Performance-driven Analog Routing via Heterogeneous 3DGNN and Potential Relaxation

Peng Xu\textsuperscript{1,2}, Guojin Chen\textsuperscript{1}, Keren Zhu\textsuperscript{1}, Tinghuan Chen\textsuperscript{2}, Tsung-Yi Ho\textsuperscript{1}, Bei Yu\textsuperscript{1}

\textsuperscript{1}Chinese University of Hong Kong
\textsuperscript{2}Chinese University of Hong Kong, Shenzhen
Outline

1 Background

2 Method
   2.1 Performance-Driven Analog Routing
   2.2 Non-uniform Routing Guidance
   2.3 AnalogFold Framework for Performance Prediction and Relaxation

3 Experiments
Background Knowledge
Analog circuit routing is critical to optimal performance, but obtaining a decent circuit layout requires significant time and expertise.
Ou et al. propose different levels of geometrical matching constraints\textsuperscript{1}.

\textbf{Existing Methods: Heuristic Constraint-based Methods}

There are other works that optimize power routing\(^2\) and propose shielding critical nets\(^3\).


A ML-Guided Analog Routing Problem

Can we automatically summarize the human layout intelligence leveraging ML?\(^4\)

**Heuristic constraints**
Use a set of detailed heuristics as routing constraints.

**Routing guidance**
Routing strategies learned from human

Human Layout data

- Pre-process the GDS layouts into images
- Extract training data where the human would likely route the nets
- Problem #1 The human experts’ layout data is **pretty scarce**.
2D Uniform Routing Guidance

- Predict a 2D probability map of the routing likelihoods in each region.
- The 2D uniform routing guidance is honored via penalties in the cost function.
- Problem #2 Fail to deal with designs of different sizes or aspect ratios and resource competition between different pins close to each other.
Leveraging variational autoencoder (VAE) to reconstruct the routing solutions.

Minize the distance between ground truth and inferred output.

Problem #3 The generative model makes it hard to guarantee a performance boost.
Proposed Method: PARoute
• We introduce a performance-driven analog routing approach.

• Learn from the **automatically generated routing patterns** and their **simulation results** without **human labeling effort**.
Problem #2: Non-uniform Routing Guidance

(a) Two examples of non-uniform routing guidance; (b) The 3D visualization.

- We propose a non-uniform and adaptive routing guidance, which assigns different routing guidance $c_i$ along different directions for each net $n_i$.
- Adapt the route guide distribution to areas with different densities and support a 3D cost map.
• We proposed a customized AnalogFold framework to enable accurate modeling of the performance potential of routing guidance.

• AnalogFold contains a heterogeneous routing graph, a protein-inspired 3DGNN network, and a pool-aided potential relaxation process.
We design a heterogeneous graph $G_H = < V_{AP}, V_M, E_{PP}, E_{PM}, E_{MM} >$ to represent the interactions between pin access points and modules.

- The vertex sets $V_{AP}$ and $V_M$ correspond to the pin access points and modules.
- $E_{PP}$ is designed to reflect the interactions between different pin access points.
- $E_{MM}$ contains the edges that connect the modules according to the netlist.
- We add the edge $E_{PM}$ to model the relationship between the pin access points and the modules.
We can define the distance honors routing cost as follows:

\[ d_{\text{cost}}(v_k, v_s) = \sqrt{(c[0] \cdot h_{ks})^2 + (c[1] \cdot w_{ks})^2 + (c[2] \cdot z_{ks})^2}, \]  

where \( c \) is the cost guide assigned for each access point, \( h_{ks} / w_{ks} / z_{ks} \) is the distance between \( v_k \) and \( v_s \) along horizontal/vertical/Z-axis direction. The distance between nodes is embedded to reflect the routing resource competition.
In 3D-GNN, the proposed cost-aware message passing can be defined as:

$$e_k^l = \phi^e \left( e_k, v_{r_k}, v_{s_k}, \mathcal{E}_{s_k}, \rho^{p\rightarrow e} \left( \{r_h\}_{h=r_k \cup s_k} \right) \right),$$

$$v_i^l = \phi^v \left( v_i, \rho^{e\rightarrow v} \left( \mathcal{E}_i^l \right) \right), u^l = \phi^u \left( u, \rho^{v\rightarrow u} \left( \mathcal{V}^l \right) \right),$$

where $\phi^e$, $\phi^v$, and $\phi^u$ are three information update functions on edges, pin access points/modules, and the whole graph, respectively. Especially, the 3D information in $P$ is incorporated to update each message $e_k$. 
We created a differentiable model using the 3DGNN to predict the post-layout performance of the routing guidance.

We then apply a gradient-based optimization of routing guidance potential \textit{multiple times with different initialization} to derive the top-$N$ routing guidance results.
Experiment Results
### Post-layout Performance Comparisons on OTA benchmarks

**Table:** The comparisons between baseline methods and the proposed method.

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Schematic</th>
<th>MagicalRoute(^5)</th>
<th>GeniusRoute(^6)</th>
<th>PARoute (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset Voltage((\mu V)) ↓</td>
<td>-</td>
<td>1.000</td>
<td>10.426</td>
<td>0.546</td>
</tr>
<tr>
<td>CMRR(dB) ↑</td>
<td>-</td>
<td>1.000</td>
<td>0.998</td>
<td>1.163</td>
</tr>
<tr>
<td>BandWidth(MHz) ↑</td>
<td>-</td>
<td>1.000</td>
<td>1.002</td>
<td>1.113</td>
</tr>
<tr>
<td>DC Gain(dB) ↑</td>
<td>-</td>
<td>1.000</td>
<td>0.999</td>
<td>2.368</td>
</tr>
<tr>
<td>Noise((\mu V_{rms})) ↓</td>
<td>-</td>
<td>1.000</td>
<td>1.007</td>
<td>0.787</td>
</tr>
<tr>
<td>Runtime(s) ↓</td>
<td>-</td>
<td><strong>1.000</strong></td>
<td>17.147</td>
<td>7.480</td>
</tr>
</tbody>
</table>


• Although the average runtime of our proposed approach is $7.48 \times$ slower than MagicalRoute\textsuperscript{7}, it is nearly $2.29 \times$ faster than GeniusRoute\textsuperscript{8} due to the simplified 3D graph structure.

• The most consuming part is the model training part, which takes 80.22\% of the total runtime and 3.71\% of the total time for the routing cost generation.


THANK YOU!