Jun. 2019.

[13] Y. Chen, D. Zhang, and H. Chen, "A standard deviation incorporated attention module enhanced U-Net for routability prediction," *Proceedings of the International Conference on Electronic Technology and Information Science (ICETIS)*, pp. 671–674, May 2024.

[14] S. F. Almeida, R. Netto, T. A. Fontana, E. Aghaeekiasaraee, U. Gandhi, A. F. Tabrizi, J. L. Güntzel, L. Behjat, and C. Meinhardt, "Eh-DRVP: Combining placement and global routing data in a hyper-image-based DRV predictor," *Integration*, Vol. 101, pp. 1–14, Mar. 2024.

[15] A. B. Kahng, S. Kundu, and D. Yoon, "Placement tomography-based routing blockage generation for DRV hotspot mitigation," *Proceedings of the IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, Oct. 2024.

[16] Z. Qi, Q. Peng, S. Hu, and H. You, "Effective heterogeneous graph neural network for routing congestion prediction," *Proceedings of the International Symposium of Electronics Design Automation (ISEDA)*, pp. 369–373, May 2024.

[17] H. Park, K. Baek, S. Kim, K. Choi, and T. Kim, "Pin accessibility and routing congestion aware DRC hotspot prediction for designs in advanced technology nodes with consolidated practical applicability and sustainability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 43, No. 12, pp. 4786–4799, May 2024.

[18] P. Shrestha, S. Phatharodom, and I. Savidis, "Graph representation learning for gate arrival time prediction," *Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD*, pp. 127–133, Sept. 2022.

[19] Z. Guo, M. Liu, J. Gu, S. Zhang, D. Z. Pan, and Y. Lin, "A timing engine inspired graph neural network model for pre-routing slack prediction," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 1207–1212, Aug. 2022.

[20] Y. Ye, T. Chen, Y. Gao, H. Yan, B. Yu, and L. Shi, "Fast and accurate wire timing estimation based on graph learning," *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, Apr. 2023.

[21] G. He, W. Ding, Y. Ye, X. Cheng, Q. Song, and P. Cao, "An optimization-aware pre-routing timing prediction framework based on heterogeneous graph learning," *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 177–182, Jan. 2024.

[22] H. Liu, S. Tao, Z. Huang, B. Xie, X. Li, and G. Li, "Instance-level timing learning and prediction at placement using Res-UNet network," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, May 2024.

[23] K. Chang and T. Kim, "Pre-route timing prediction and optimization with graph neural network models," *Integration*, Vol. 99, pp. 1–13, Nov. 2024.

[24] J. Ahn, K. Chang, K.-M. Choi, T. Kim, and H. Park, "DTOC-P: deep-learning-driven timing optimization using commercial EDA tool with practicality enhancement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 43, No. 8, pp. 2493–2506, Feb. 2024.

[25] P. Cao, Y. Qin, G. He, W. Ding, X. Cheng, Z. Zhang, and Y. Ye, "An optimization-aware prerouting timing prediction framework based on multimodal learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 44, No. 10, pp. 3896–3909, Oct. 2025.

[26] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," *The World Wide Web Conference (WWW)*, pp. 2022–2032, May 2019.

[27] M. A. Manzoor, S. Albarri, Z. Xian, Z. Meng, P. Nakov, and S. Liang, "Multimodality representation learning: A survey on evolution, pretraining and its applications," *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 20, No. 3, pp. 1–34, Oct. 2023.

[28] C. A. Ellis, R. Zhang, V. D. Calhoun, D. A. Carbajal, M. S. Sendi, M. D. Wang, and R. L. Miller, "A novel local ablation approach for explaining multimodal classifiers," *Proceedings of the International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 1–6, Oct. 2021.

[29] L. H. Li, M. Yatskar, D. Yin, C. J. Hsieh, and K. W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, pp. 1–14, Aug. 2019.

[30] J. Li, T. Tang, W. X. Zhao, Z. Wei, N. Jing Yuan, and J. R. Wen, "Few-shot knowledge graph-to-text generation with pretrained language models," *arXiv preprint arXiv:2106.01623*, pp. 1–11, Jun. 2021.

[31] R. Zhong, X. Du, S. Kai, Z. Tang, S. Xu, H.-L. Zhen, J. Hao, Q. Xu, M. Yuan, and J. Yan, "Llm4eda: Emerging progress in large language models for electronic design automation," *arXiv preprint arXiv:2401.12224*, pp. 1–15, Dec. 2023.

[32] M. Liu, N. Pinckney, B. Khailany, and H. Ren, "Verilogeval: Evaluating large language models for verilog code generation," *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–8, Oct. 2023.

[33] S. Thakur, B. Ahmad, H. Pearce, B. Tan, B. Dolan-Gavitt, R. Karri, and S. Garg, "Verigen: A large language model for verilog code generation," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 29, No. 3, pp. 1–31, Apr. 2024.

[34] J. Blocklove, S. Garg, R. Karri, and H. Pearce, "Evaluating LLMs for hardware design and test," *Proceedings of the IEEE LLM Aided Design Workshop (LAD)*, pp. 1–6, Jun. 2024.

[35] U. Sharma, B. Y. Wu, S. R. D. Kankipati, V. A. Chhabria, and A. Rovinski, "Openroad-assistant: An open-source large language model for physical design tasks," *Proceedings of the ACM/IEEE International Symposium on Machine Learning for CAD*, pp. 1–7, Sept. 2024.

[36] P. Shrestha, A. Aversa, S. Phatharodom, and I. Savidis, "EDA-schema: A graph datamodel schema and open dataset for digital design automation," *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pp. 1–8, Jun. 2024.

[37] J. Jung, A. B. Kahng, S. Kim, and R. Varadarajan, "METRICS2.1 and flow tuning in the IEEE CEDA robust design flow and OpenROAD," *Proceedings of the IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pp. 1–9, Nov. 2021.

[38] Z. Chai, Y. Zhao, W. Liu, Y. Lin, R. Wang, and R. Huang, "Circuitnet: An open-source dataset for machine learning in VLSI cad applications with improved domain-specific evaluation metric and learning strategies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 42, No. 12, pp. 5034–5047, Jun. 2023.

[39] R. Liang, A. Agnesina, G. Pradipta, V. A. Chhabria, and H. Ren, "CircuitOps: An ML infrastructure enabling generative AI for VLSI circuit optimization," *Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, pp. 1–6, Oct. 2023.

[40] P. Shrestha and I. Savidis, "EDA-ML: Graph representation learning framework for digital IC design automation," *Proceedings of the International Symposium on Quality Electronic Design (ISQED)*, pp. 241–246, Apr. 2024.

[41] T. Ajayi and D. Blaauw, "OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain," *Proceedings of the Government Microcircuit Applications and Critical Technology Conference (GOMACTech)*, pp. 1–6, Mar. 2019.