WHERE **INNOVATION** BEGINS

**DESIGN AUTOMATION CONFERENCE**

# JULY 9–13, 2023

**MOSCONE WEST CENTER**
**SAN FRANCISCO, CA, USA**

# Mitigating Distribution Shift for Congestion Optimization in Global Placement

**Su Zheng**[1,2], Lancheng Zou[1], Siting Liu[1],
Yibo Lin[2], Bei Yu[1], Martin Wong[1]

[1]Chinese University of Hong Kong
[2]Peking University

# Outline

# Introduction

# Placement and Congestion Modeling

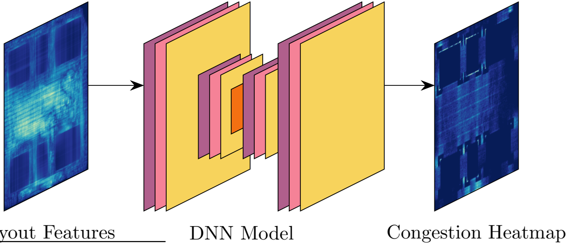
- Placement is crucial but time-consuming
- Congestion modeling and optimization is important
- Congestion optimization techniques
  - Trial global routing[1]
  - Analytical model[2]

[1] C.-C. Huang *et al.*, "NTUplace4dr: A detailed-routing-driven placer for mixed-size circuit designs with technology and region constraints", *IEEE TCAD*, vol. 37, no. 3, pp. 669–681, 2018.

[2] Y. Wei *et al.*, "GLARE: Global and local wiring aware routability evaluation", in *Proc. DAC*, 2012, pp. 768–773.

# Congestion Modeling via Deep Learning

- Fully Convolutional Networks[3]
- Generative Adversarial Networks[4]
- Graph Neural Networks[5]



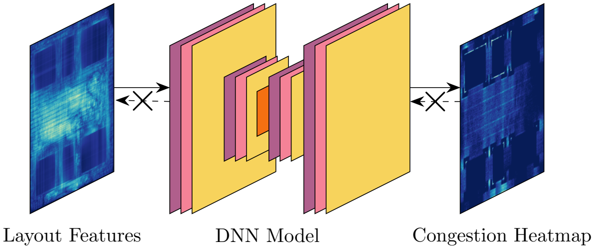

[3] Z. Xie *et al.*, "RouteNet: Routability prediction for mixed-size designs using convolutional neural network", in *Proc. ICCAD*, 2018, 80:1–80:8.

[4] C. Yu and Z. Zhang, "Painting on placement: Forecasting routing congestion using conditional generative adversarial nets", in *Proc. DAC*, 2019.

[5] B. Wang *et al.*, "LHNN: Lattice hypergraph neural network for VLSI congestion prediction", in *Proc. DAC*, 2022, pp. 1297–1302.

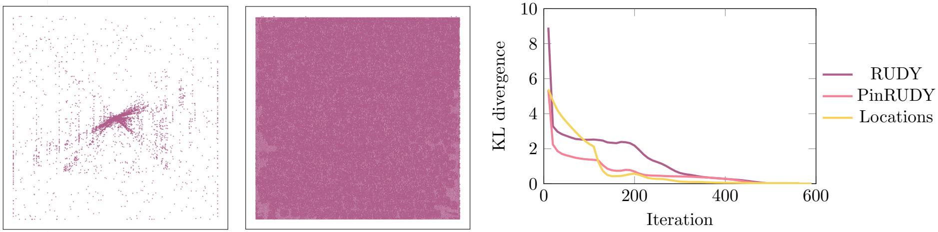
# Problems of Existing Methods

- Observations: **prediction only**, useless in placement



# Problems of Existing Methods

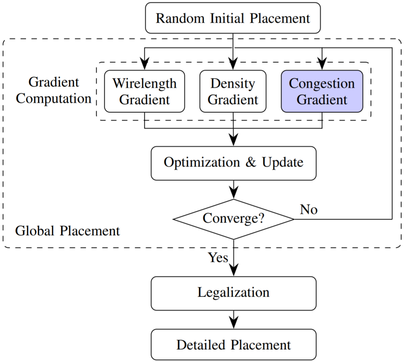
- Observations: **distribution shift** during placement



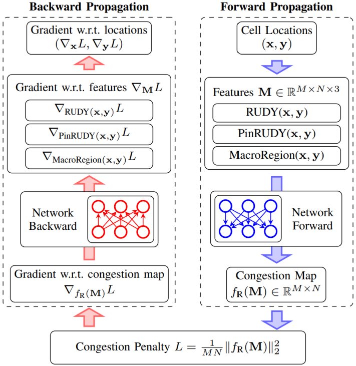

(a) (b) (c)

# Solutions

- Congestion-driven Placement with DNN



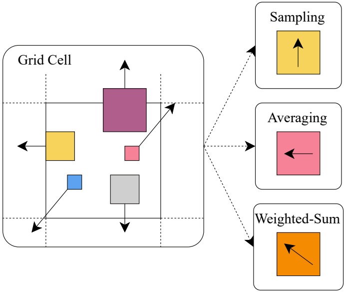

(a)



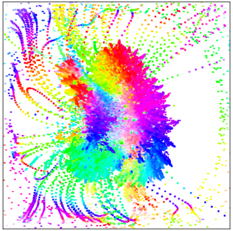

(b)

# Solutions

- Look-ahead via Cell Flow Prediction




(a)

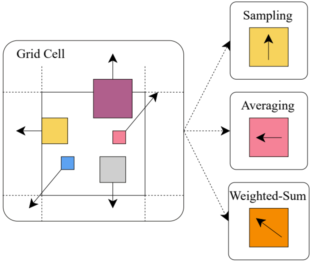


(b)

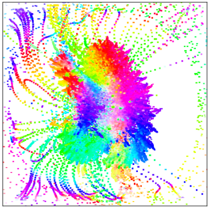
# Proposed Method

# Cell Flow Prediction

- Cell flow measures the motions of the cells
  - $c_i'(x_{i,j}, y_{i,j}) = (x_{i,j} - x_{i-K,j}, y_{i,j} - y_{i-K,j})$
  - $(x_{i,j}, y_{i,j})$ is cell $j$'s location at $i$-th iteration, $K$ is the step size
  - Inspired by optical flow



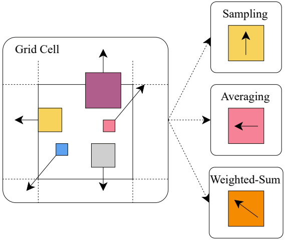

(a)



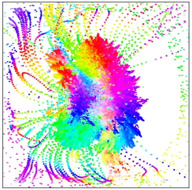
(b)

# Quasi-voxelization

- Grid-cell $b_{k,l}$ contains multiple cells, we need downsampling
  - Sampling: $\boldsymbol{c}_i(k,l) = s_{\hat{j}}\boldsymbol{c}'_i(x_{\hat{j}}, y_{\hat{j}}), \quad \hat{j} = \arg\max_j s_j, (x_{i,j}, y_{i,j}) \in b_{k,l}$.
  - Averaging: $\boldsymbol{c}_i(k,l) = \frac{1}{N_{k,l}} \sum_{(x_{i,j},y_{i,j})\in b_{k,l}} \boldsymbol{c}'_i(x_{i,j}, y_{i,j})$.
  - Weighted-sum: $\boldsymbol{c}_i(k,l) = \sum_{(x_{i,j},y_{i,j})\in b_{k,l}} \frac{s_j}{N_{k,l}} \times \boldsymbol{c}'_i(x_{i,j}, y_{i,j})$. ✓



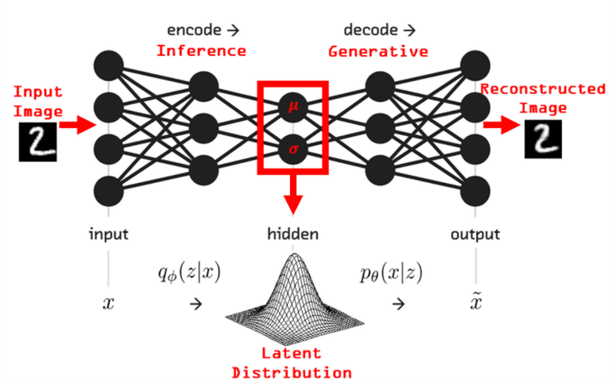

(a)



(b)

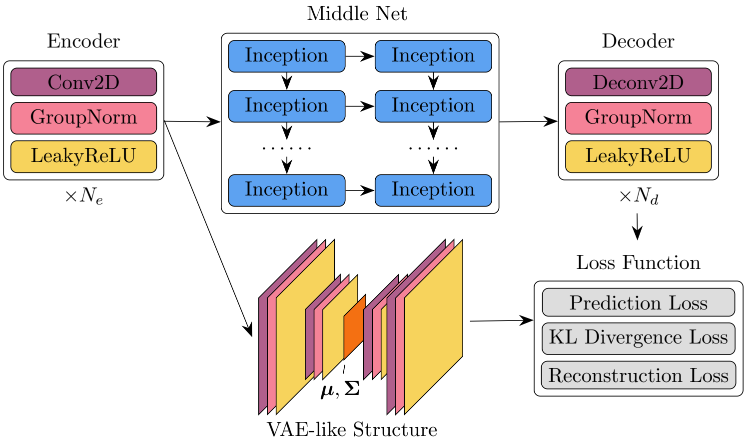
# Invariant Feature Space

- Invariant feature space learning



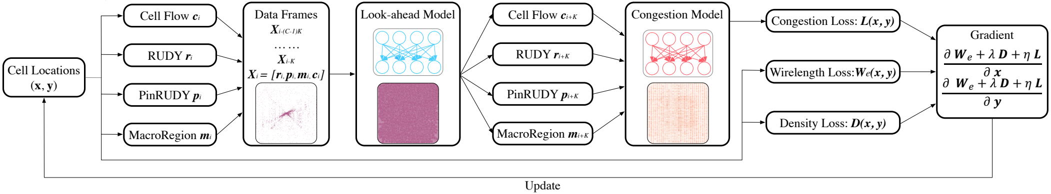
# Complete Flow

- Cell flow prediction + invariant feature space learning



# Complete Flow

- DREAMPlace[6] + Look-ahead Congestion Optimization




[6] Y. Lin *et al.*, "DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement", in *Proc. DAC*, 2019.

# Experiments

# Experimental Settings

- Placement Platform: DREAMPlace[7]
- Baseline: DREAM-Cong[8]
- Congestion Prediction Metrics:

$$\mathrm{NRMS}(\overline{\boldsymbol{Y}}, \boldsymbol{Y}) = \frac{\|\overline{\boldsymbol{Y}} - \boldsymbol{Y}\|_2}{(Y_{\max} - Y_{\min})\sqrt{N_Y}}, \tag{1}$$

$$\mathrm{SSIM}(\overline{\boldsymbol{Y}}, \boldsymbol{Y}) = \frac{(2\mu_{\boldsymbol{Y}}\mu_{\overline{\boldsymbol{Y}}} + C_1)(2\sigma_{\boldsymbol{Y},\overline{\boldsymbol{Y}}} + C_2)}{(\mu_{\boldsymbol{Y}}^2 + \mu_{\overline{\boldsymbol{Y}}}^2 + C_1)(\sigma_{\boldsymbol{Y}}^2 + \sigma_{\overline{\boldsymbol{Y}}}^2 + C_2)}. \tag{2}$$

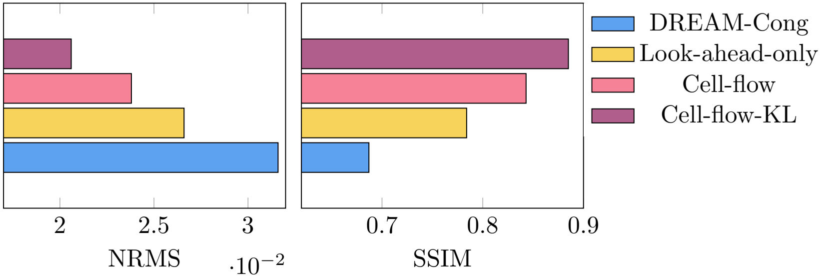
- Placement Metrics: (Given by Innovus)
  - Wire Length (WL)
  - Worst Congestion Score (WCS)

---

[7] Y. Lin *et al.*, "DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement", in *Proc. DAC*, 2019.

[8] S. Liu *et al.*, "Global placement with deep learning-enabled explicit routability optimization", in *Proc. DATE*, 2021.

# Comparison on Congestion Prediction




(a) (b)

# Comparison on Congestion Optimization

| Benchmark | #Cells | #Nets | DREAMPlace | | | DREAM-Cong | | | LACO | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $WCS_H$ | $WCS_V$ | WL($10^5 \mu m$) | $WCS_H$ | $WCS_V$ | WL($10^5 \mu m$) | $WCS_H$ | $WCS_V$ | WL($10^5 \mu m$) |
| des_perf_1 | 113k | 113k | 0.47 | 0.40 | 13.88 | 0.47 | 0.40 | 13.82 | 0.40 | 0.40 | 13.87 |
| des_perf_a | 109k | 110k | 2.25 | 1.67 | 22.21 | 1.89 | 1.60 | 22.33 | 1.69 | 1.30 | 22.27 |
| des_perf_b | 113k | 113k | 0.07 | 0.27 | 16.70 | 0.13 | 0.27 | 16.71 | 0.07 | 0.20 | 16.57 |
| edit_dist_a | 130k | 131k | 4.05 | 4.14 | 53.54 | 4.30 | 4.07 | 53.62 | 3.50 | 3.14 | 53.40 |
| fft_1 | 35k | 33k | 0.59 | 0.40 | 4.96 | 0.43 | 0.47 | 4.95 | 0.46 | 0.40 | 4.91 |
| fft_2 | 35k | 33k | 0.40 | 0.78 | 5.86 | 0.36 | 0.67 | 5.88 | 0.27 | 0.61 | 5.84 |
| fft_a | 34k | 32k | 0.55 | 0.56 | 10.56 | 0.83 | 0.77 | 10.53 | 0.50 | 0.56 | 10.52 |
| fft_b | 34k | 32k | 3.50 | 2.33 | 12.13 | 3.50 | 2.67 | 12.16 | 3.33 | 2.33 | 12.12 |
| matrix_mult_1 | 160k | 159k | 0.71 | 0.53 | 25.85 | 0.88 | 0.58 | 28.95 | 0.68 | 0.44 | 25.83 |
| matrix_mult_2 | 160k | 159k | 0.65 | 0.42 | 25.71 | 0.78 | 0.84 | 29.99 | 0.61 | 0.45 | 25.71 |
| matrix_mult_a | 154k | 154k | 0.47 | 0.40 | 36.99 | 0.44 | 0.37 | 37.02 | 0.47 | 0.37 | 36.78 |
| matrix_mult_b | 151k | 152k | 8.69 | 2.65 | 35.08 | 8.69 | 2.65 | 35.29 | 8.69 | 2.65 | 35.07 |
| matrix_mult_c | 151k | 152k | 0.53 | 0.40 | 35.42 | 0.50 | 0.27 | 35.97 | 0.47 | 0.30 | 35.42 |
| pci_bridge32_a | 30k | 30k | 2.06 | 0.84 | 6.12 | 1.83 | 0.87 | 6.14 | 1.89 | 0.95 | 6.13 |
| pci_bridge32_b | 29k | 29k | 0.03 | 0.23 | 9.77 | 0.14 | 0.31 | 10.57 | 0.10 | 0.20 | 9.65 |
| superblue11_a | 954k | 936k | 1.10 | 25.00 | 392.78 | 1.15 | 23.00 | 396.98 | 1.10 | 25.00 | 392.93 |
| superblue12 | 1293k | 1293k | 3.00 | 3.00 | 414.10 | 2.73 | 2.57 | 414.12 | 2.45 | 2.57 | 413.95 |
| superblue14 | 634k | 620k | 1.10 | 4.17 | 277.32 | 1.06 | 4.67 | 277.69 | 1.00 | 3.50 | 277.97 |
| superblue16_a | 698k | 697k | 0.91 | 10.75 | 309.04 | 1.00 | 10.00 | 310.17 | 1.00 | 9.75 | 309.03 |
| superblue19 | 522k | 512k | 1.70 | 3.67 | 201.34 | 1.30 | 4.33 | 202.36 | 1.57 | 3.50 | 201.27 |
| Average | - | - | 1.64 | 3.13 | 95.47 | 1.62 | 3.07 | 96.22 | **1.51** | **2.93** | **95.46** |
| Ratio | - | - | 1.00 | 1.00 | 1.00 | 0.99 | 0.98 | 1.01 | **0.92** | **0.94** | **1.00** |

# Conclusion

# Conclusion

- Look-ahead, cell flow, invariant feature space learning bring better congestion prediction
- More accurate congestion prediction leads to better congestion optimization
- Up to 8% improvement in the maximum routing overflow

# THANK YOU!