

# GCN-RL Circuit Designer: Transferable Transistor Sizing with Graph Neural Networks and Reinforcement Learning

Hanrui Wang<sup>1</sup>, Kuan Wang<sup>1</sup>, Jiacheng Yang<sup>1</sup>, Linxiao Shen<sup>2</sup>, Nan Sun<sup>2</sup>,  
Hae-Seung Lee<sup>1</sup>, Song Han<sup>1</sup>

<sup>1</sup>Massachusetts Institute of Technology

<sup>2</sup>UT Austin



# AI is Revolutionizing EDA

- Fast: 
  - Inference of an AI model is fast thanks to well-optimized libraries (e.g. cuDNN)
  - Can be accelerated by GPUs and AI accelerators
- Accurate: 
  - Data-driven
  - With enough data, performance of AI models can exceed traditional methods
  - Enable continuous learning with newly obtained data

# AI is Revolutionizing EDA

- AI for Physical Design & Manufacture:
  - “DreamPlace”<sup>1</sup> for placement
  - “LithoGAN”<sup>2</sup> for lithography modeling
  - “Google’s Chip Design AI”<sup>3</sup> for floorplaning
- AI for Circuit Design
  - “Circuits-GNN”<sup>4</sup> for RF circuits
  - “Learning to Design Circuits”<sup>5</sup> for Analog IC
  - “Analog and Digital Circuits Classifier”<sup>6</sup> for sub-circuits classification
- AI for System-Level Modeling & Optimization
  - “PowerNet”<sup>7</sup> for power modeling
  - “Resource Management with RL”<sup>8</sup> for many-core resources management
  - “Combine Evolutionary with Deep Learning”<sup>9</sup> for Interface Optimization

<sup>1</sup>Lin, Y., Dhar, S., Li, W., Ren, H., Khailany, B., & Pan, D. Z. DREAMPlace: Deep learning toolkit-enabled GPU acceleration for modern VLSI placement. In DAC 2019.

<sup>2</sup>Ye, W., Alawieh, M. B., Lin, Y., & Pan, D. Z. Lithogan: End-to-end lithography modeling with generative adversarial networks. In DAC 2019.

<sup>3</sup>Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J., Songhor, E., Wang, S., ... & Nazi, A. (2020). Chip Placement with Deep Reinforcement Learning. arXiv preprint arXiv:2004.10746.

<sup>4</sup>Zhang, G., He, H., & Katabi, D. (2019, May). Circuit-GNN: Graph Neural Networks for Distributed Circuit Design. In *International Conference on Machine Learning* (pp. 7364-7373).

<sup>5</sup>Wang, H., Yang, J., Lee, H. S., & Han, S. (2018). Learning to design circuits. *NeurIPS 2018, ML for System Workshop*.

<sup>6</sup>Liou, G. H., Wang, S. H., Su, Y. Y., & Lin, M. P. H. (2018, July). Classifying Analog and Digital Circuits with Machine Learning Techniques Toward Mixed-Signal Design Automation. In *SMACD 2018*

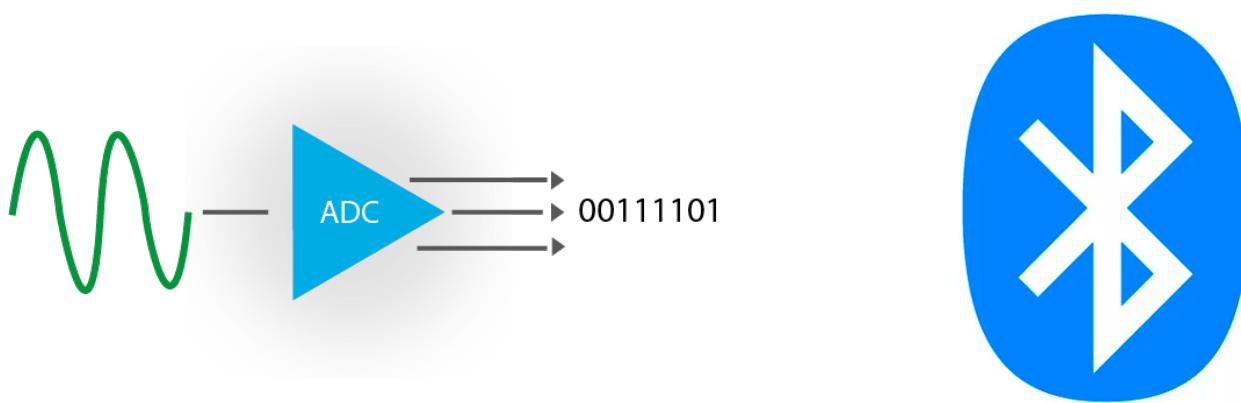
<sup>7</sup>Chen, J., Alawieh, M. B., Lin, Y., Zhang, M., Zhang, J., Guo, Y., & Pan, D. Z. (2020). Powernet: SOI Lateral Power Device Breakdown Prediction With Deep Neural Networks. *IEEE Access*

<sup>8</sup>Mao, H., Alizadeh, M., Menache, I., & Kundula, S. Resource management with deep reinforcement learning. In *15th ACM Workshop on Hot Topics in Networks*.

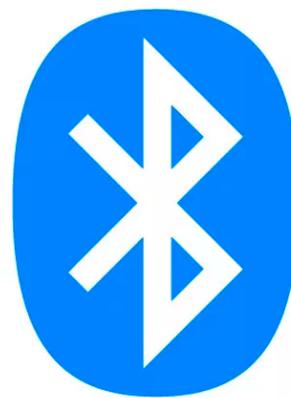
<sup>9</sup>Servadei, L., Mosca, E., Werner, M., Esen, V., Wille, R., & Ecker, W. Combining Evolutionary Algorithms and Deep Learning for Hardware/Software Interface Optimization.



# Analog IC is Ubiquitous



Analog-Digital Convertor

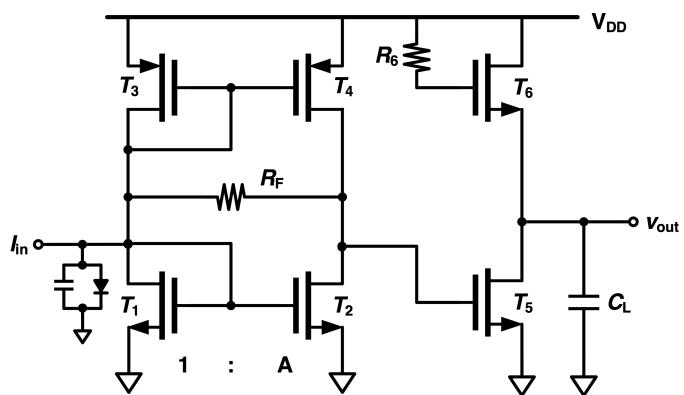


Bluetooth

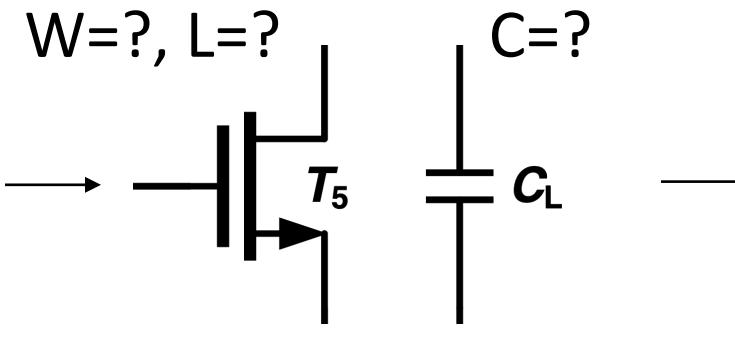


WiFi

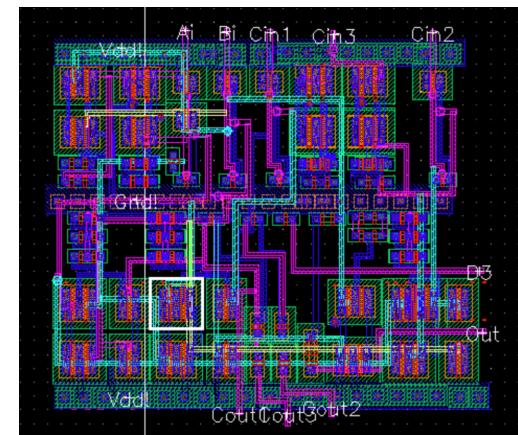
# Analog IC Design



Schematic Design



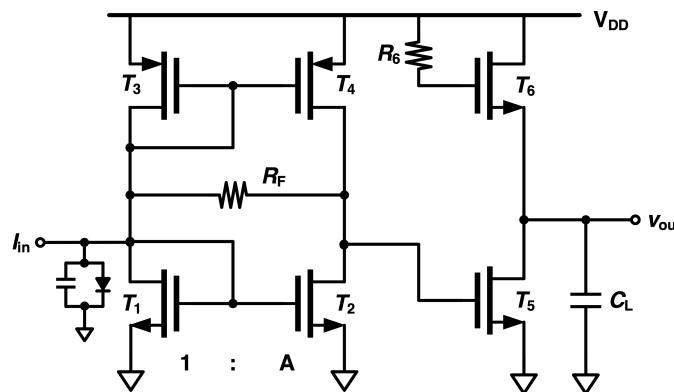
Transistor Sizing



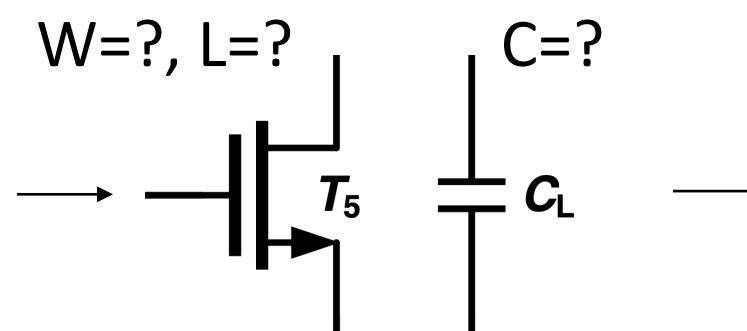
Layout Design

# Analog IC is Labor Intensive for Design & Porting

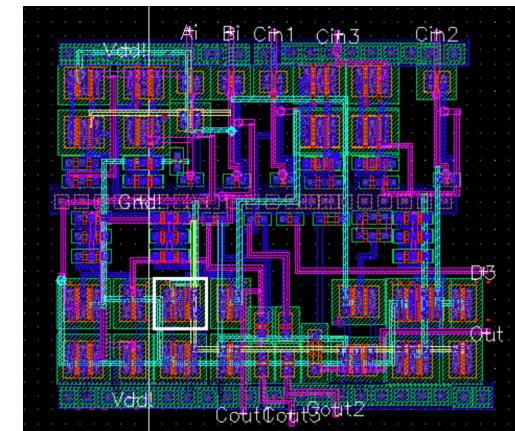
- Analog IC Design
  - **Manual** schematic design
  - **Manual** transistor sizing
  - **Manual** layout design
  - Lacks Automation & Labor Intensive



Schematic Design



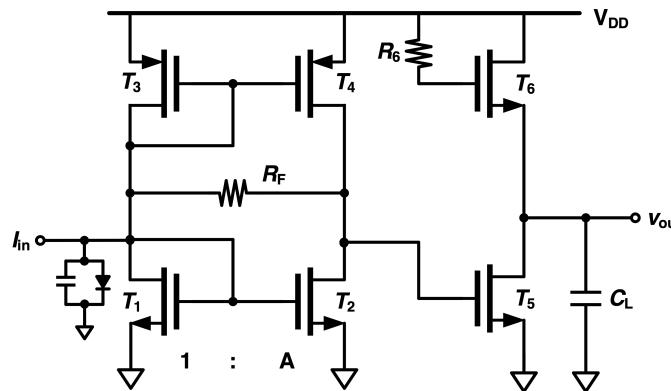
Transistor Sizing



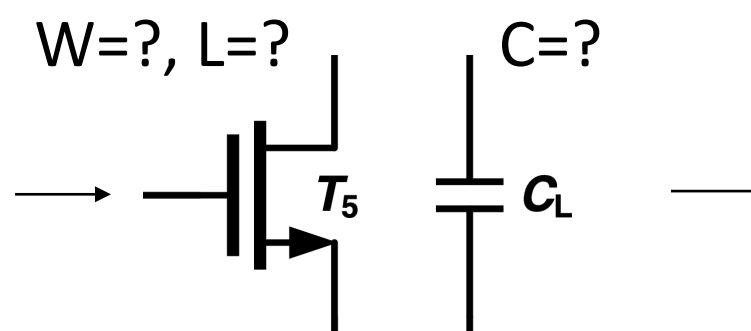
Layout Design

# Analog IC is Labor Intensive for Design & Porting

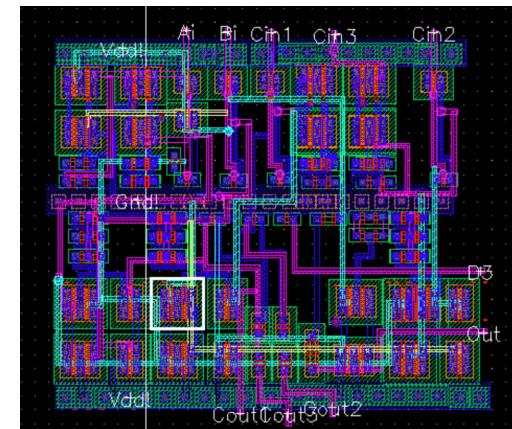
- Analog IC Design
  - **Manual** schematic design
  - **Manual** transistor sizing
  - **Manual** layout design
  - Lacks Automation & Labor Intensive
- Analog IC Porting
  - Inherit schematic
  - **Manual** transistor sizing again
  - **Manual** layout design again
  - **Large** Overhead & Effort



Schematic Design



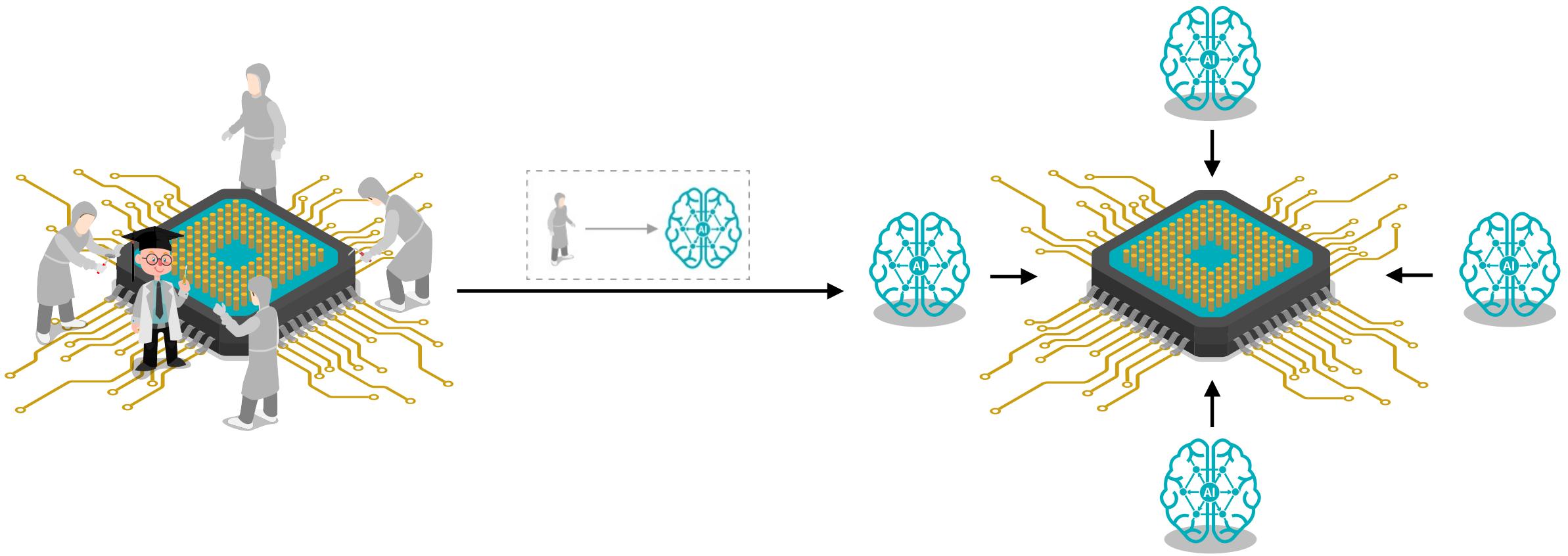
Transistor Sizing



Layout Design



# GCN-RL Circuit Designer



- AI for higher analog IC design efficiency

# GCN-RL Circuit Designer

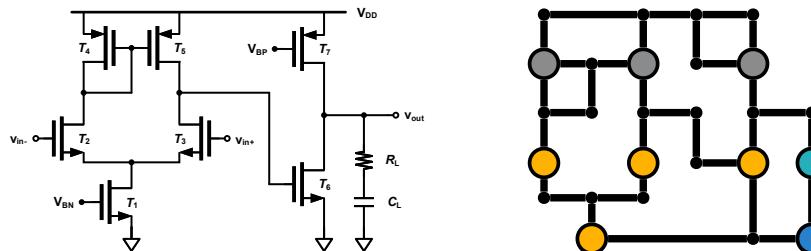
- Target at the challenging transistor sizing problem
  - Large design space
  - Complex trade-offs



# GCN-RL Circuit Designer

- Target at the challenging transistor sizing problem
  - Large design space
  - Complex trade-offs
- Leverage Graph Convolutional Neural Networks (GCN)

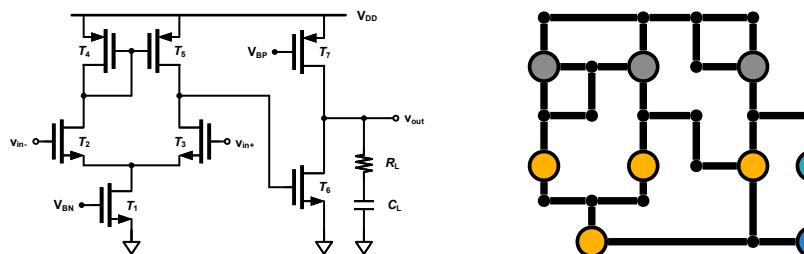
- **Circuit is a graph**



- Involve graph information into optimization loop

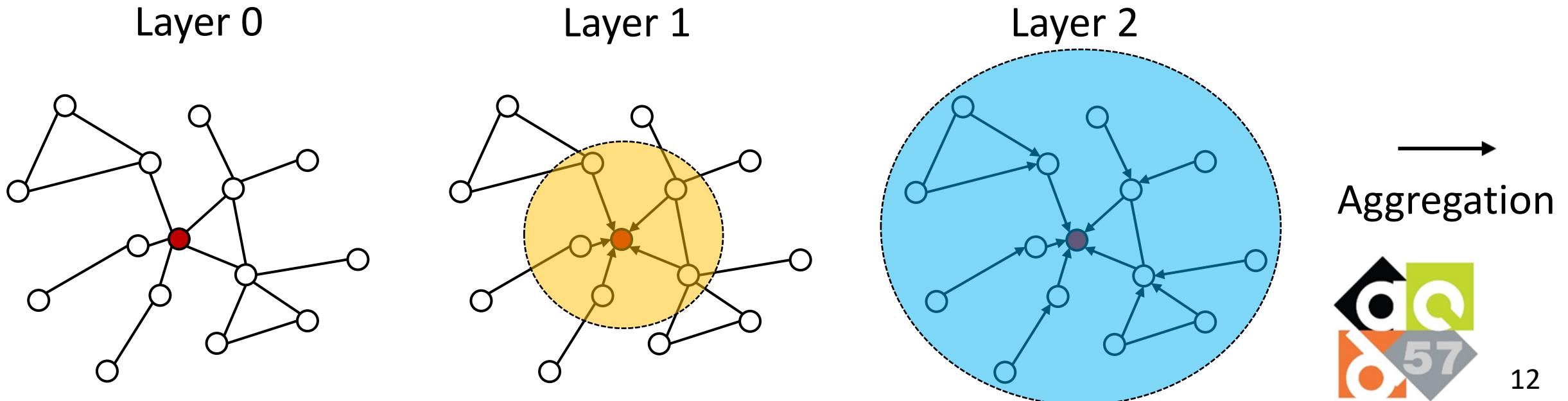
# GCN-RL Circuit Designer

- Target at the challenging transistor sizing problem
  - Large design space
  - Complex trade-offs
- Leverage Graph Convolutional Neural Networks (GCN)
  - **Circuit is a graph**
    - Involve graph information into optimization loop
- Transfer learning to reduce design porting overhead
  - Between technology nodes
  - Between different circuit topologies

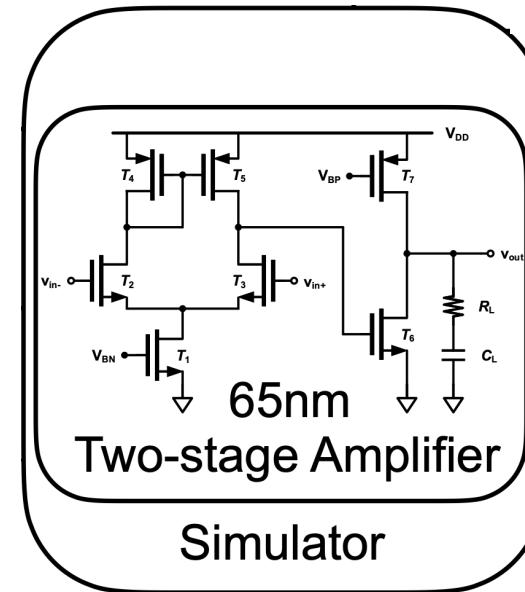


# GCN-RL Circuit Designer: What is GCN?

- Graph Convolutional Neural Networks (GCN) [Kipf et al, 2017]
  - Each node aggregates information from neighbors
  - The more layers, the larger receptive field of each node



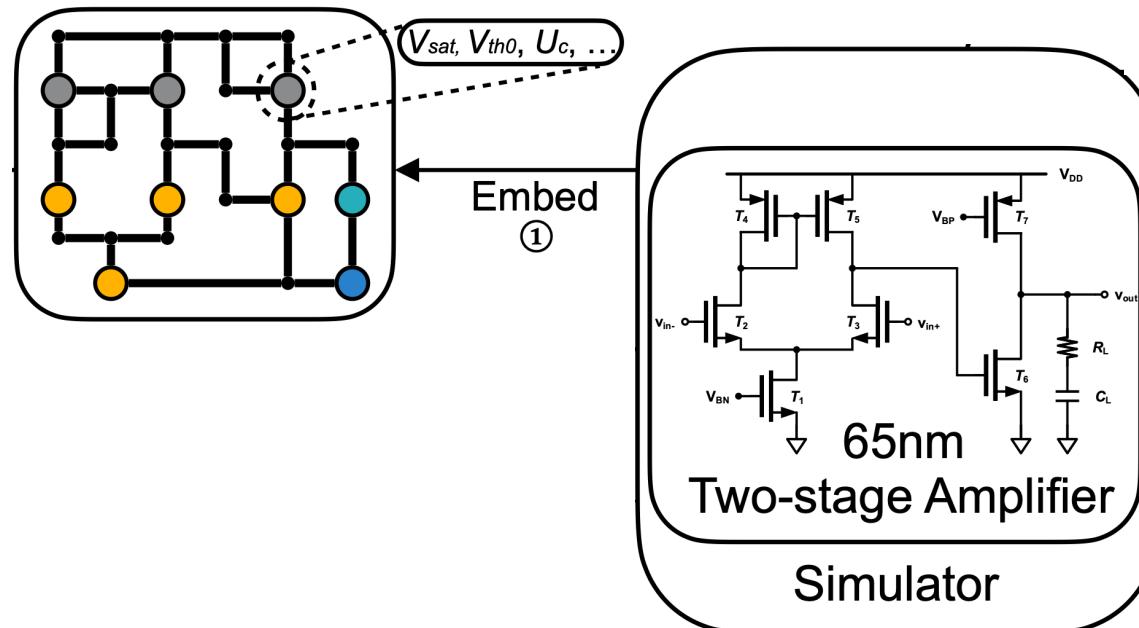
# GCN-RL Circuit Designer Overview



- Step 0: We have a target circuit and a simulator as environment

# One Optimization Iteration Contains 6 Steps

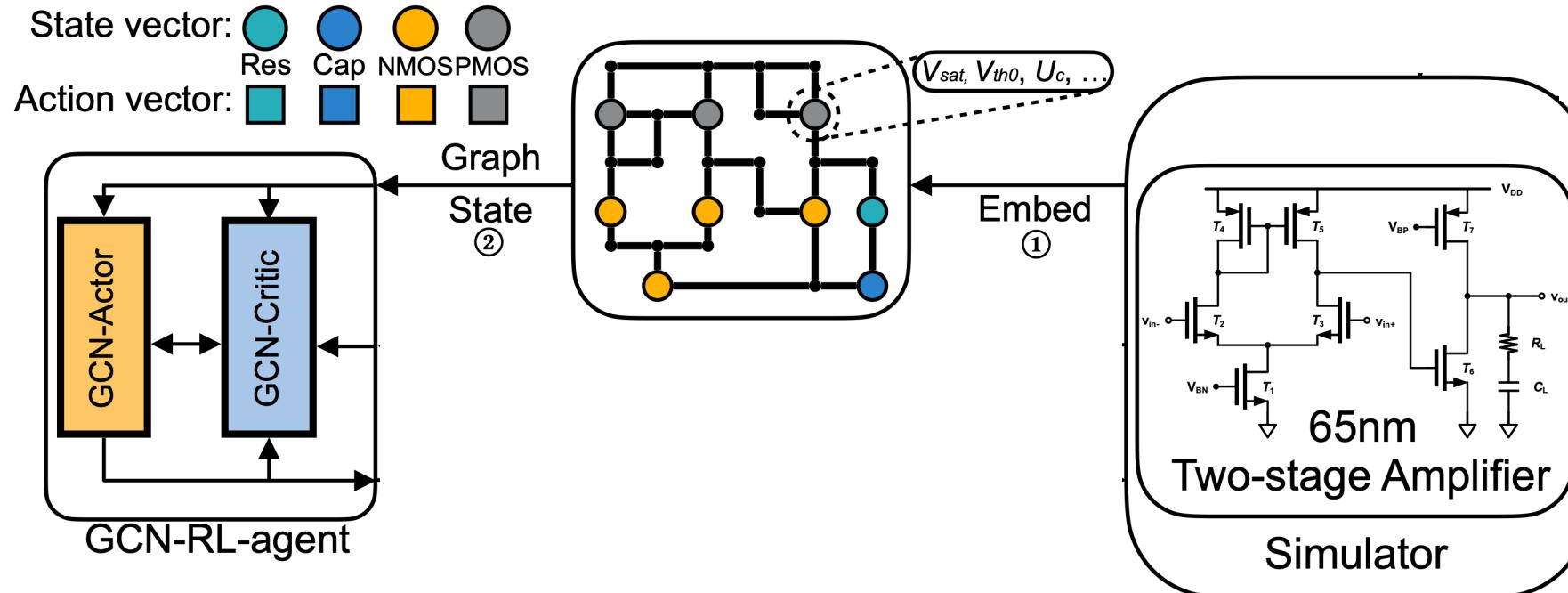
State vector: Res Cap NMOS PMOS  
Action vector: Res Cap NMOS PMOS



- Step 1: Environment embeds the topology into a graph whose nodes are components (transistor, resistor) and edges are wires

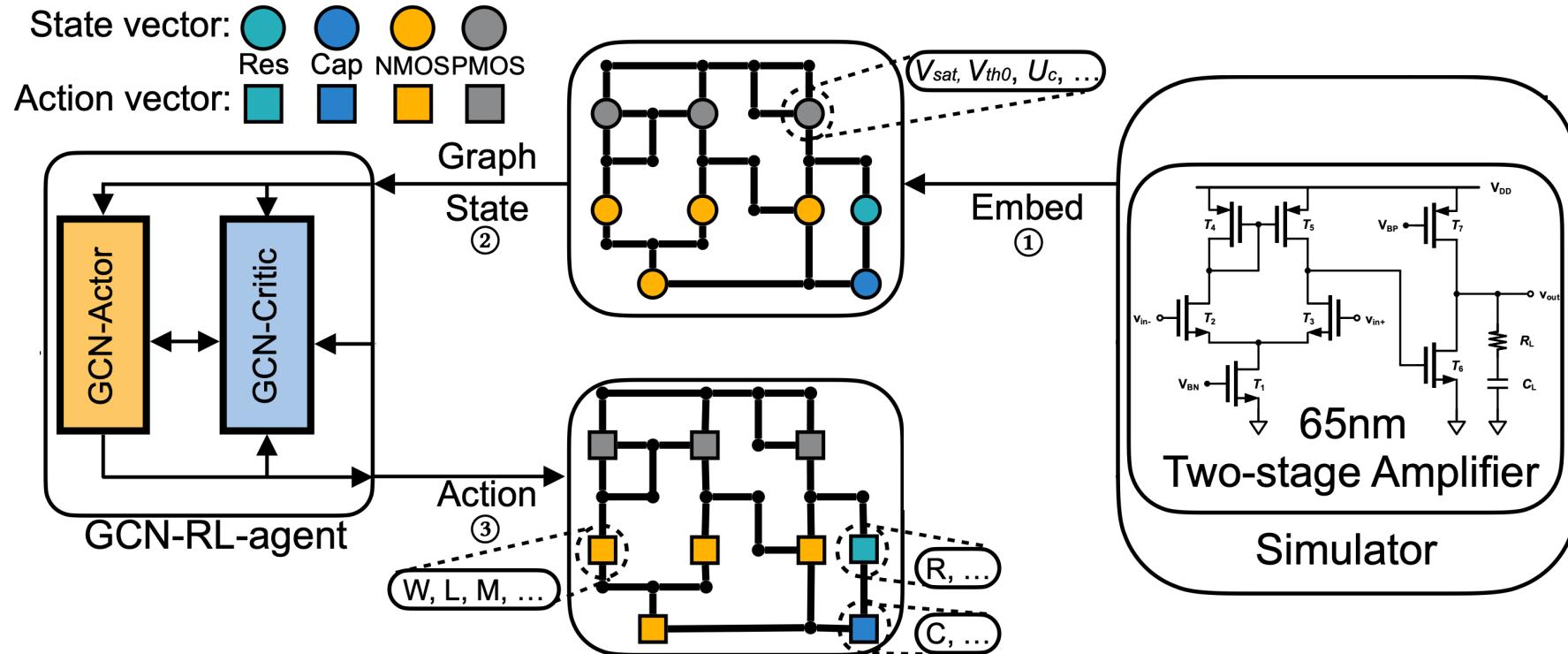


# One Optimization Iteration Contains 6 Steps



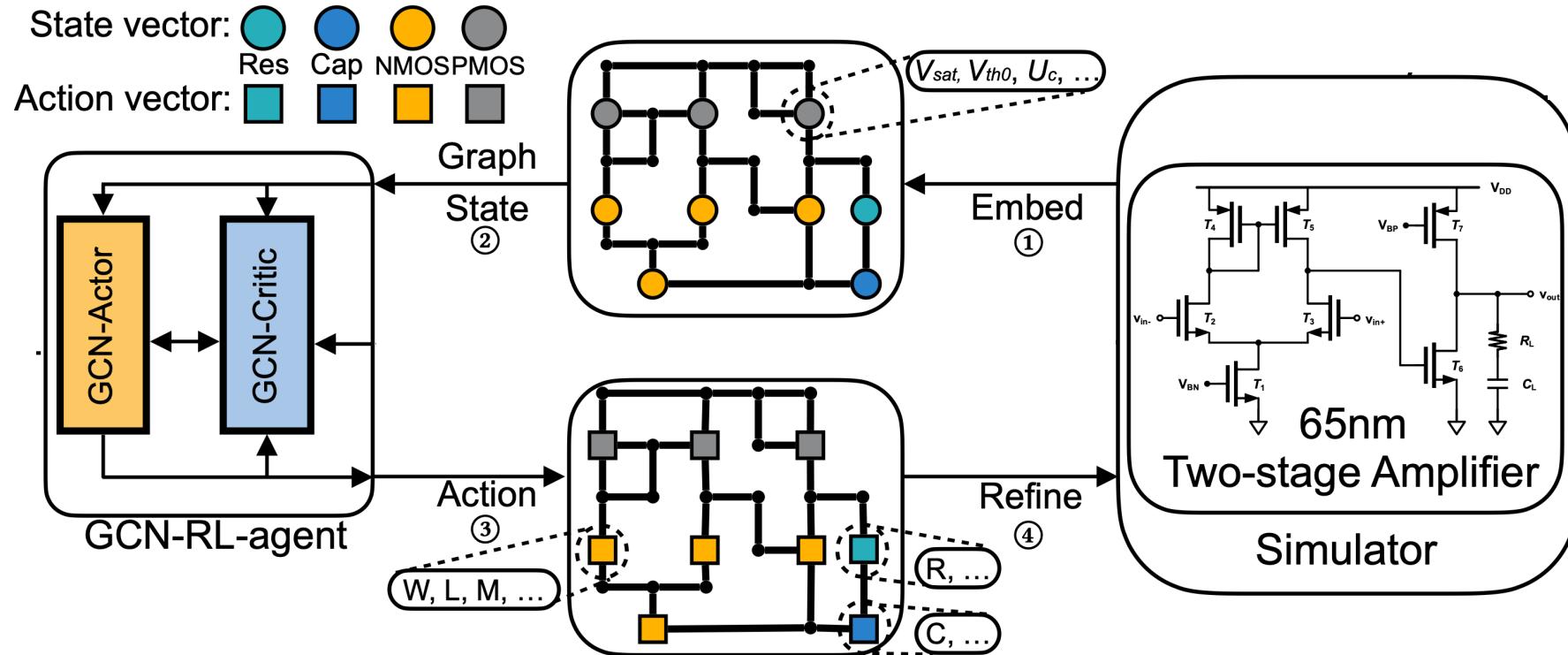
- Step 2: The environment generates a state vector for each component and passes them to the RL agent

# One Optimization Iteration Contains 6 Steps



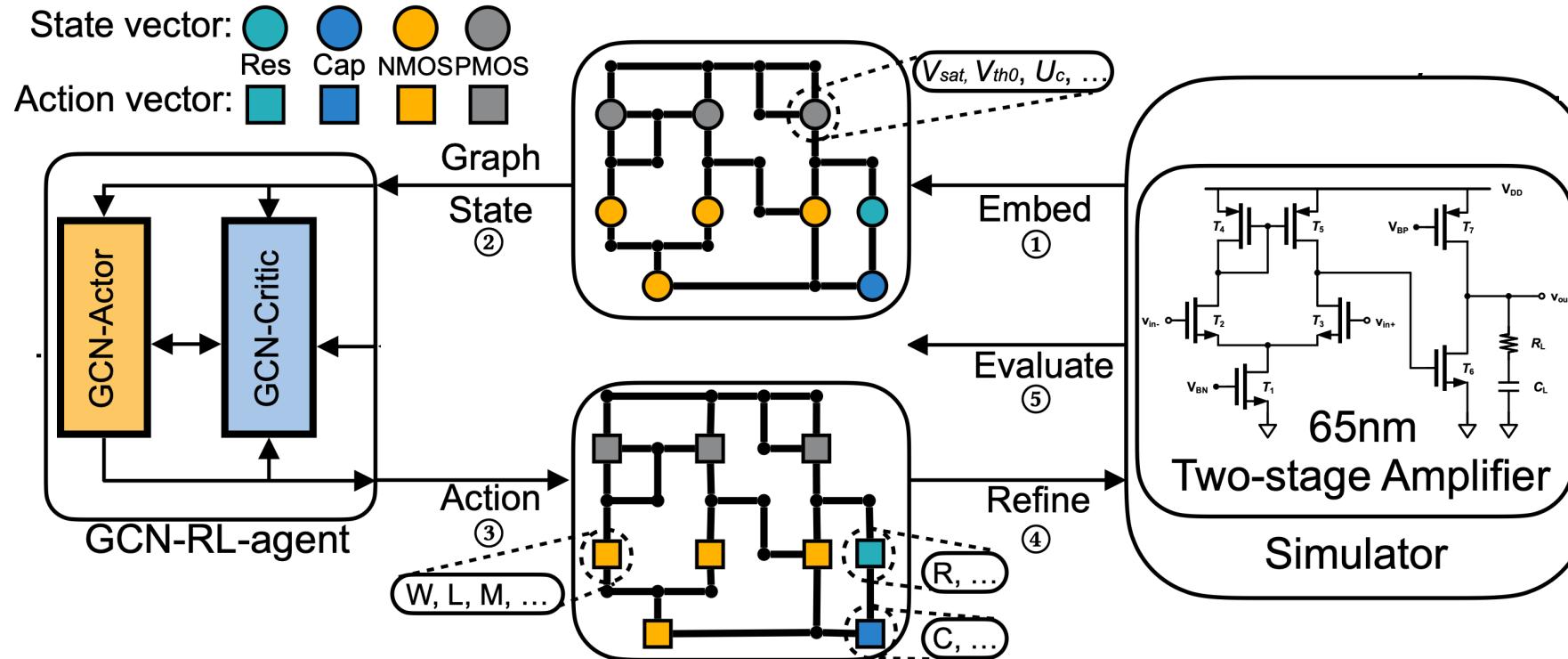
- Step 3: The RL agent processes the graph and states, and generates the sizes (W/L, R, C, etc.) for each node

# One Optimization Iteration Contains 6 Steps



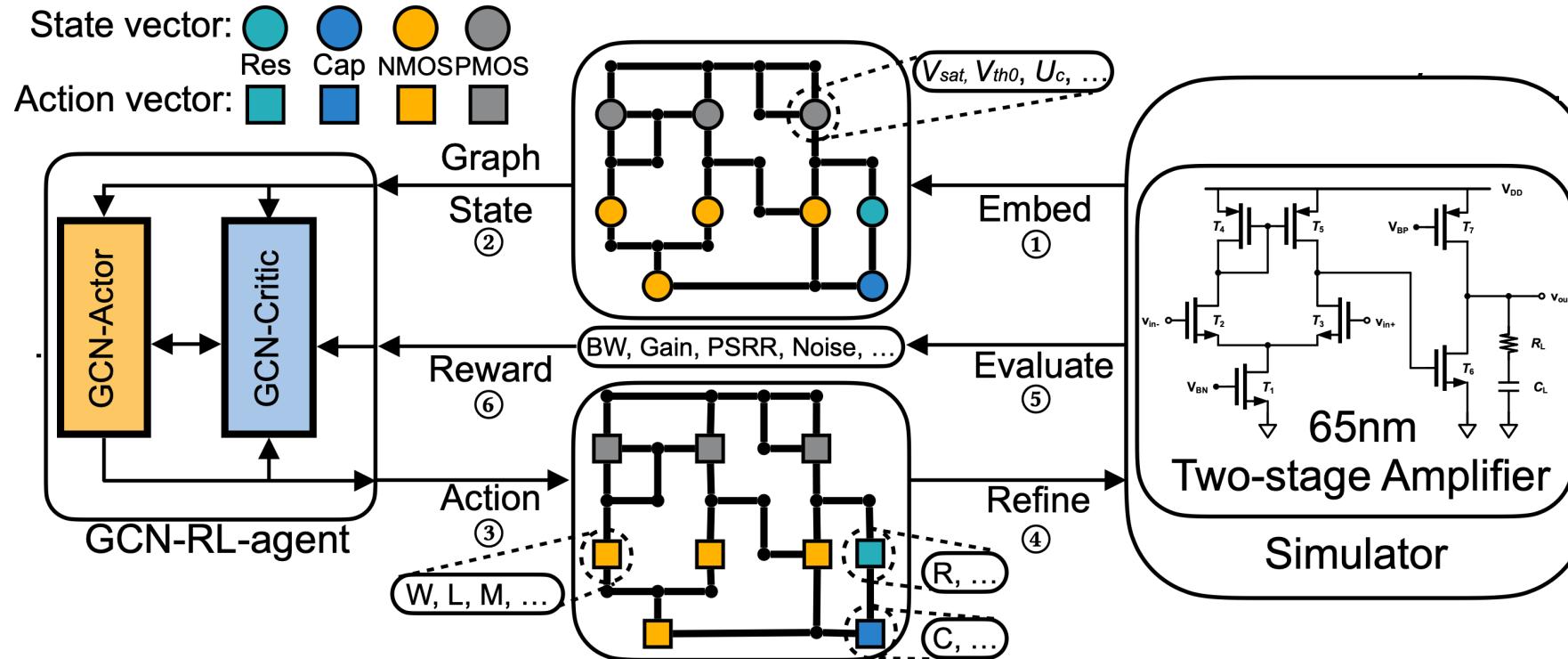
- Step 4: The environment refines the sizes to meet circuit requirements such as transistor matching

# One Optimization Iteration Contains 6 Steps



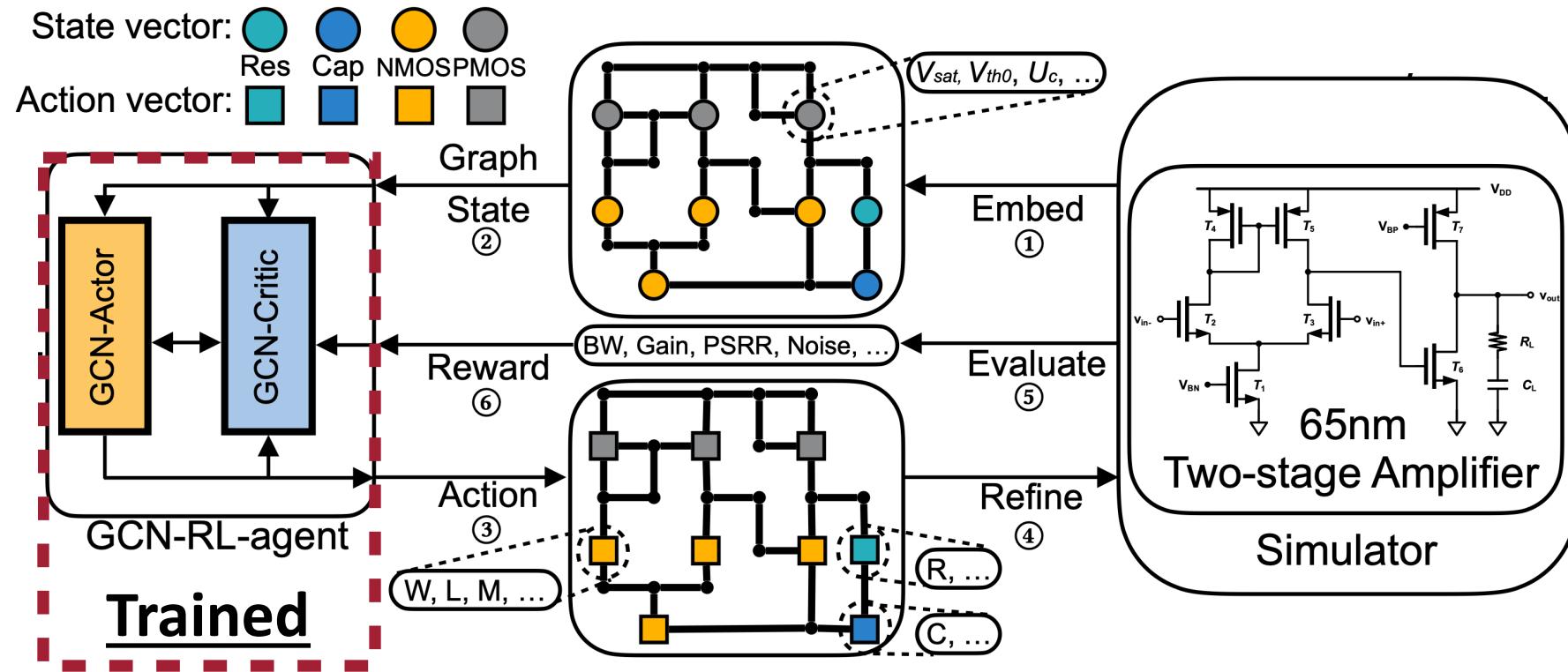
- Step 5: Simulator simulates the circuit with the generated sizes

# One Optimization Iteration Contains 6 Steps



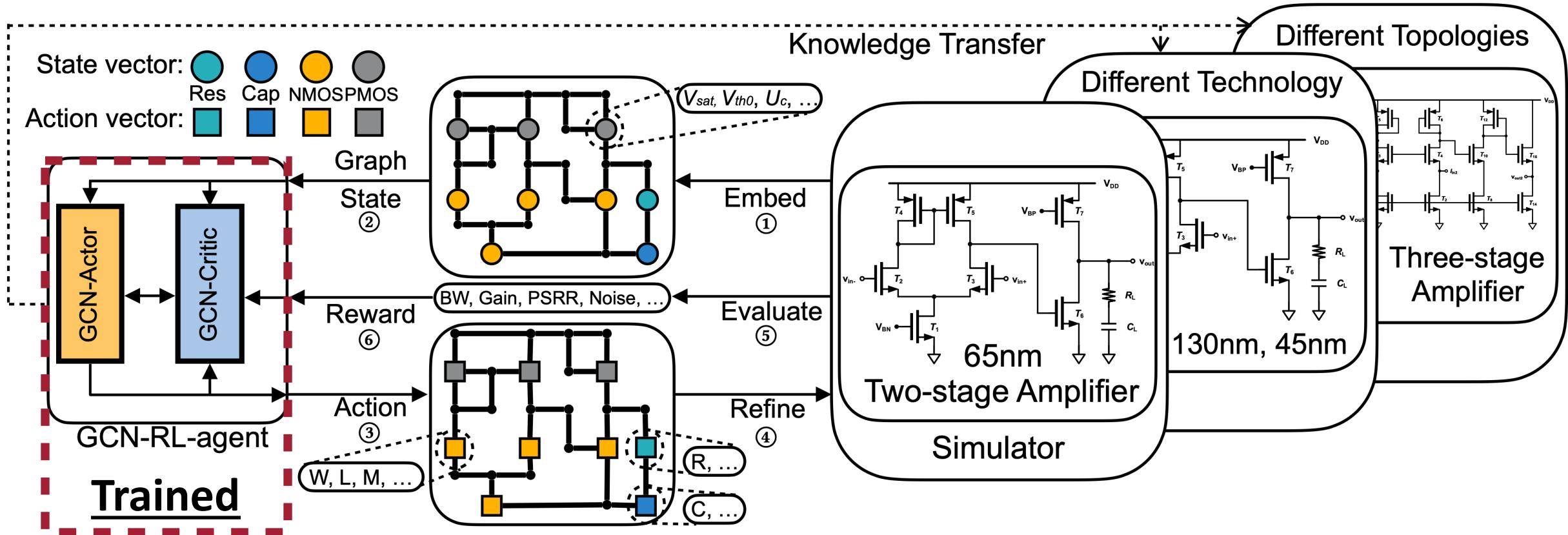
- Step 6: Compute a reward (FoM) value and feed it to RL agent to update policy

# Many Iterations Later



- After many iterations, we get the sizes with high FoM and a trained RL agent

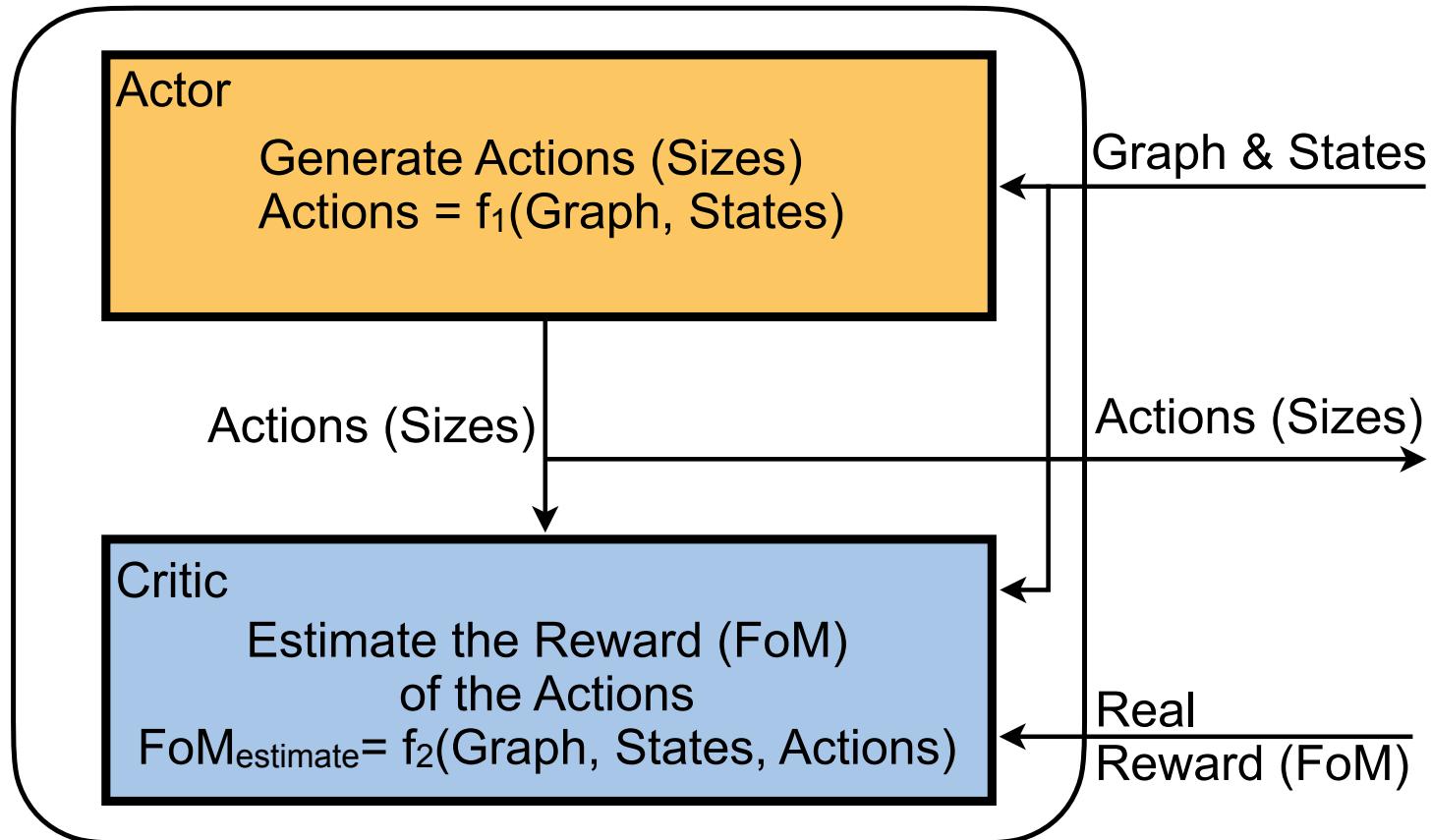
# Knowledge Transfer to Other Technology and Topology



- **Inherit** the trained RL agent to design circuits in other technology nodes and topologies

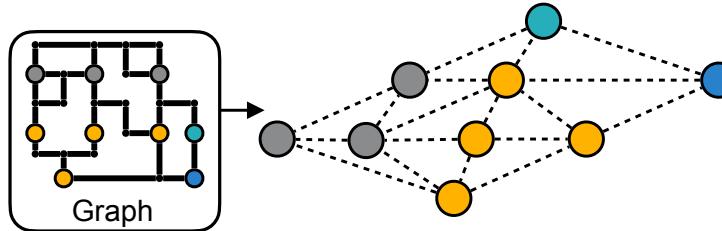


# GCN RL Agent: Actor-Critic Algorithm



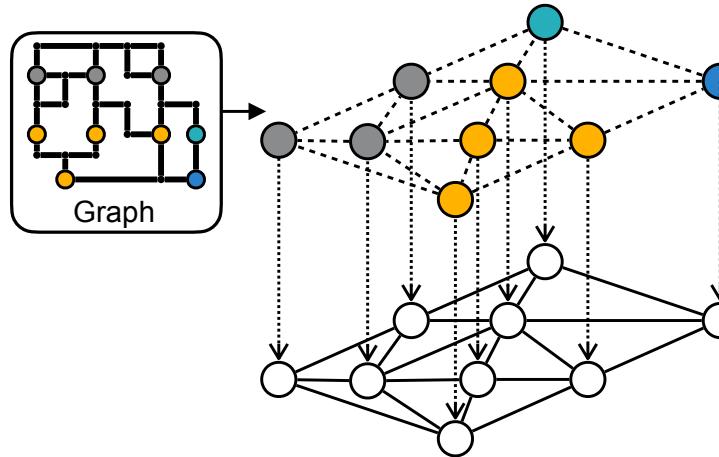
- Actor: Generates actions (sizes)
- Critic: Estimates reward (FoM) and **guides** the actor to generate actions with **high** reward

# GCN RL Agent: GCN Model



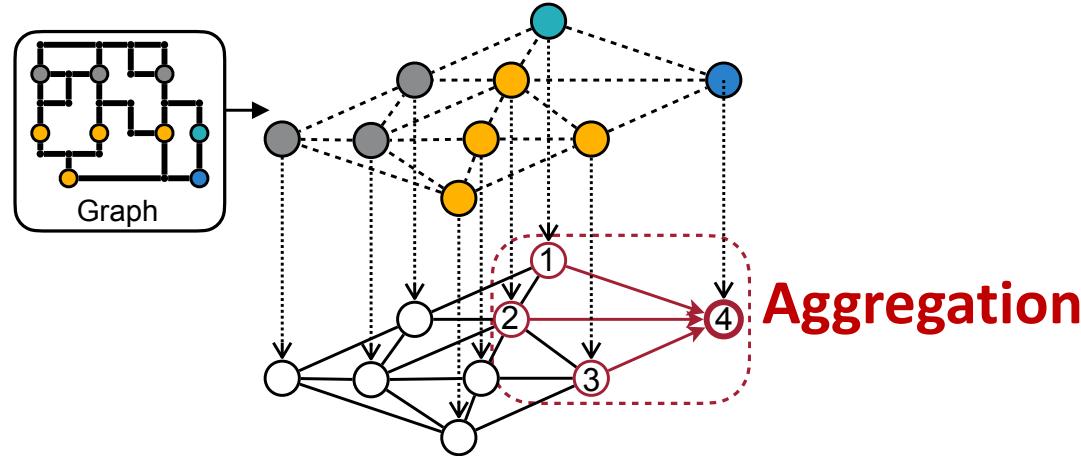
- Actor Input: graph and states on each node

# GCN RL Agent: GCN Model



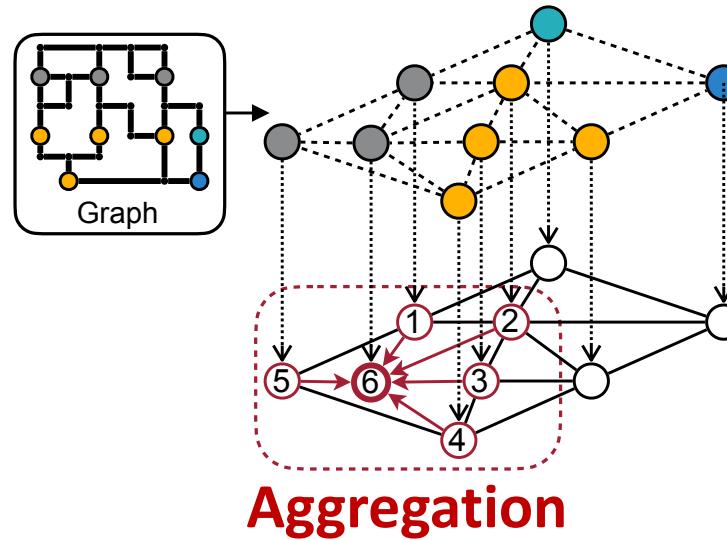
- First GCN layer: multiply input features with the same weight matrix for each node to get hidden features

# GCN RL Agent: GCN Model



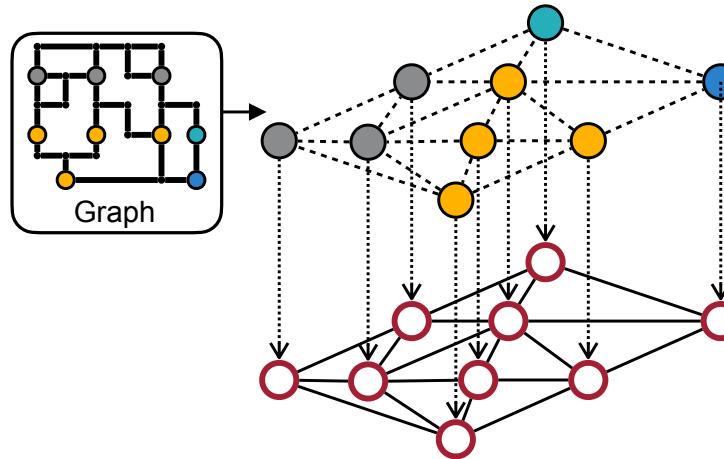
- First GCN layer: for each node, accumulate hidden features with neighbors to get aggregated features

# GCN RL Agent: GCN Model



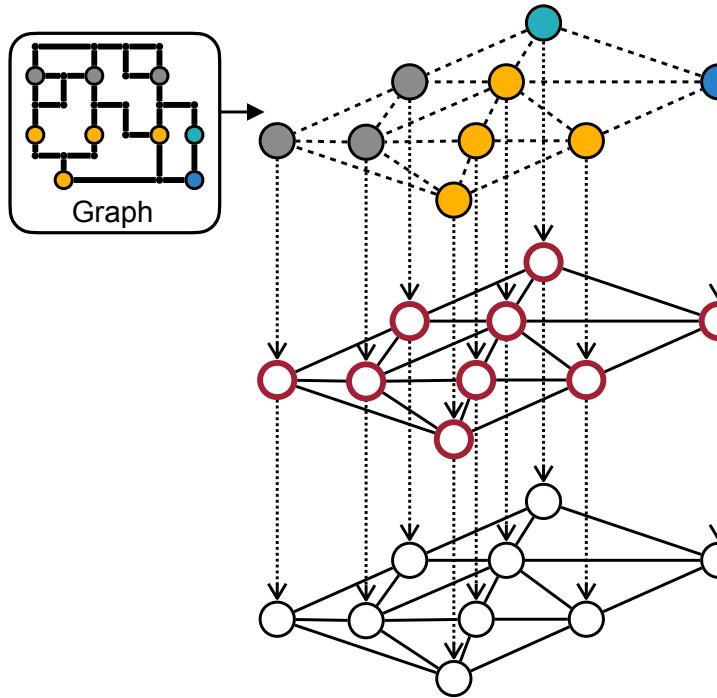
- First GCN layer: for each node, accumulate hidden features with neighbors to get aggregated features

# GCN RL Agent: GCN Model



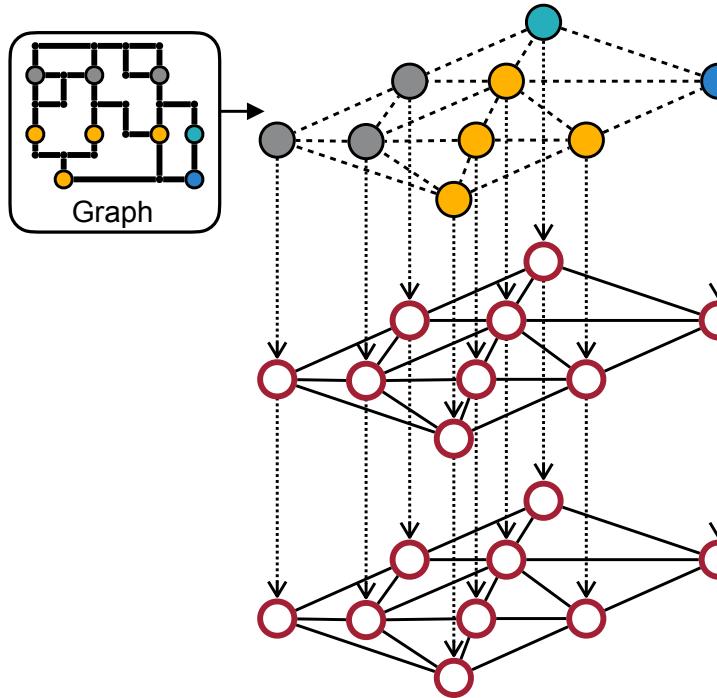
- First GCN layer: for each node, accumulate hidden features with neighbors to get aggregated features

# GCN RL Agent: GCN Model



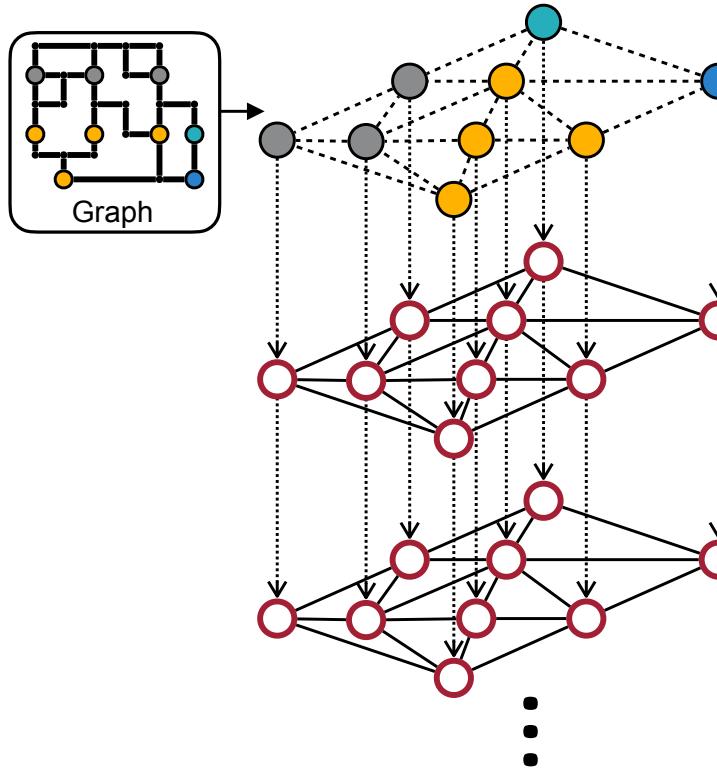
- Stack multiple GCN layers, each has one FC and one aggregation operation

# GCN RL Agent: GCN Model



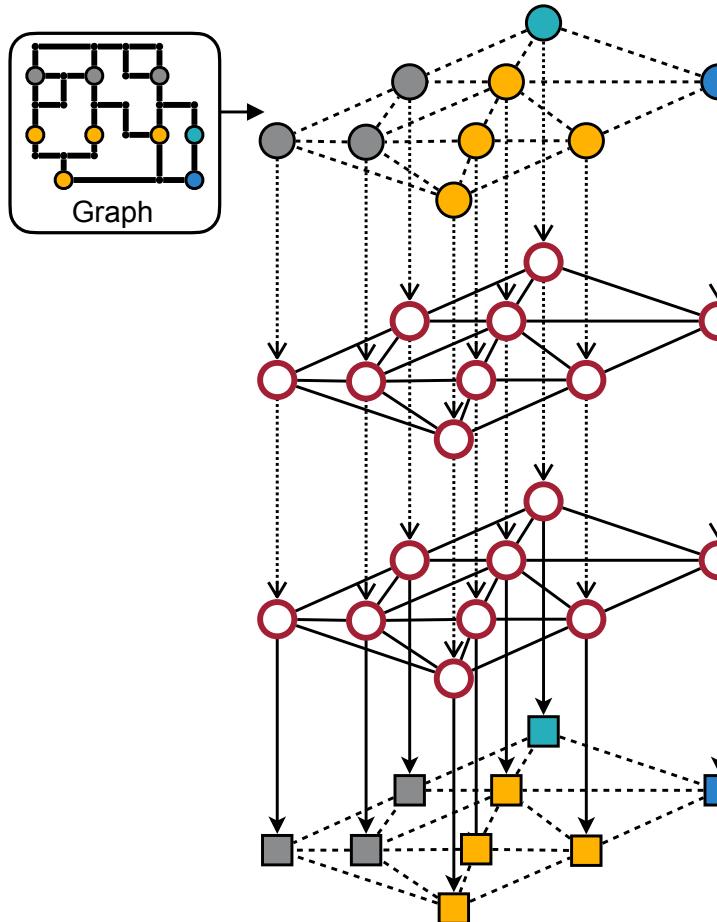
- Stack multiple GCN layers, each has one FC and one aggregation operation

# GCN RL Agent: GCN Model



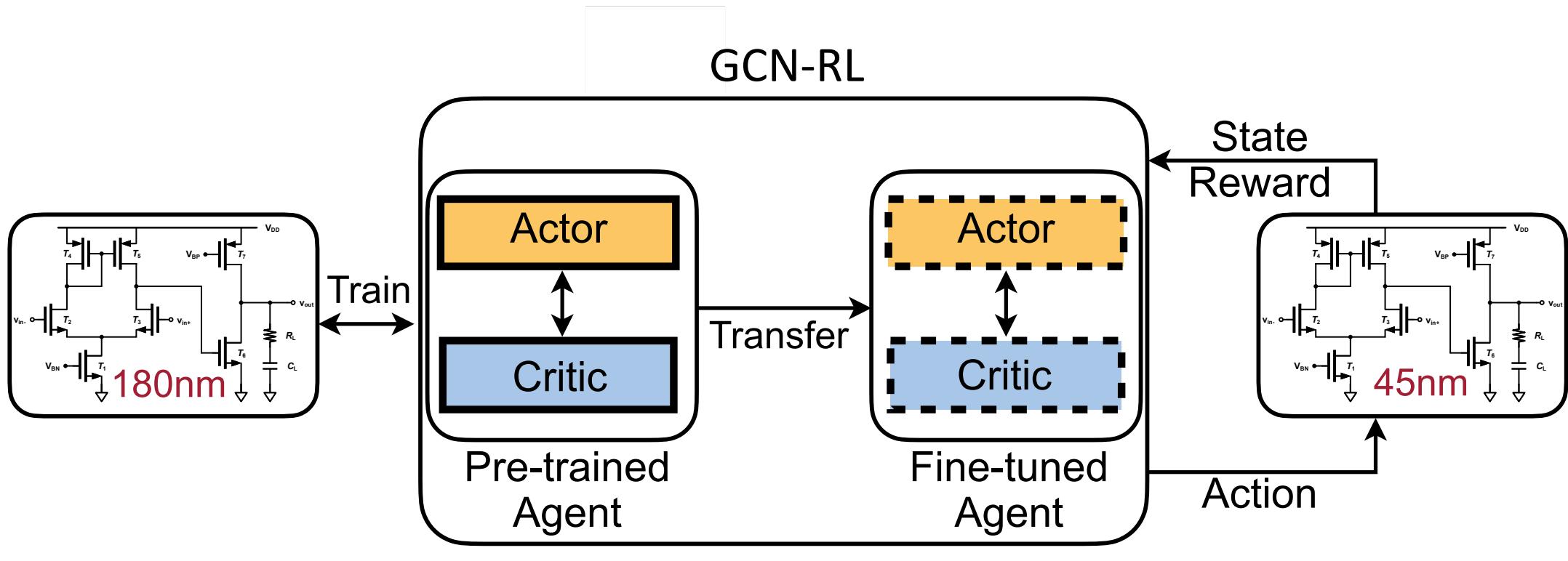
- After several GCN layers, each node has a **large** receptive field

# GCN RL Agent: GCN Model



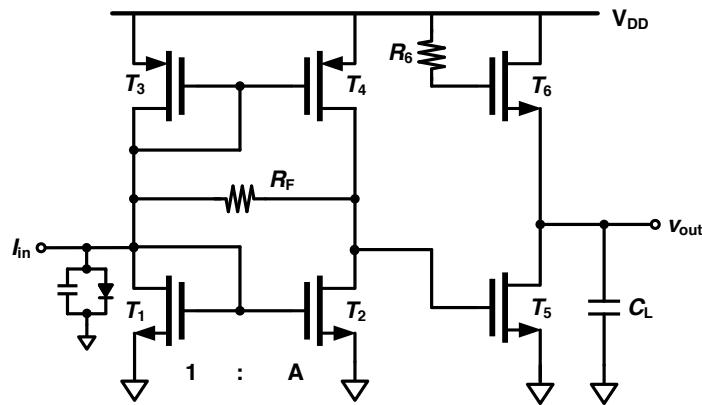
- After the last GCN layer: output actions (sizes) with a unique FC for each node
- Critic is similar but the last layer outputs estimated FoM

# GCN-RL Enables Knowledge Transfer

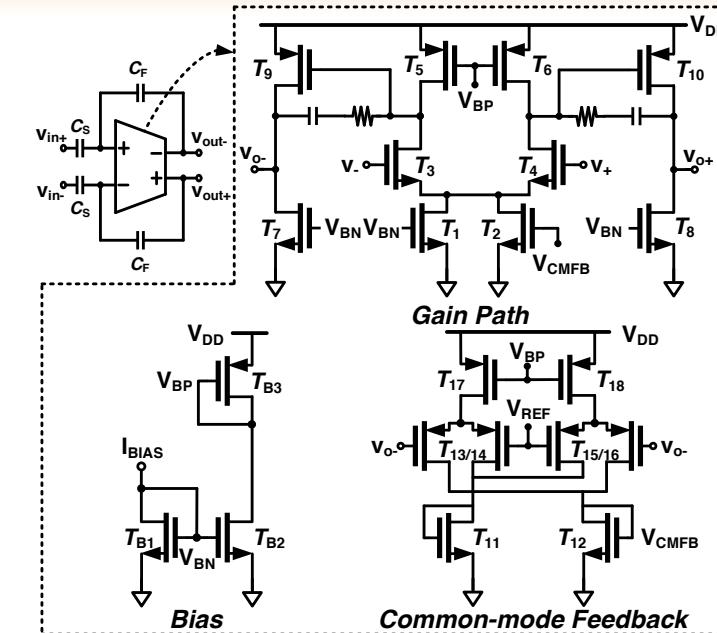


- Bayesian Optimization & Evolutionary Search **cannot** transfer knowledge

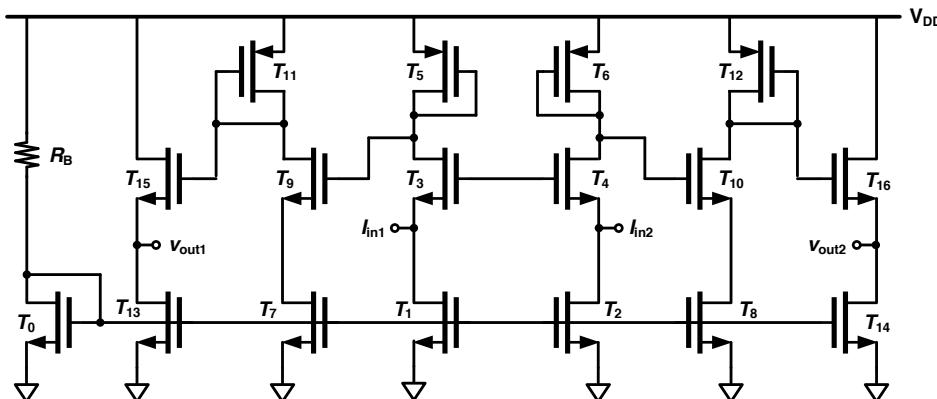
# Evaluations



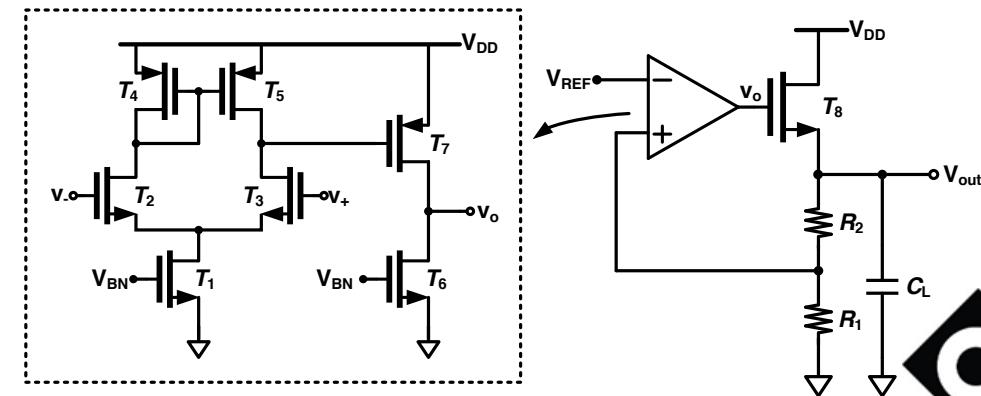
Two-Stage TIA



Two-Stage Voltage Amp



Three-Stage TIA



Low-Dropout Regulator



# Experiment Setups

- Figure of Merits (FoM):
  - Weighted sum of multiple normalized performance metrics such as bandwidth, power, and gain
- Baselines:
  - Human Expert Design
  - Random Search
  - Evolutionary Search (ES)
  - Bayesian Optimization (BO)
  - MACE (multi-objective version of BO)
  - NG-RL (Non-graph RL, no GCN layer aggregation)



# GCN-RL Achieves the Highest FoMs

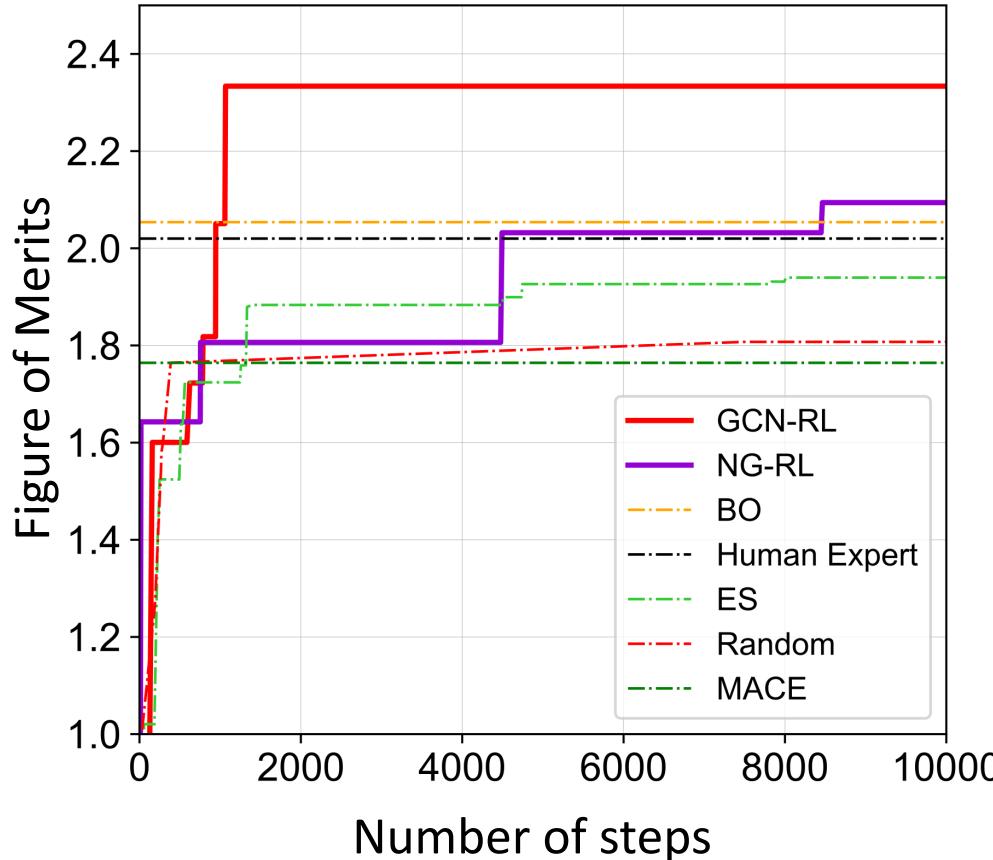
Method	Two-TIA	Two-Volt	Three-TIA	LDO
Human	2.32	2.02	1.15	0.61
Random	$2.46 \pm 0.02$	$1.74 \pm 0.06$	$0.74 \pm 0.03$	$0.27 \pm 0.03$
ES	$2.66 \pm 0.03$	$1.91 \pm 0.02$	$1.30 \pm 0.03$	$0.40 \pm 0.07$
BO	$2.48 \pm 0.03$	$1.85 \pm 0.19$	$1.24 \pm 0.14$	$0.45 \pm 0.05$
MACE	$2.54 \pm 0.01$	$1.70 \pm 0.08$	$1.27 \pm 0.04$	$0.58 \pm 0.04$
NG-RL	$2.59 \pm 0.06$	$1.98 \pm 0.12$	$1.39 \pm 0.01$	$0.71 \pm 0.05$
<b>GCN-RL (Ours)</b>	<b><math>2.69 \pm 0.03</math></b>	<b><math>2.23 \pm 0.11</math></b>	<b><math>1.40 \pm 0.01</math></b>	<b><math>0.79 \pm 0.02</math></b>

- GCN-RL consistently achieves the **highest FoMs with small variances**

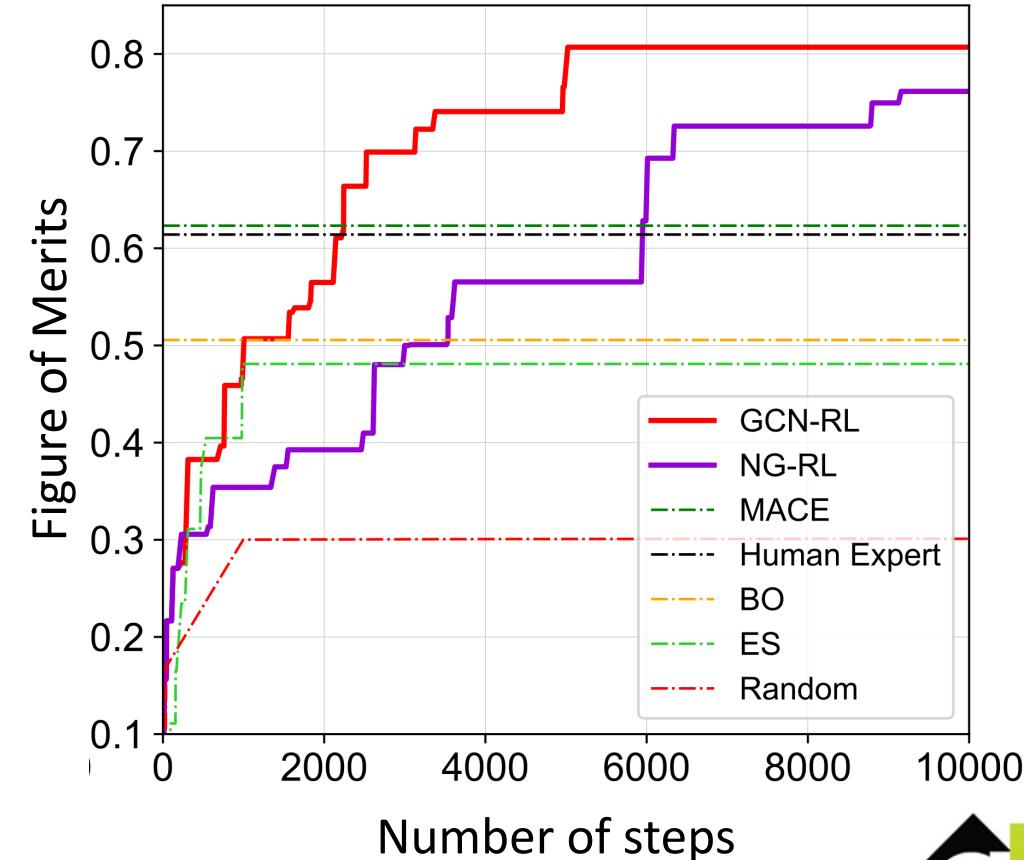


# Learning Curves

Two-Stage Voltage Amplifier

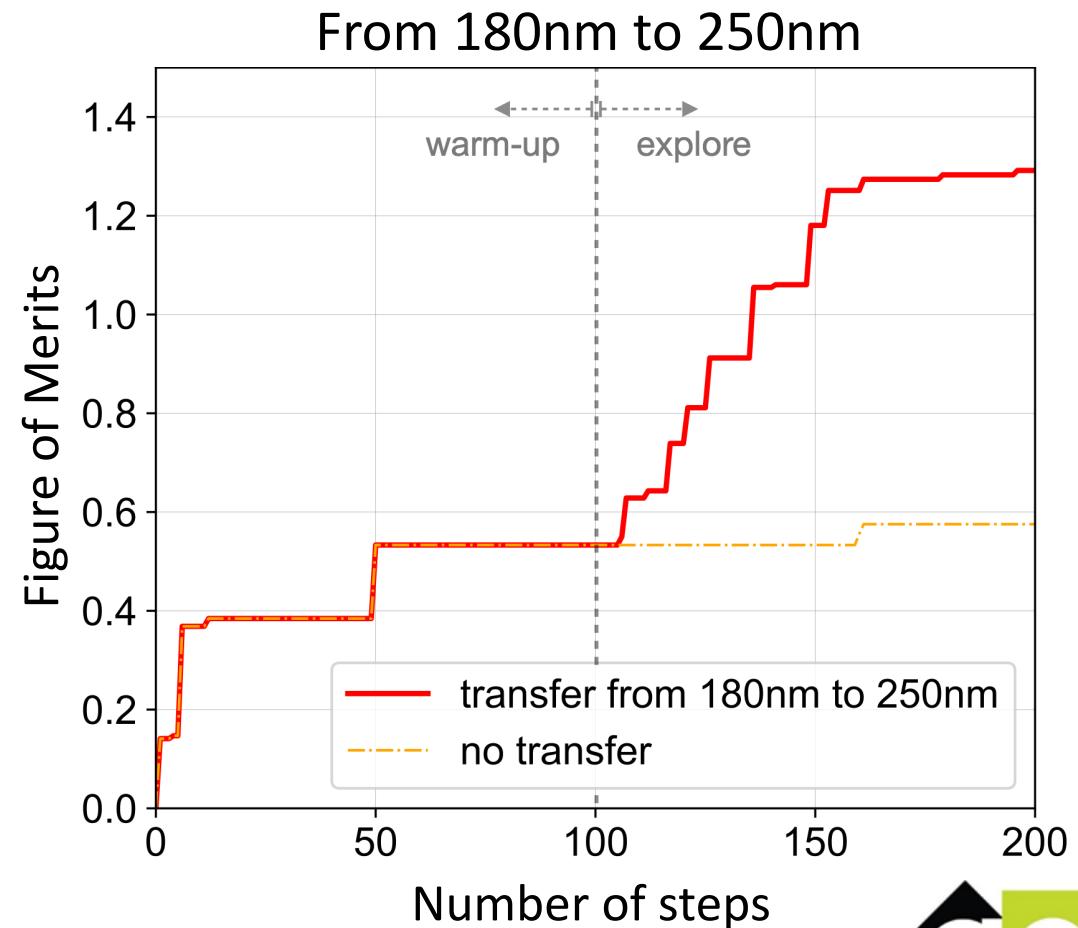
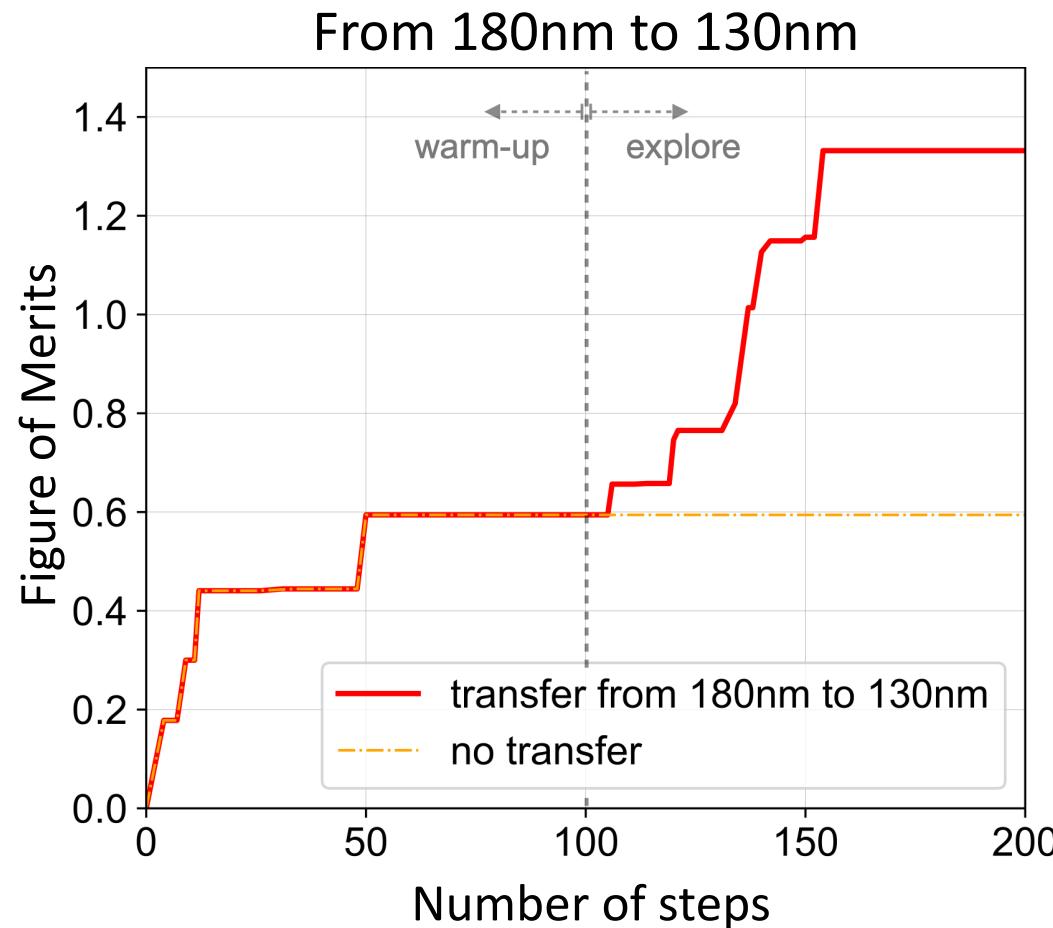


Low-Dropout Regulator



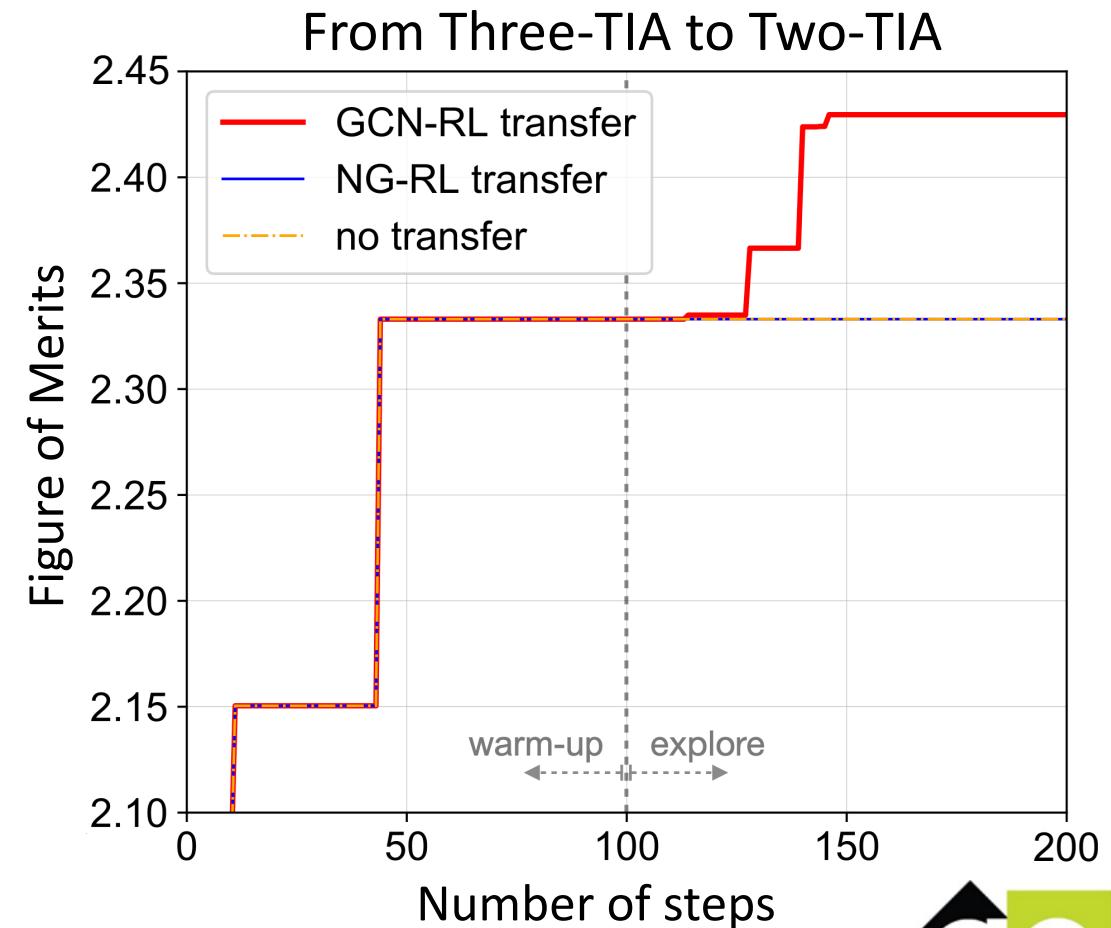
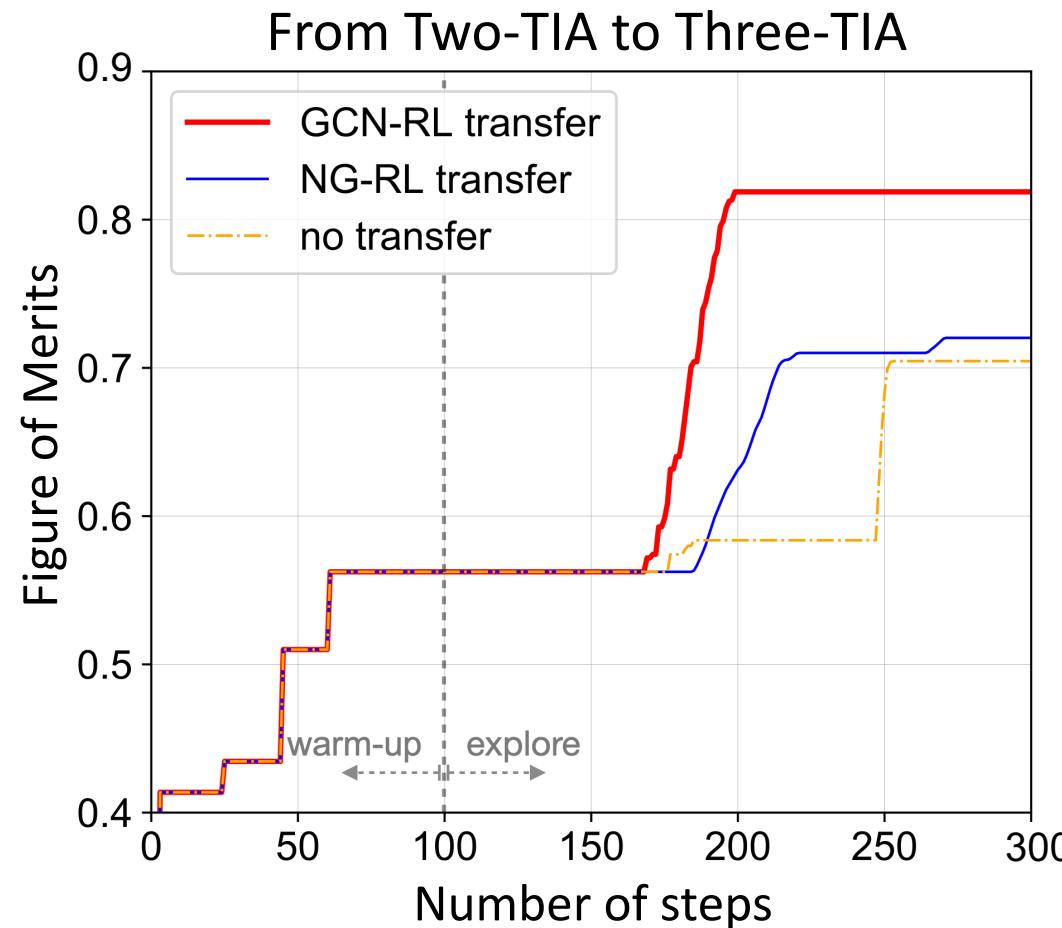
- GCN-RL has highest FoMs and **fastest convergence speed**

# Transfer between Tech Nodes on Three-Stage TIA



- GCN-RL achieves much higher FoM after 200 steps

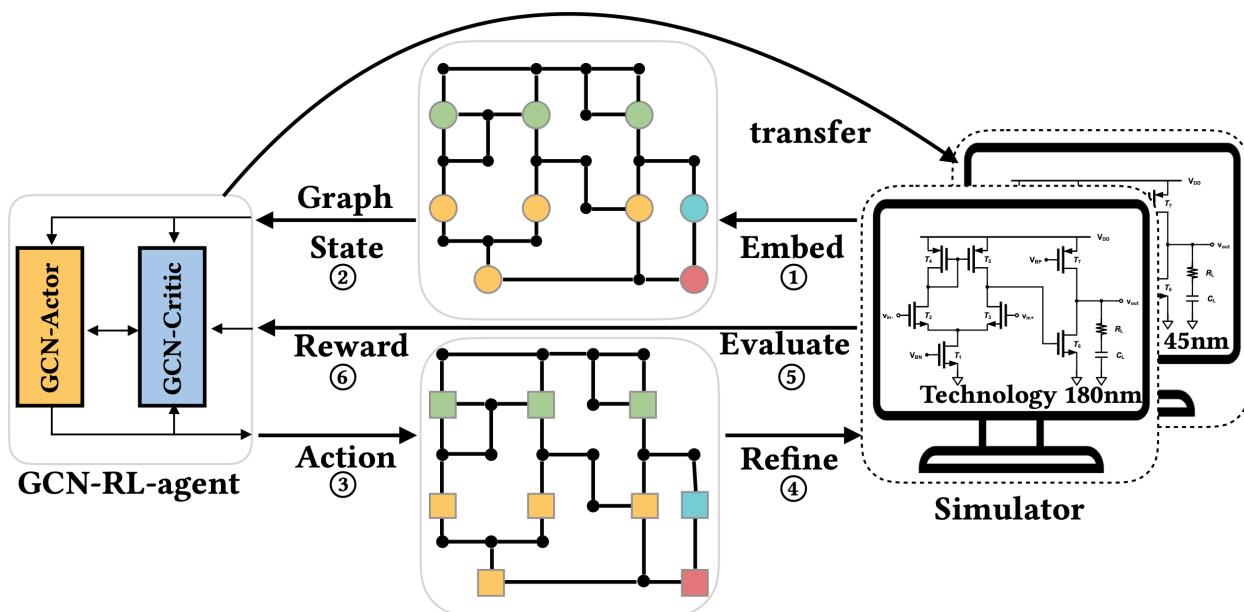
# Transfer between Two-Stage and Three-Stage TIA



- GCN-RL with transfer achieves highest FoM
- **Graph information is critical for knowledge transfer**

# GCN-RL Circuit Designer

- AI for EDA achieves **high** performance with **less** human labor
- GCN-RL consistently achieves highest FoMs
- GCN-RL has transfer learning ability
  - Between technology nodes
  - Between topologies



<https://gcnrl.mit.edu>

DAC 2020  
Area: Design, Machine Learning/AI  
Session: Learning to Yield  
Fri. July 24, 2:00pm to 3:00pm



# Questions

