Machine Learning in Nanometer AMS Design-for-Reliability
(Invited Paper)

Tinghuan Chen
Chinese University of Hong Kong
thchen@cse.cuhk.edu.hk

Qi Sun
Chinese University of Hong Kong
qsun@cse.cuhk.edu.hk

Bei Yu
Chinese University of Hong Kong
byu@cse.cuhk.edu.hk

***Abstract*—With continued scaling, the susceptibility of nanometer CMOS to reliability issues has increased significantly in analog/mixed-signal (AMS) circuits. The industrial large-scale AMS circuits bring grand challenges in the efficiency of reliability design and verification. Machine learning (ML) provides one promising direction to achieve significant speedup in design closure. In this paper, we introduce typical reliability issues and review some excellent arts in applying ML approaches to AMS circuits reliability verification and design-for-reliability (DFR). We also discuss some open challenges in the industry and provide potential ML-based solutions. We hope this paper can promote the development of AMS circuits DFR.**

I. Introduction

With continued scaling, the reliability issues cause an increasing failure of nanometer-scale analog/mixed-signal (AMS) circuits [1]. Compared with digital circuits, AMS circuits are more easily influenced or harmed by reliability issues since they are more susceptible to noises and variations. It is difficult to achieve perfect reconstruction even if many advanced calibrations were developed [2].

The typical reliability issues are categorized as spatial and temporal reliability issues [3]. The spatial reliability mainly relies on the circuit layout and the impact on the structure and geometry of the circuit. The issues occur immediately after fabrication [4], [5]. In practice, the process variation and the scale wavelength of the light source for lithography are two main factors to cause spatial reliability issues.

Different from the spatial reliability issues, temporal reliability issues are time-varying and would change for operating conditions, such as the temperature, operating voltage, switching activity, presence and activity of neighboring circuits. According to the duration of reliability issues, the temporal reliability issues can be further classified as aging effects and transient effects. The aging effects mainly contain hot carrier injection (HCI), bias temperature instability (BTI), time-dependent dielectric breakdown (TDDB), and electromigration (EM) [3]. These aging impacts result in a gradual circuit failure, while transient effects distort electrical signals transmitted in the circuit for a time slot. A high-quality signal is beneficial to transfer high-speed reliable data in the circuit. However, a signal waveform is easy to distort by electromagnetic interference (EMI) and unwanted noise in the transient unreliable circuits. Typically, signal integrity (SI) is adopted to measure the quality of a signal in a circuit changes under these transient effects.

To effectively save development costs and provide an opportunity for interactive feedback, many excellent computer-aided design (CAD) techniques are adopted to achieve the reliability of AMS circuits. As shown in Fig. 1, a typical AMS circuit CAD flow contains circuit topology design, device sizing, pre-layout simulation, device generation, placement, routing, parasitic extraction, post-layout simulation and reliability verification. If there are one or more violations in the reliability verification, as a typical verification-then-fix approach, the designer will go back to previous steps to fix them.

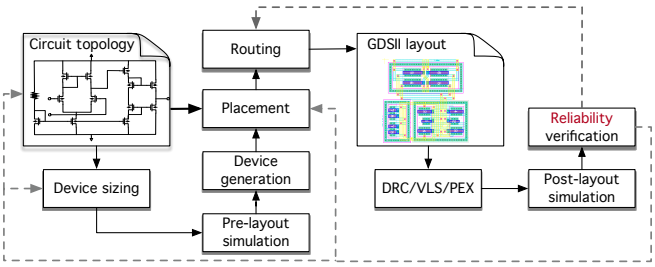

Fig. 1 AMS circuits design flow.

In the past few decades, there were many analytical models proposed to perform reliability simulation and verification in CAD flow [3]. Typically, these analytical models are built and verified with a rich of semi-conductor and technology knowledge accumulation and a mass of reliability experiments, respectively, before they are integrated into CAD flow. However, these analytical models are computationally expensive. For example, EM analysis requires solving partial differential equations [6]. Besides, AMS circuit design is still a heavily manual, time-consuming, and error-prone task due to its high design flexibility and sensitive impact on the circuit performance and reliability by even minor changes in the circuit implementation. Thus the traditional verification-then-fix approaches cause the low efficiency of design and verification with the ever-growing reliability violations at advanced technology nodes and the large-scale AMS designs.

Recently, machine learning (ML) techniques, including deep learning, have achieved impressive successes in various applications [7]. ML techniques are adopted for fast predictions and estimations after the knowledge is acquired from large amounts of data. Moreover, we see a clear trend of incorporating ML techniques into CAD flow to improve efficiency and accelerate the process of AMS circuits design and reliability verification [8].

This paper gives a comprehensive review of some excellent arts in applying ML approaches to AMS circuits design-for-reliability (DFR). The rest of this paper is organized as follows. In Section II, we review some recent works in applying ML approaches to AMS circuit reliability verifications. In Section III, advanced ML-based CAD methodologies with DFR are presented. In Section IV, we discuss open challenges and promising directions, followed by the conclusion in Section V. We hope this review paper can promote the development of AMS circuits DFR.

II. ML-based AMS circuits reliability verification

The reliability verifications need to be performed to effectively save development costs and provide the opportunity for interactive feedback before committing the AMS designs to silicon. However, the industrial large-scale AMS circuits bring grand challenges in the

efficiency of reliability verification since the traditional analytical models are computationally expensive.

To improve the efficiency of reliability verification, ML-based verification approaches were proposed. Most existing ML-based reliability verification approaches are in the supervised-learning manner, where the ML model is learned on several input-output pairs in the training dataset and guided by a task-related loss function. Typically, in the training dataset, the input is directly extracted from the designed AMS circuit while the output is from reliability simulators or experiments. The ML model with the supervised learning manner is trained on the training dataset by optimizing loss function or objective function with gradient-based optimization methods, such as stochastic gradient descent. Once the ML model is well trained offline, it can be used to online detect reliability violations all the time. Besides, the ML model can be deployed on some advanced hardware platforms, such as GPUs and FPGA, to accelerate inference [9]. Thus ML model can achieve significant speedup in the detection of reliability violations.

Typically, the spatial reliability violations are detected on the layout while the temporal reliability violations are verified on either layout or netlist. Traditional ML-based reliability verification models adopt very complicated feature engineering to extract features from layout, netlist, and technology files [10], [11]. Then the extracted features are input into a traditional ML model to perform detection of reliability violations. However, these complicated feature engines cannot be guided by task orientation to extract features. Recently, various deep learning models, such as convolutional neural networks (CNNs) and graph convolutional networks (GCNs), were proposed to automatically extract features to fit distinctive tasks so that they can achieve better performance in detection accuracy.

For the reliability verifications on layout, the circuit layout can be treated as an image. Intuitively, reliability violation detection is formulated as an image classification problem. To locate the violations in the circuit layout, the whole layout needs to be partitioned into several small tiles. Then the features of the small layout tile are automatically extracted by advanced CNNs models and structures, such as attention mechanism [4]. Moreover, in order to mark the concrete positions of reliability violations by a bounding box, the reliability verification is formulated as a classification and regression problem, which is handled by a clip proposal network [5]. Besides, generative adversarial networks (GANs) are customized to estimate the distribution or probability of reliability violations in each layout location by inputting a circuit layout [6], [12]. In particular, in order to integrate stress conditions into ML-based reliability models, powers, signal arrival time, toggle rate, and currents are used as condition features [12], [13].

For the reliability verifications on AMS circuit netlist, the CNNs-based ML model cannot be directly used since the netlist is irregular grid-based data and is not straightforward for the convolution and pooling in traditional CNNs. Recently, GCNs were presented to handle these irregular grid-based data [14]. In the netlist, devices and parasitics of interconnection can be treated as nodes then the netlist can be naturally represented as a graph [15]–[21]. GCNs can be adopted to detect reliability violations on netlist by classification or regression techniques. However, the typical AMS circuit netlists have heterogeneity since they contain multi-typed basic devices and multi-typed connection pins. An inevitable problem of the homogeneous representation method is that it fails to characterize the diversities among pins, devices, connections, and relative sequential relationships. A heterogeneous GCN is proposed to characterize the heterogeneity of the AMS circuit netlist [16]. In order to achieve good scalability for the large-scale circuit netlists, the dynamic graph partitioning scheme is used among different epochs in the training stage [16]. Besides, the detections of some reliability issues such as TDDB can be transformed as the identifications of risk subcircuits since they heavily rely on the topologies of subcircuits and device parameters. Therefore, some subcircuit identification frameworks [18], [19], [22] are customized to identify subcircuits with reliability violations.

Although the ML-based reliability verification models are proposed to achieve fast and accurate reliability verifications, the traditional design-verification flows are still limited by low efficiencies. The essential reason is that the traditional verification-then-fix approaches are ill-equipped when faced with the ever-growing reliability violations at advanced technology nodes. In the next section, we will present some advanced ML-based CAD methodologies for DFR, which integrate reliability verification into multi-stage design flows.

III. ML-based AMS circuit CAD flow for DFR

A typical AMS circuit CAD flow is shown in Fig. 1. In the past few decades, many excellent AMS automatic design frameworks were developed to facilitate AMS circuit design [23]–[26]. In order to make designed circuits satisfy various design and reliability specifications, some analytical models need to be built to guide designs in multi-stage. Considering the distributions of current under different via layouts, a model about EM reliability of redundant via structures is built. Based on the EM model, redundant via layouts are chosen by using integer linear programming, to increase the EM-related lifetime [27]. The routing algorithm is enhanced by considering bending loss and temperature variations to alleviate the refractive index of silicon [28]. However, these analytical models are extremely complex so as to hinder the design flows from achieving high-quality and free-reliability-issue designs.

Recently, with the development of ML technologies, ML can fit complex reliability models via learning from the historical dataset. Besides, some ML-based AMS automatic design frameworks, including device sizing, constraint generation, device generation, placement and routing, were developed to greatly reduce the time and labor costs. Intuitively, the reliability, which can be modeled by various ML techniques, is regarded as a design specification and added into the optimization objective in these ML-based AMS design frameworks. Embedding the ML-based reliability models into stages of the AMS circuits CAD flow facilitates DFR.

In the AMS automatic design framework, the device sizing is used to tune each device's size to satisfy circuit design specifications. Typically, the device sizing is formulated as a design space exploration (DSE) problem, which can be well handled by two ML-based methodologies: Bayesian optimization [29]–[31] and reinforcement learning [20], [32]. Considering that the device sizing problem has multiple design specifications as objectives, *e.g.*, bandwidth, gain and reliability for a typical amplifier, a batch Bayesian optimization via multi-objective acquisition ensemble was proposed to tune device sizing [29]. In order to improve the generalization of the Gaussian process in Bayesian optimization, the neural networks are used to replace traditional kernel functions, such as the radial basis function [30]. Since that there are hierarchical structures in AMS circuits, a Bayesian optimization-assisted hierarchical analog layout synthesis was developed to optimize the circuit performance [31]. Representing the netlist as a graph, a reinforcement learning technique combined with GCNs was proposed to automatically tune the size of each device [20] so that the pre-layout design can satisfy

the design specifications. In these device sizing methods, search-based and gradient-based methodologies are leveraged to handle the optimization problem.

After the circuit topologies and device sizes are well determined, the physical design will be performed to generate the layout. AMS circuits are more susceptible to the quality of the signal. To alleviate the impact of noises and manufacturing defects, and enhance circuit robustness, AMS circuits use topologies together with device matchings as the constraints in physical design. The annotations of physical design constraints can be formulated as a node classification problem on a graph. Straightforwardly, GCNs were customized to automatically annotate netlist [18], [19]. Besides, a graph matching algorithm is performed to identify primitives [18], which is formulated as a subgraph isomorphism problem. Moreover, considering that the AMS circuit has a hierarchical tree structure, an unsupervised GCN-learning-based method was proposed to identify system-level and device-level constraints [22].

Placement and routing are two key steps for circuit performance and reliability in the physical design stage of the AMS circuit. To optimize design objectives in the placement stage, *e.g.*, reliability, an ML-based model is built to treat the performance estimation of layouts as an image regression problem, as illustrated in Section II. A 3-D CNN was used to effectively capture the relative location information between the different placement subcircuits so that design specifications can be accurately estimated [33]. Furthermore, to directly estimate the performance according to the topology of the circuit, the netlist with the location of devices is formulated as a graph regression problem and GCNs were then utilized for prediction [21]. Then the traditional search algorithm such as the simulated annealing is adopted for the performance-driven AMS circuit placement. For AMS circuit routing, a generative neural network was proposed to guide routing by mimicking the sophisticated manual layout approaches [34]. Then routing guidance is honored via penalties in the cost function and the A* search algorithm is used for detailed routing.

IV. Open Challenges and Promising Directions

There are still some open challenges although ML has achieved great successes in nanometer AMS circuits DFR. In this section, we revisit the aforementioned studies, discuss these open challenges, and provide some promising directions.

How to use the ML-based reliability model to verify dynamic reliability. In the reliability verification stage, temporal reliability issues heavily rely on the dynamic stress conditions. Intuitively, the dynamic stress conditions, such as signal arrival time, toggle rate and currents, as (conditional) features, are used to input into ML-based model [12], [13]. However, traditional convolution and graph convolution fail to aggregate these dynamic stress conditions constrained by the large sizes of the industrial AMS circuit layout and netlist. Besides, if partition the circuit, the dynamic stress conditions would not be taken for granted to be inherited by the small-size tiles and subgraphs. The straightforward method is that the transient simulation is performed to obtain electric characteristics at each net in layout or netlist. Then the small size tile and subgraph with these electric characteristics input into the ML-based model to perform reliability estimation and prediction. But it is very time-consuming to perform the transient simulation for the large-scale AMS circuits since it needs a large number of accepted transient steps.

There are two possible solutions to this challenge. One solution is to detect the worst case among all possible dynamic stress conditions, to judge the circuit reliability. However, it is possible that the worst case is underestimated in the training dataset. The prominent issue in training ML models on the underestimated worst case is that the typical ML models often suffer from overfitting to these contaminated data resulting in the degradation of the estimated performance which leads to estimation performance degradation. Thus some works on learning from label noisy labels may be customized to handle this issue. Another solution is that some sub-circuits are incrementally, heuristically and hierarchically simulated to obtain dynamic stress conditions of some nets in the large-scale circuit. However, it is hard to make a better trade-off between simulation time and the accuracy of reliability estimation.

How to integrate the ML-based reliability model into the device sizing stage. As mentioned in Section III, the device sizing can be formulated as a DSE problem, where complex design objectives are modeled by Gaussian process (GP) regression [35] in Bayesian optimization or CNN and GCN models in reinforcement learning [20], [32]. A straightforward method is that the reliability performance is modeled by the ML method and then the reliability model, as a regularization term, is added to the optimization objective function. However, the huge search space brings low-quality Pareto-set solutions and several computationally expensive simulation steps to evaluate designs.

A more advanced method is to consider the reliability specifications as constraints in the ML model, instead of the regularization term of the objective function in Bayesian optimization [36]. By using this method, the sequence of inputs for evaluation can be efficiently selected to uncover high-quality Pareto-set solutions while satisfying constraints. However, GP has limited fitting ability for the very complicated model while CNNs and GCNs cannot be directly integrated into the Bayesian optimization framework since they cannot provide estimation uncertainty. One potential solution to this dilemma is using neural processes [37] and graph neural processes [38], which have the better fitting ability and provide estimation uncertainty.

How to integrate the ML-based reliability models into the placement stage. In the performance-driven placement, the performance of placement layout is modeled by CNNs [33] and GCNs [20]. Thus, the reliability performance can also be modeled in the same manner then it is used to guide reliability-aware placement.

In the optimization stage, the traditional evolutionary algorithm such as simulated annealing is computationally expensive since the ML-based reliability model inference has to be performed at each search step. Recently, the ML-based performance model was integrated into the placement objective then the placement problem is cast to training a neural network, which can be handled by a deep learning toolkit, such as Pytorch and TensorFlow, to achieve significant speedup on GPUs [39]. Thus in the future, the ML-based reliability-driven placement can be conducted in the same manner.

How to integrate the ML-based reliability models into the routing stage. There are few previous works to guide routing via ML-based performance and reliability model although there are some arts to guide routing via the ML-based routing preference model, such as [34]. The critical challenges are that it is difficult to model the performance and reliability via routing layout and integrate ML-based models into routing cost function. In practice, reliability effects mainly rely on the local layout, instead of the global layout. For example, EMI noise can be alleviated by routing power wiring and signal wiring in separate locations. Thus it brings an opportunity to bring the ML-based reliability model into the routing stage. Intuitively, ML-based routing reliability model cost can be added

into the typical routing cost function to guide routing. However, as mentioned above, in each search step, ML-based model inference has to be performed, which brings expensive computation.

A promising solution is that the gradient values of the ML-based reliability model can be used as cost values in the routing stage. The gradient values are updated by the back-propagation for several routing nets to achieve a better trade-off between runtime and layout routing reliability quality. Another critical point is that the layer-varying layout brings channel-varying input into the ML-based reliability model since the traditional layer-by-layer modeling cannot consider the impact on reliability issues (*e.g.*, EMI and mean time between failure caused by thermal distribution) of the layer itself from neighboring layers. One potential solution is using the Long Short-Term Memory model to handle the channel-varying input. Another potential solution is that GCNs, instead of CNNs, are used to model the reliability performance for grid-based routing algorithm.

V. Conclusion

It is promising to apply ML techniques to achieve AMS circuits DFR with high efficiency. In this way, the ML-based reliability model can learn from previous simulations and experiences and achieve AMS circuits DFR at hand more efficiently. So far ML techniques have found their applications in many ASM circuits DFR stages. In this paper, we have provided a comprehensive review of the literature, discuss some open challenges and promising solutions about ML in nanometer AMS circuits DFR. Although remarkable progress has been made in the area, we are looking forward to more studies to promote the development of AMS circuits DFR.

References

[1] B. Yu, X. Xu, S. Roy, Y. Lin, J. Ou, and D. Z. Pan, "Design for manufacturability and reliability in extreme-scaling VLSI," *Science China Information Sciences*, vol. 59, no. 6, pp. 1–23, 2016.

[2] T. Chen, B. Lin, H. Geng, and B. Yu, "Sensor drift calibration via spatial correlation model in smart building," in *Proc. DAC*, 2019.

[3] E. Maricau and G. Gielen, *Analog IC reliability in nanometer CMOS*. Springer Science & Business Media, 2013.

[4] H. Geng, H. Yang, L. Zhang, J. Miao, F. Yang, X. Zeng, and B. Yu, "Hotspot detection via attention-based deep layout metric learning," in *Proc. ICCAD*, 2020.

[5] R. Chen, W. Zhong, H. Yang, H. Geng, X. Zeng, and B. Yu, "Faster region-based hotspot detection," in *Proc. DAC*, 2019.

[6] W. Jin, S. Sadiqbatcha, Z. Sun, H. Zhou, and S. X.-D. Tan, "EM-GAN: Data-driven fast stress analysis for multi-segment interconnects," in *Proc. ICCD*, 2020.

[7] T. Chen, B. Duan, Q. Sun, M. Zhang, G. Li, H. Geng, Q. Zhang, and B. Yu, "An efficient sharing grouped convolution via Bayesian learning," *IEEE TNNLS*, 2021.

[8] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong *et al.*, "Machine learning for electronic design automation: A survey," *ACM TODAES*, vol. 26, no. 5, pp. 1–46, 2021.

[9] Q. Sun, T. Chen, J. Miao, and B. Yu, "Power-driven DNN dataflow optimization on FPGA," in *Proc. ICCAD*, 2019.

[10] C.-T. Ho and A. B. Kahng, "IncPIRD: Fast learning-based prediction of incremental IR drop," in *Proc. ICCAD*, 2019.

[11] W. Ye, M. B. Alawieh, Y. Lin, and D. Z. Pan, "Tackling signal electromigration with learning-based detection and multistage mitigation," in *Proc. ASPDAC*, 2019.

[12] H. Zhou, W. Jin, and S. X.-D. Tan, "Gridnet: fast data-driven EM-induced IR drop prediction and localized fixing for on-chip power grid networks," in *Proc. ICCAD*, 2020.

[13] Z. Xie, H. Ren, B. Khailany, Y. Sheng, S. Santosh, J. Hu, and Y. Chen, "Powernet: Transferable dynamic IR drop estimation via maximum convolutional neural network," in *Proc. ASPDAC*, 2020.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2016.

[15] Y. Ma, H. Ren, B. Khailany, H. Sikka, L. Luo, K. Natarajan, and B. Yu, "High performance graph convolutional networks with applications in testability analysis," in *Proc. DAC*, 2019, pp. 18:1–18:6.

[16] T. Chen, Q. Sun, C. Zhan, C. Liu, H. Yu, and B. Yu, "Analog IC aging-induced degradation estimation via heterogeneous graph convolutional networks," in *Proc. ASPDAC*, 2021.

[17] Y. Ma, Z. He, W. Li, L. Zhang, and B. Yu, "Understanding graphs in EDA: From shallow to deep learning." in *Proc. ISPD*, 2020.

[18] K. Kunal, T. Dhar, M. Madhusudan, J. Poojary, A. Sharma, W. Xu, S. M. Burns, J. Hu, R. Harjani, and S. S. Sapatnekar, "GANA: graph convolutional network based automated netlist annotation for analog circuits," in *Proc. DATE*, 2020.

[19] X. Gao, C. Deng, M. Liu, Z. Zhang, D. Z. Pan, and Y. Lin, "Layout symmetry annotation for analog circuits with graph neural networks," in *Proc. ASPDAC*, 2021.

[20] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "GCN-RL circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *Proc. DAC*, 2020.

[21] Y. Li, Y. Lin, M. Madhusudan, A. Sharma, W. Xu, S. S. Sapatnekar, R. Harjani, and J. Hu, "A customized graph neural network model for guiding analog IC placement," in *Proc. ICCAD*, 2020.

[22] H. Chen, K. Zhu, M. Liu, X. Tang, N. Sun, and D. Z. Pan, "Universal symmetry constraint extraction for analog and mixed-signal circuits with graph neural networks," in *Proc. DAC*, 2021.

[23] K. Kunal, M. Madhusudan, A. K. Sharma, W. Xu, S. M. Burns, R. Harjani, J. Hu, D. A. Kirkpatrick, and S. S. Sapatnekar, "Align: Open-source analog layout automation from the ground up," in *Proc. DAC*, 2019.

[24] B. Xu, K. Zhu, M. Liu, Y. Lin, S. Li, X. Tang, N. Sun, and D. Z. Pan, "Magical: Toward fully automated analog IC layout leveraging human and machine intelligence," in *Proc. ICCAD*, 2019.

[25] B. Xu, S. Li, X. Xu, N. Sun, and D. Z. Pan, "Hierarchical and analytical placement techniques for high-performance analog circuits," in *Proc. ISPD*, 2017.

[26] B. Xu, B. Basaran, M. Su, and D. Z. Pan, "Analog placement constraint extraction and exploration with the application to layout retargeting," in *Proc. ISPD*, 2018.

[27] J. Pak, B. Yu, and D. Z. Pan, "Electromigration-aware redundant via insertion," in *Proc. ASPDAC*, 2015.

[28] F. Jiao, S. Dong, B. Yu, B. Li, and U. Schlichtmann, "Thermal-aware placement and routing for 3D optical networks-on-chips," in *Proc. ISCAS*, 2018.

[29] W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Batch Bayesian optimization via multi-objective acquisition ensemble for automated analog circuit design," in *Proc. ICML*, 2018.

[30] S. Zhang, W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Bayesian optimization approach for analog circuit synthesis using neural network," in *Proc. DATE*, 2019.

[31] M. Liu, K. Zhu, X. Tang, B. Xu, W. Shi, N. Sun, and D. Z. Pan, "Closing the design loop: Bayesian optimization assisted hierarchical analog layout synthesis," in *Proc. DAC*, 2020.

[32] Z. He, L. Zhang, P. Liao, Y. Ma, and B. Yu, "Reinforcement learning driven physical synthesis," in *Proc. ICSICT*, 2020.

[33] M. Liu, K. Zhu, J. Gu, L. Shen, X. Tang, N. Sun, and D. Z. Pan, "Towards decrypting the art of analog layout: Placement quality prediction via transfer learning," in *Proc. DATE*, 2020.

[34] K. Zhu, M. Liu, Y. Lin, B. Xu, S. Li, X. Tang, N. Sun, and D. Z. Pan, "Geniusroute: A new analog routing paradigm using generative neural network guidance." in *Proc. ICCAD*, 2019.

[35] Q. Sun, T. Chen, S. Liu, J. Miao, J. Chen, H. Yu, and B. Yu, "Correlated multi-objective multi-fidelity optimization for HLS directives design," in *Proc. DATE*, 2021.

[36] S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value entropy search for multi-objective Bayesian optimization with constraints," *arXiv preprint arXiv:2009.01721*, 2020.

[37] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. Eslami, and Y. W. Teh, "Neural processes," *arXiv preprint arXiv:1807.01622*, 2018.

[38] A. Carr and D. Wingate, "Graph neural processes: Towards Bayesian graph neural networks," *arXiv preprint arXiv:1902.10042*, 2019.

[39] S. Liu, Q. Sun, P. Liao, Y. Lin, and B. Yu, "Global placement with deep learning-enabled explicit routability optimization," *Proc. DATE*, 2021.