MacroRank: Ranking Macro Placement Solutions Leveraging Translation Equivariancy

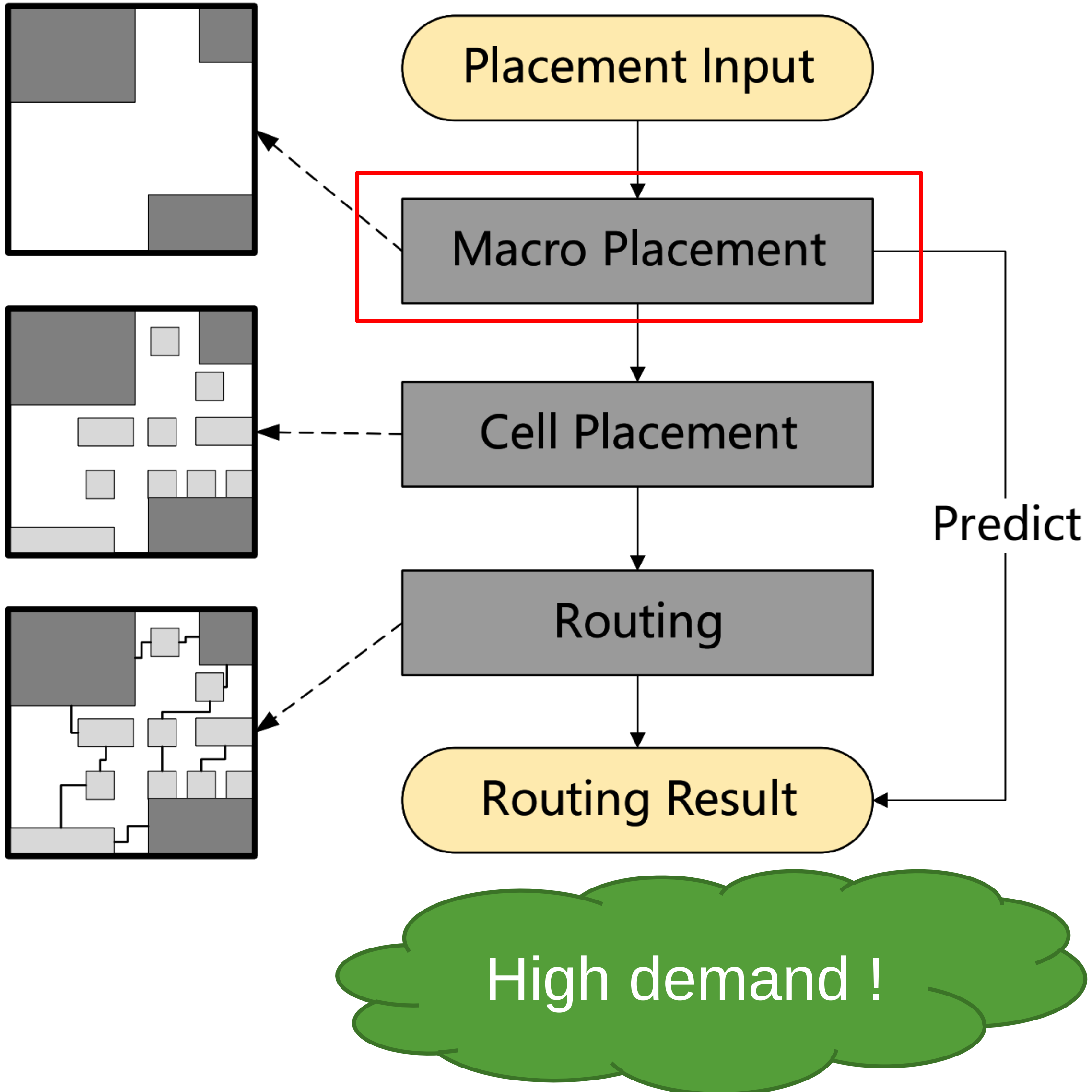
Yifan Chen, Jing Mai, Xiaohan Gao, Muhan Zhang, Yibo Lin

Peking University, Beijing, China
chenyifan2019@pku.edu.cn

Introduction: Placement & Routing Flow

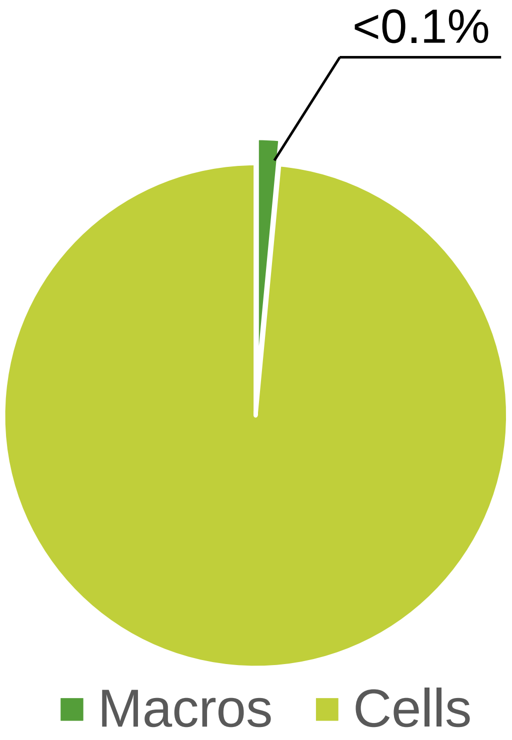
- Macro position: high impact
- Entire flow: time consuming

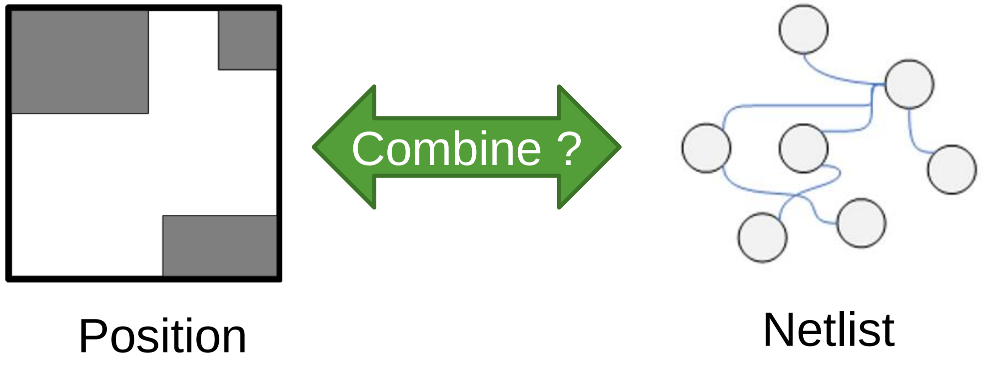
- Early prediction of routing performance at the macro placement stage



Introduction: Challenges

- Only know the position of macros
- How to combine geometry and interconnection information





Introduction: Related Works

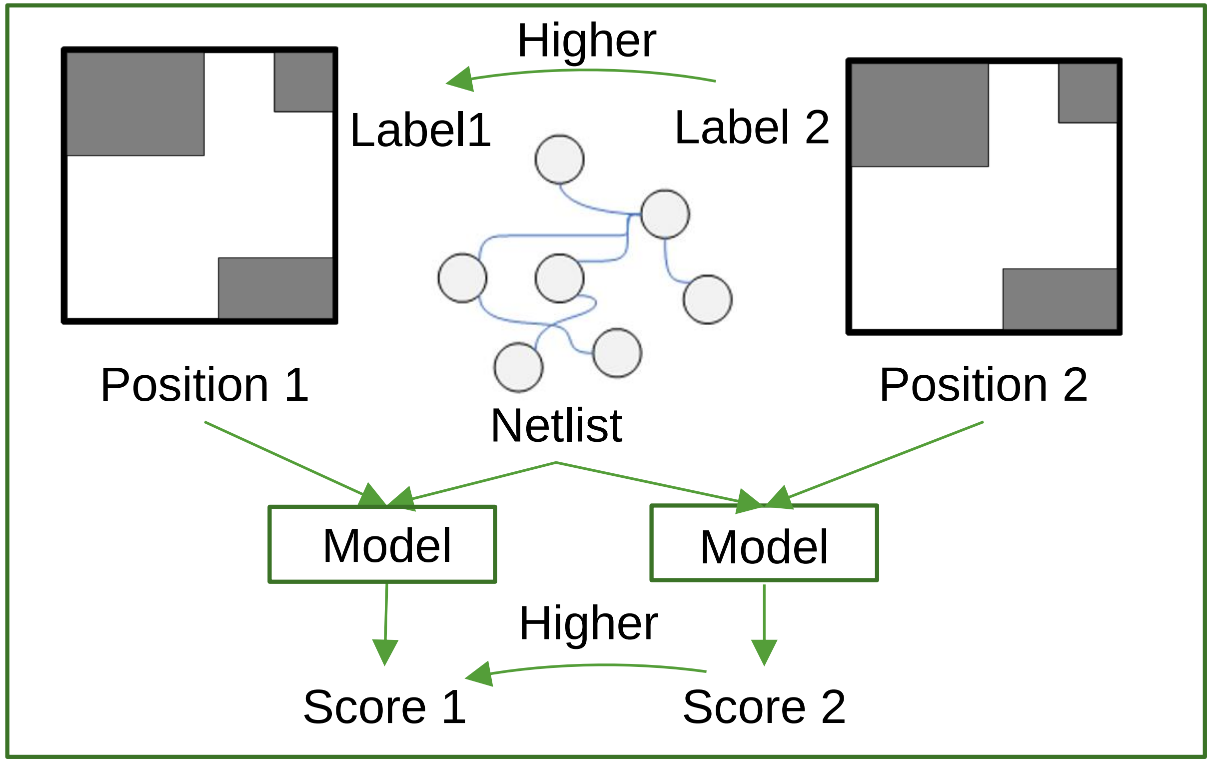
	Model	Geometry Info	Interconnection Info	Pin Info	Loss
Huang et al.	CNN	Image	×	√	MSLE
Mirhoseini et al.	GNN	Coordinate	√	×	MSE
Ours	GNN	Relative Coordinate	√	√	Ranking Loss

Introduction: Contribution

- MacroRank framework: rank macro placement solutions by routing quality
- EHNN: translation equivariant, extract both netlist and macro location information
- Learning to Rank (LTR): learn the relative order of macro placement solutions
- Better performance than the SOTA model
  - Improve the Kendall rank correlation coefficient by 49.5%.
  - Improve the average performance of top-30 prediction by 8.1%, 2.3%, and 10.6% on wirelength, vias, and shorts, respectively

Preliminary: Problem Formulation

- Input: Macro Position, Netlist
- Output: Score
- Target: Order-preserving



- Global Routing Metrics:
  - ICCAD2019 Contest
  - WL, #Vias, #Shorts
- Prediction Accuracy Metrics:
  - Mean Relative Error

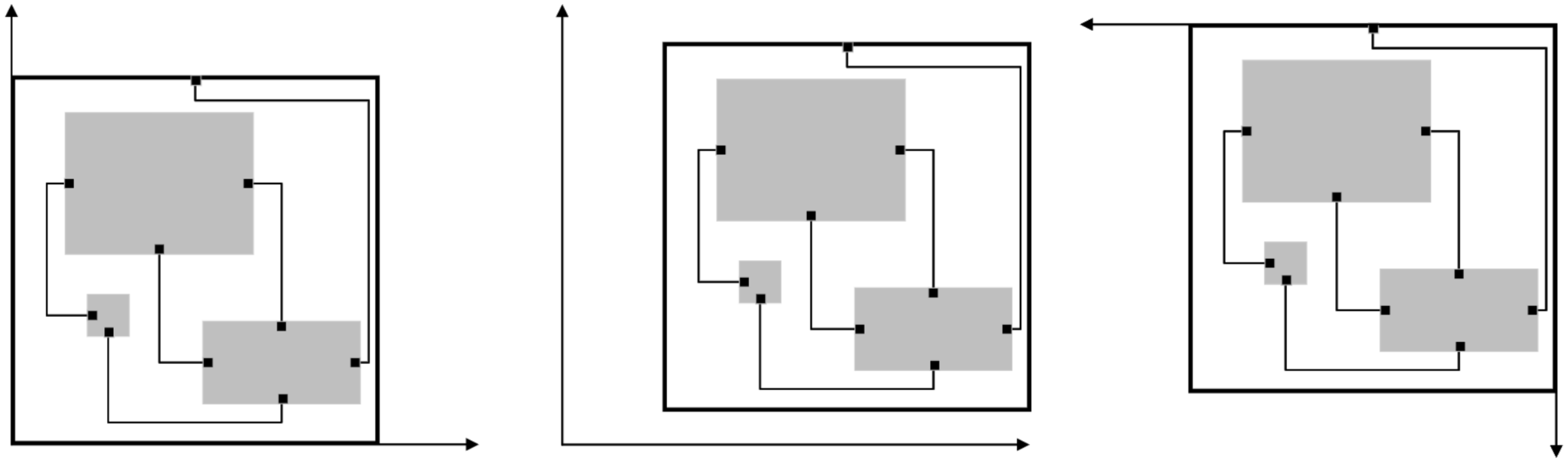
$$MRE = \left| \frac{y_{pred} - y_{label}}{y_{label}} \right|$$

  - Kendall's correlation coefficient

$$\tau = \frac{n_{concordant} - n_{discordant}}{\frac{1}{2} n(n-1)}$$

Preliminary: Equivariance

- Rigid body transformation



(a) Origin (b) Translation (c) Rotation

- Will not affect the optimal solutions of placement and routing
  - E(2)-equivariance
- But in practice, only suboptimal solutions can be found

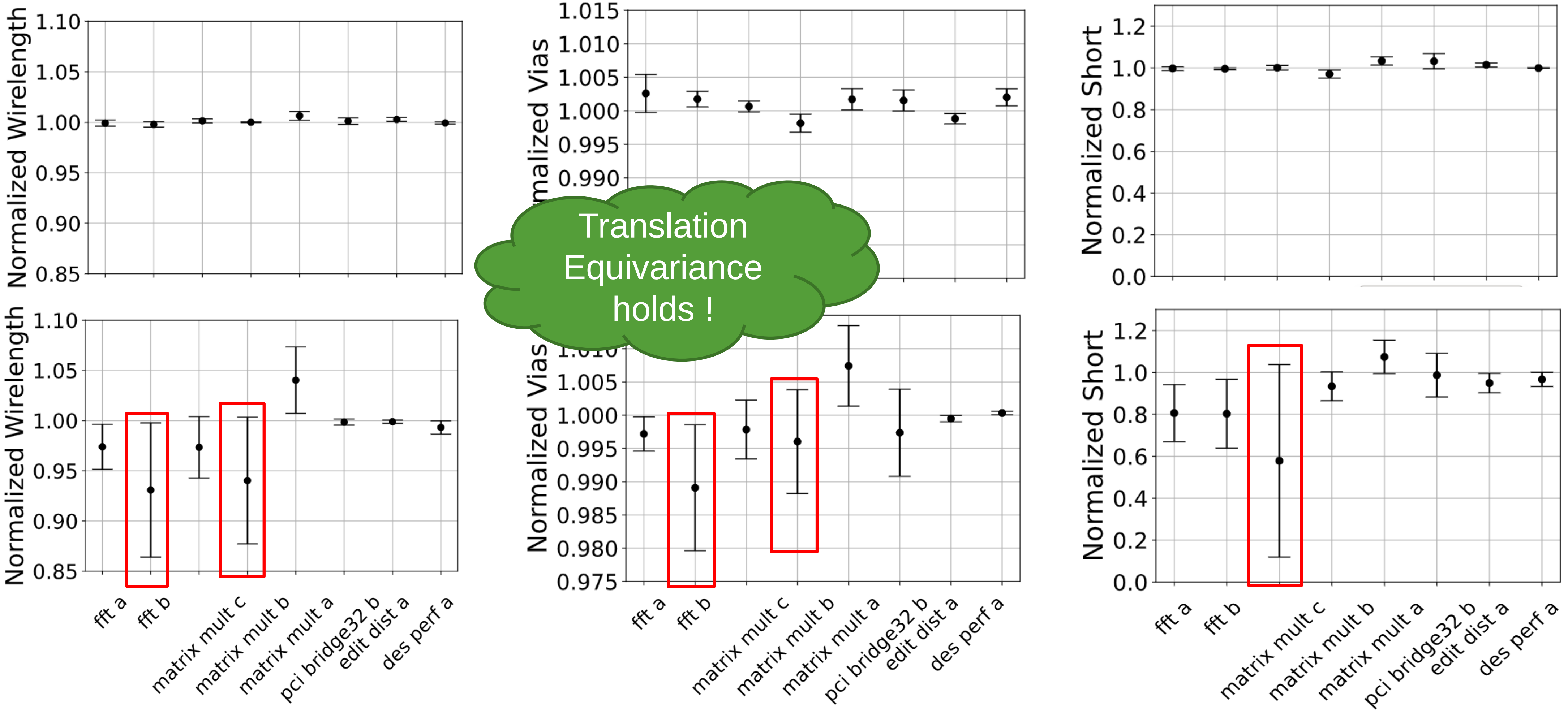
Do equivariance really holds ?

Preliminary: Equivariance

DREAMPlace + CU.GR

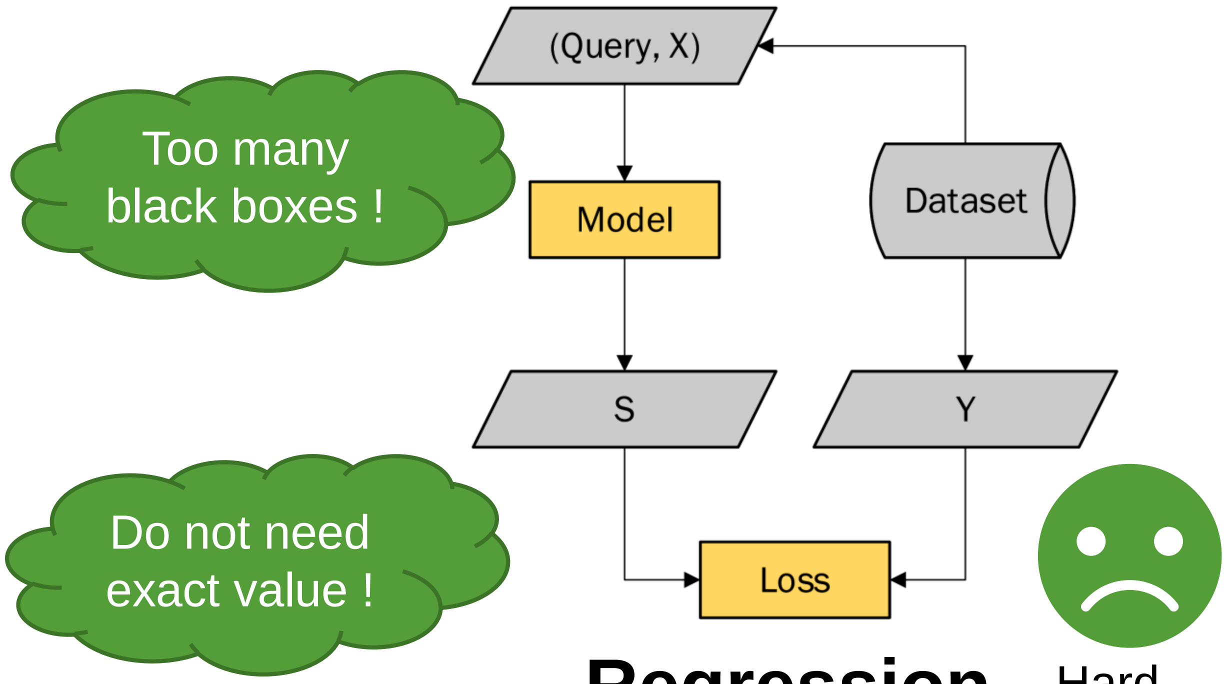
Translation

Rotation
&
Reflection



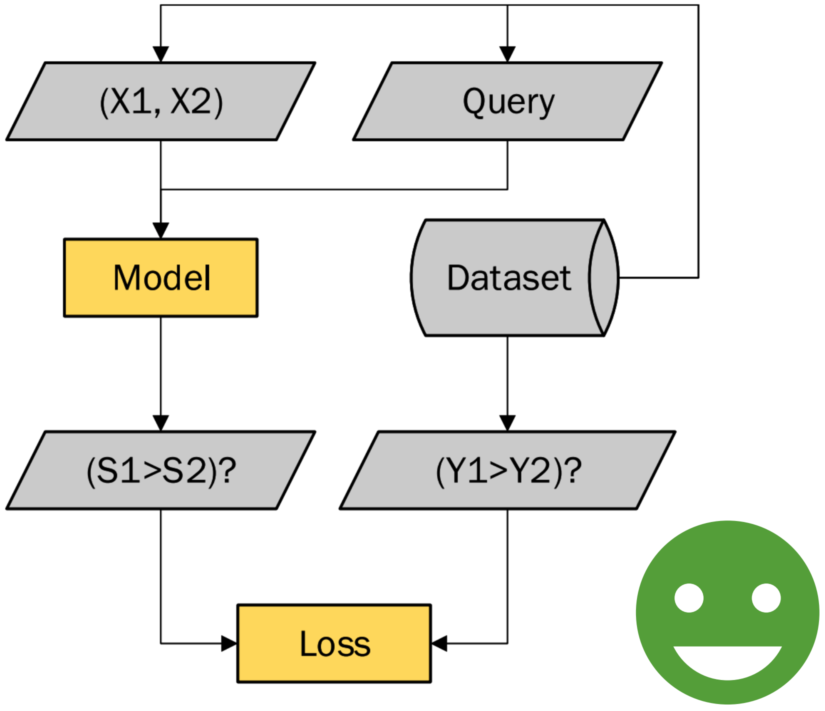
Preliminary: Learning to Rank (LTR)

- Pairwise LTR: Given a pair of samples, predict which is better
  - Need a scoring function f, takes sample X as input and outputs a score s
  - If $s_i > s_j$, then X_i is better than X_j



Regression

Hard
Not needed

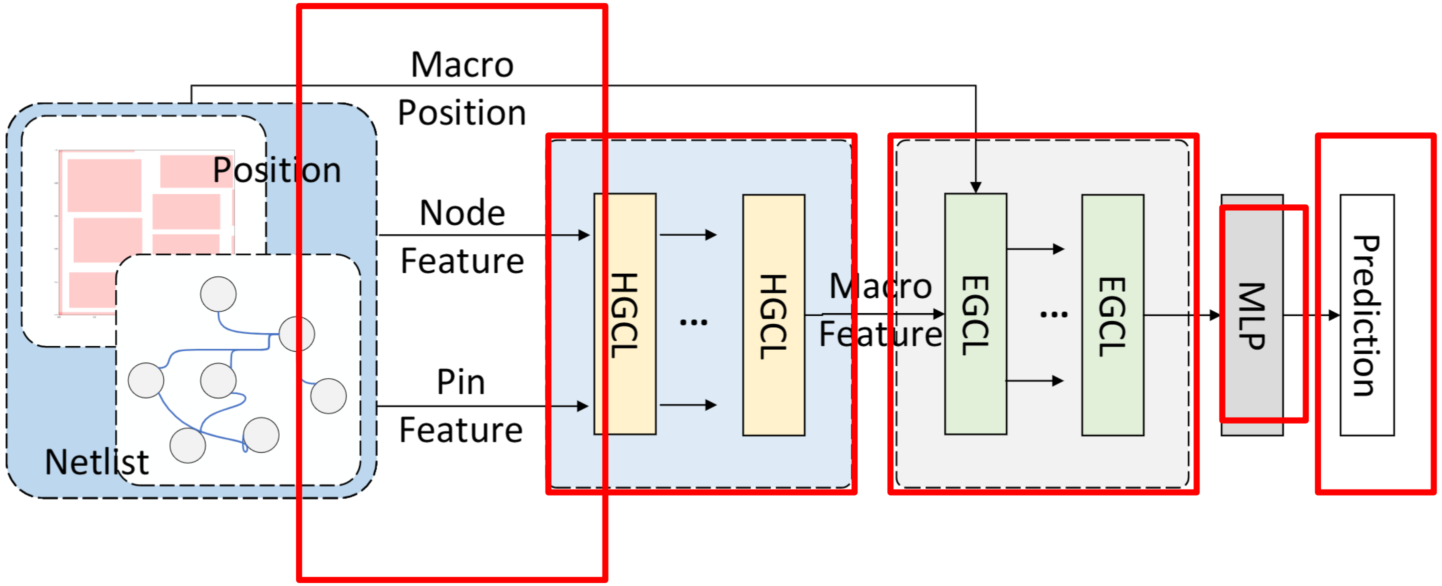


Pairwise LTR

Easier
What we want

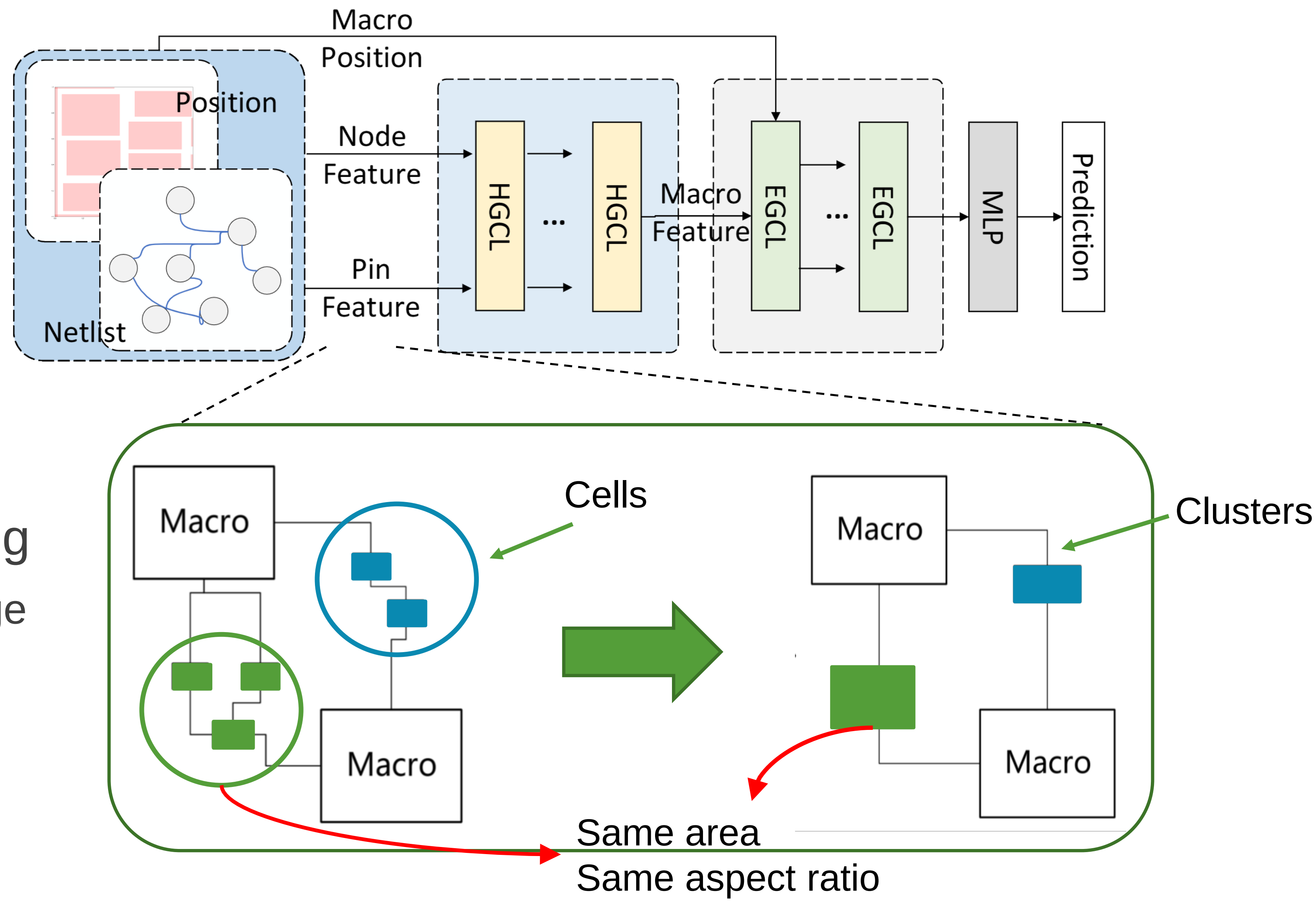
MacroRank: Architecture

- EHNN



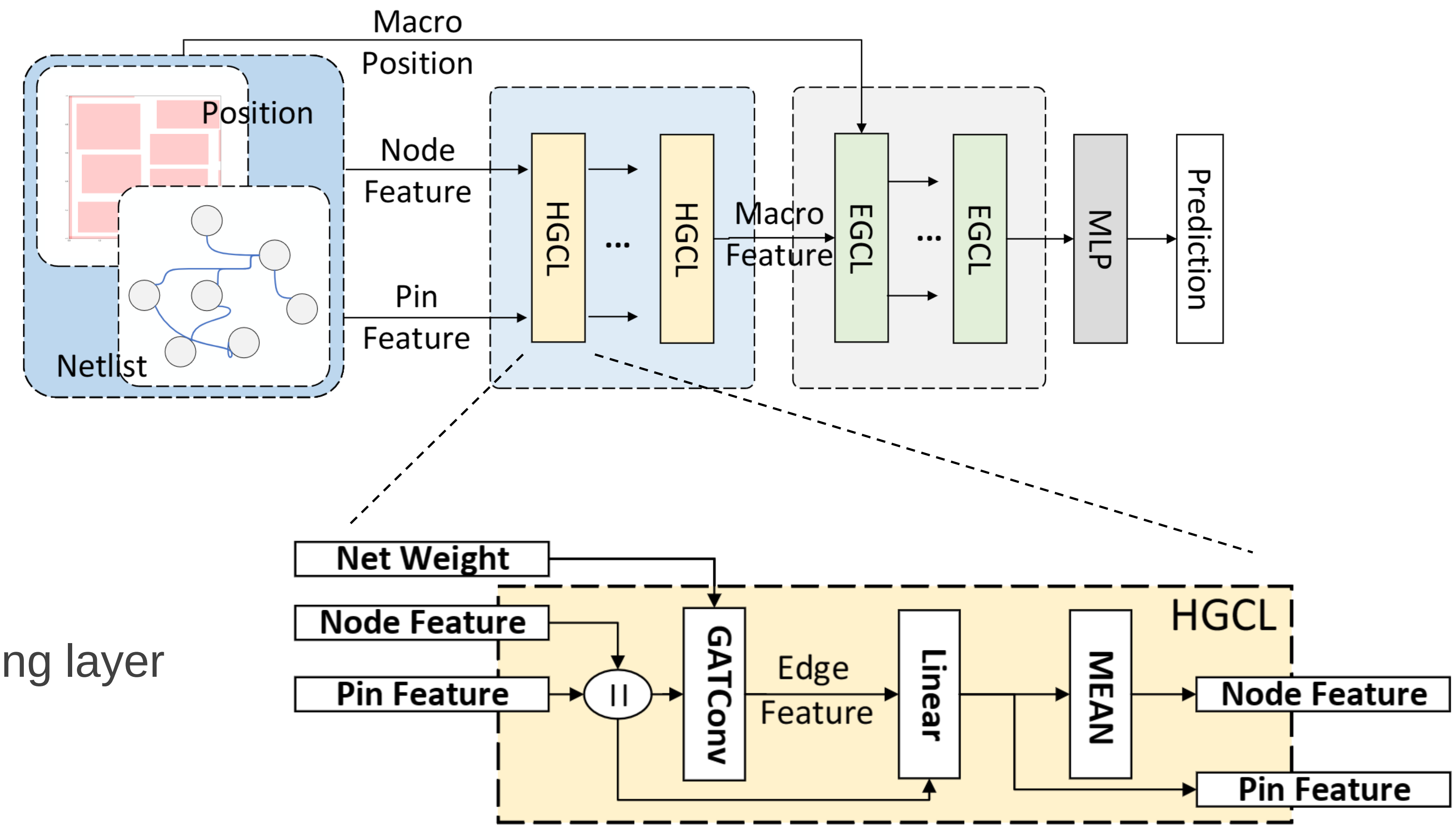
MacroRank: Clustering

- EHNN
- Cell Clustering
  - Netlist too large
  - Few macros
  - hMETIS



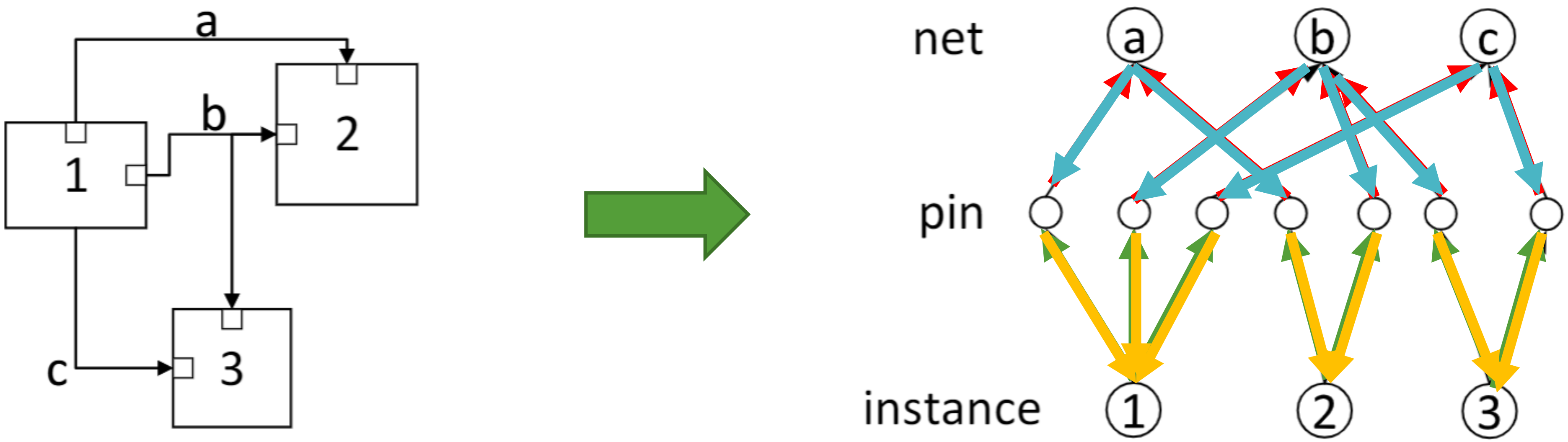
MacroRank: HGCL

- EHNN
- HGCL
  - Netlist encoding layer



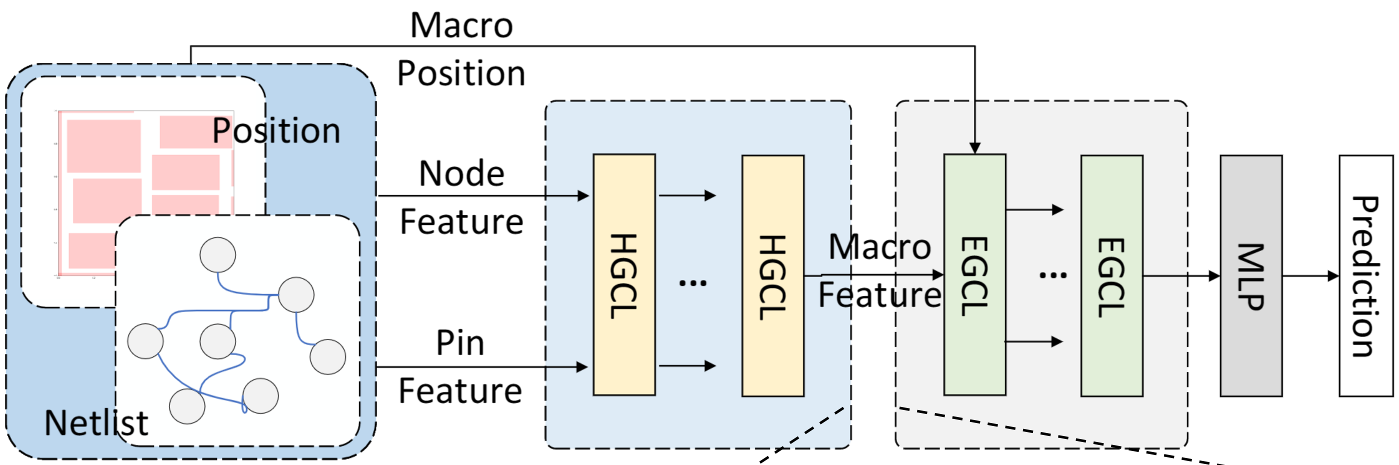
MacroRank: HGCL

- Modeling netlist as a tripartite graph.
- Two stage message passing:
  - Instance to pin (Concatenation), pin to net (GAT)
  - Net to pin (Linear), pin to instance (MEAN)

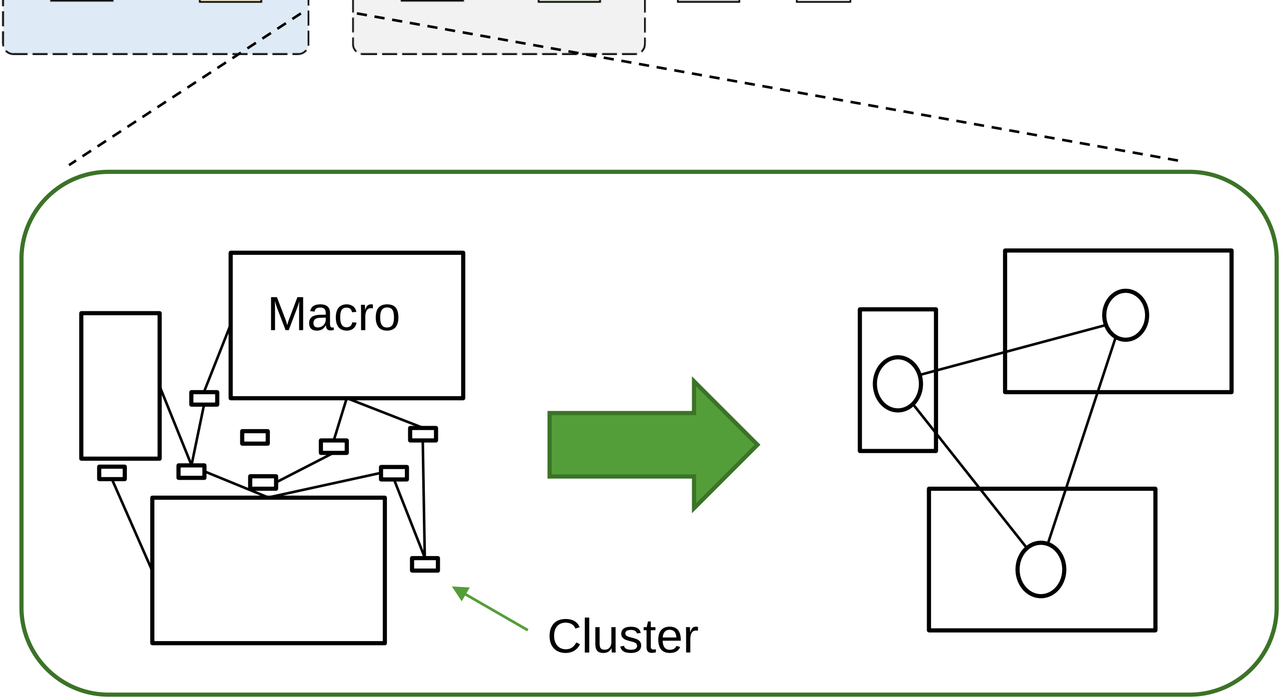


MacroRank: EGCL

- EHNN

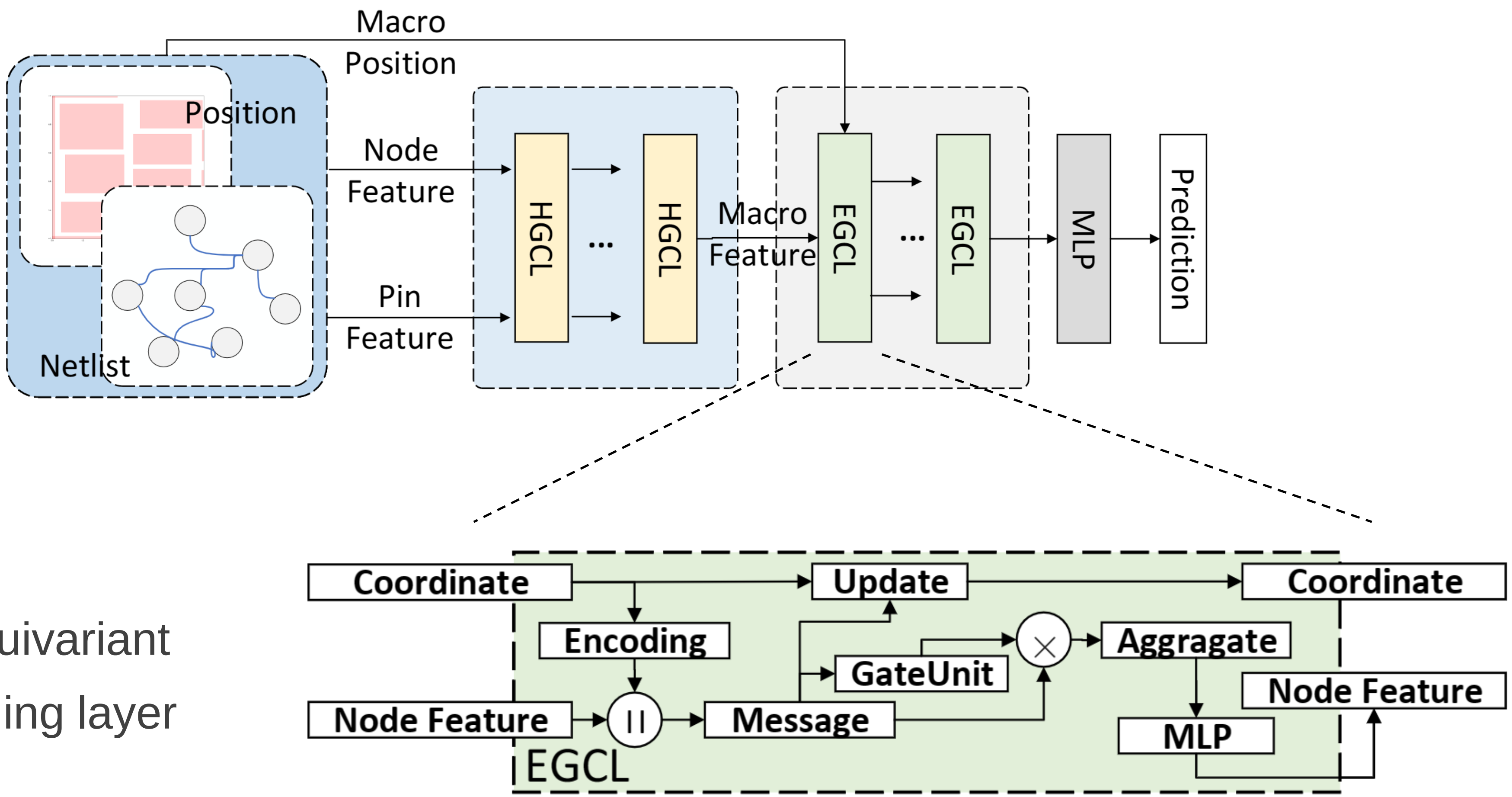


- From netlist to macro only graph
  - Remove all clusters
  - Connect to K nearest neighbors.



MacroRank: EGCL

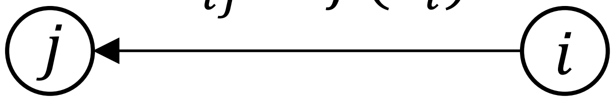
- EHNN
- EGCL
  - Translation Equivariant
  - Position encoding layer



MacroRank: EGCL

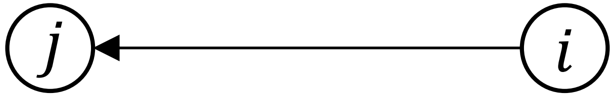
- Translation equivariant neighborhood message passing
  - Directly pass x_i, no equivariance.
  - Pass $\mathrm{d}(x_i - x_j)$, depends on encoding function $d(\cdot)$.

$$m_{ij} = f(\vec{x_i})$$



No equivariance

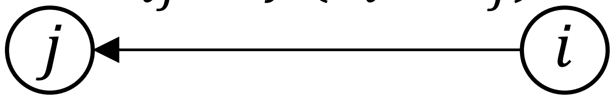
$$m_{ij} = f(d(\vec{x_i} - \vec{x_j}))$$



Depends on $d(\cdot)$.

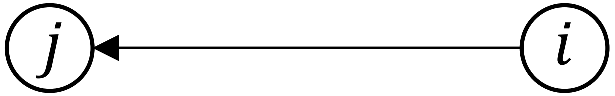
- For example,

$$m_{ij} = f(\vec{x_i} - \vec{x_j})$$



Translation Equivariance

$$m_{ij} = f(||\vec{x_i} - \vec{x_j}||_2)$$

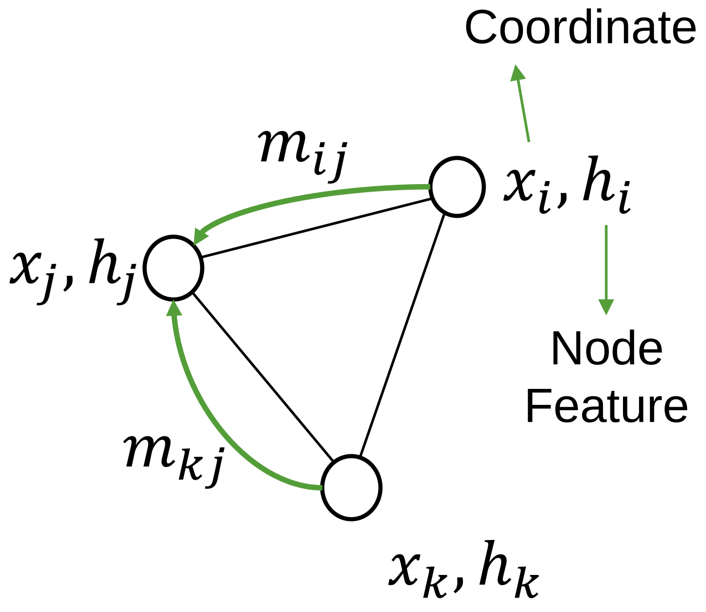


E(2)-Equivariance

MacroRank: EGCL

- Position encoding

$$m_{ij} = \Phi(h_i, h_j, PE_n(\vec{x_i} - \vec{x_j}))$$



where

$$PE_n = \begin{bmatrix} \sin \pi ||\vec{x_i} - \vec{x_j}||_2 \\ \cdots \\ \sin 2^n \pi ||\vec{x_i} - \vec{x_j}||_2 \\ \cos \pi ||\vec{x_i} - \vec{x_j}||_2 \\ \cdots \\ \cos 2^n \pi ||\vec{x_i} - \vec{x_j}||_2 \\ \vec{x_i} - \vec{x_j} \end{bmatrix}$$

E(2)-equivariant

Translation equivariant

Translation Equivariant

Sensitive to small position changes.

MacroRank: Pairwise Rank Loss

- Predicted probability of $x_i \succ x_j$:

$$P\left(x_i \succ x_j\right) = \text{Sigmoid}(s_i - s_j)$$

- Weighted binary cross-entropy loss:

$$L_{ij} = \log\{1 + \exp(s_j - s_i)\}|\Delta Z_{ij}|$$

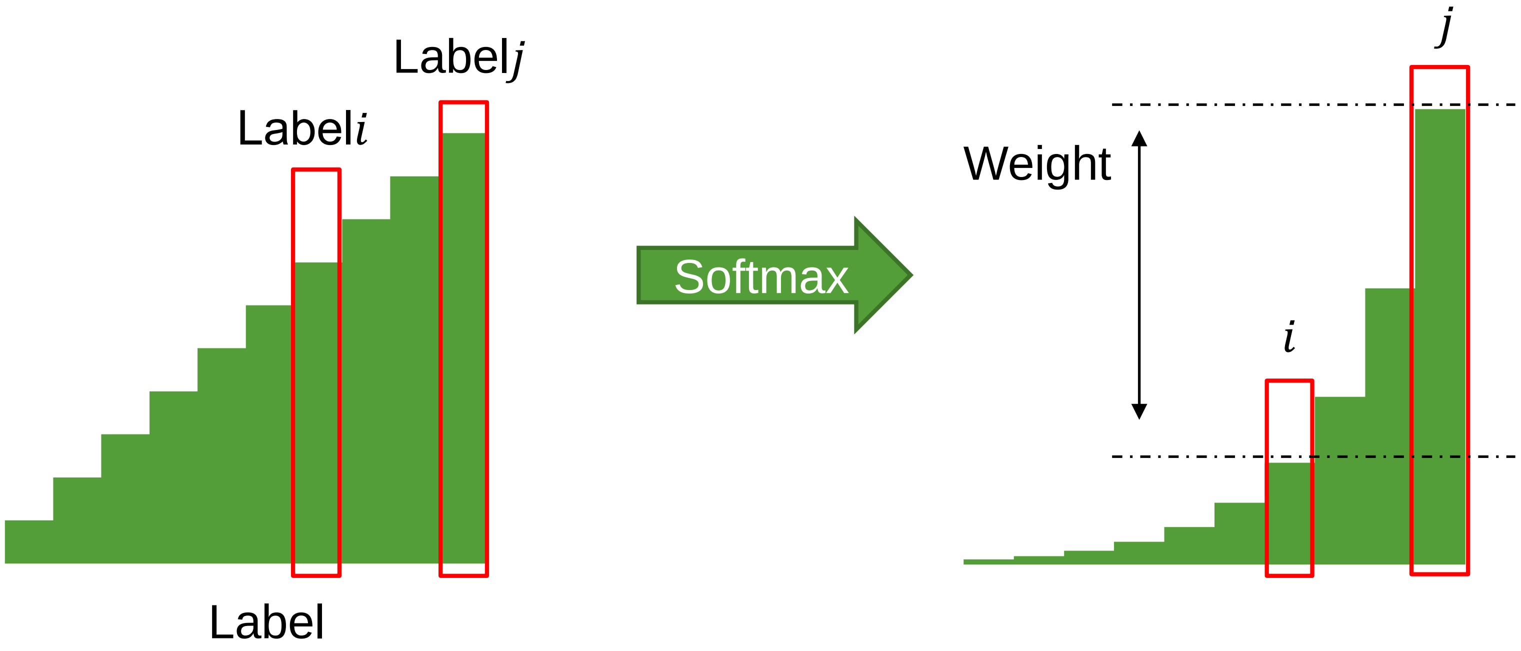
- Final loss function:

$$Loss = \sum_{design} \sum_{pair(i,j)} L_{ij}$$

MacroRank: Pairwise Rank Loss

- Weighting coefficient ΔZ_{ij} : focus on the samples with higher rank

$$\Delta Z_{ij} = \mathrm{Softmax}\left(y_i^{label}\right) - \mathrm{Softmax}\left(y_j^{label}\right) = \frac{\exp y_i}{\sum_p \exp y_p} - \frac{\exp y_j}{\sum_p \exp y_p}$$



- Higher rank, greater weight
- Larger difference, greater weight

Experiment: Dataset

- Dataset:
  - 12 designs in ISPD 2015 benchmark, free all macros.
  - Placed by DREAMPlace, perturb the result in macro legalization stage.
  - Global Routing: CU. GR
  - Divided to 2 groups, one for training, one for testing, cross validation.

Group	Design Name	#Macros	Macro Coverage	#Instances	#Nets	#Macro Placements
1	des perf a	4	50%	108666	110281	300
	fft a	6	65%	33641	32088	300
	matrix mult a	10	67%	154460	154284	296
	matrix mult c	10	67%	151247	151612	296
	superblue14	336	48%	633661	619697	299
	superblue19	280	60%	521805	511606	298
2	edit dist a	6	29%	129993	131134	300
	fft b	11	69%	33646	32088	300
	matrix mult b	10	67%	151247	151612	294
	pci bridge32 b	8	47%	29283	29417	299
	superblue11 a	1443	59%	954445	935613	284
	superblue16 a	419	48%	698367	680450	299

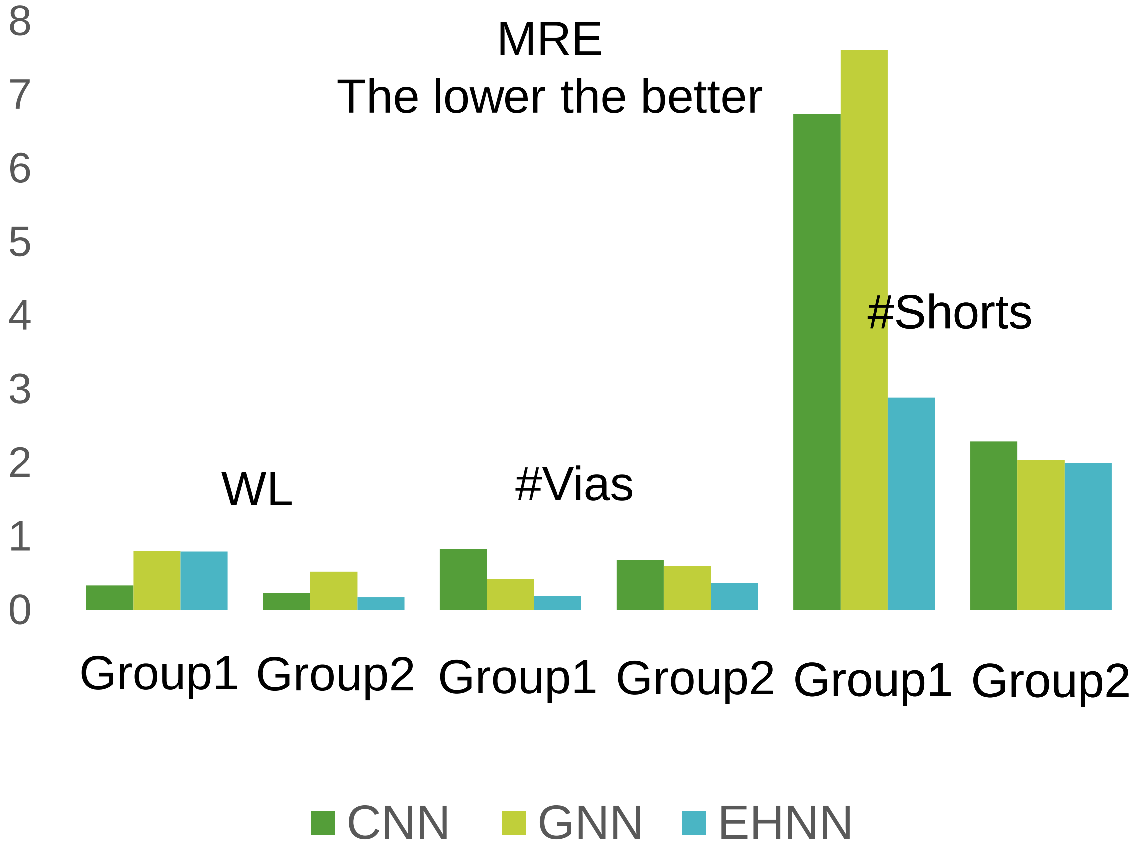
Experiment: Setting

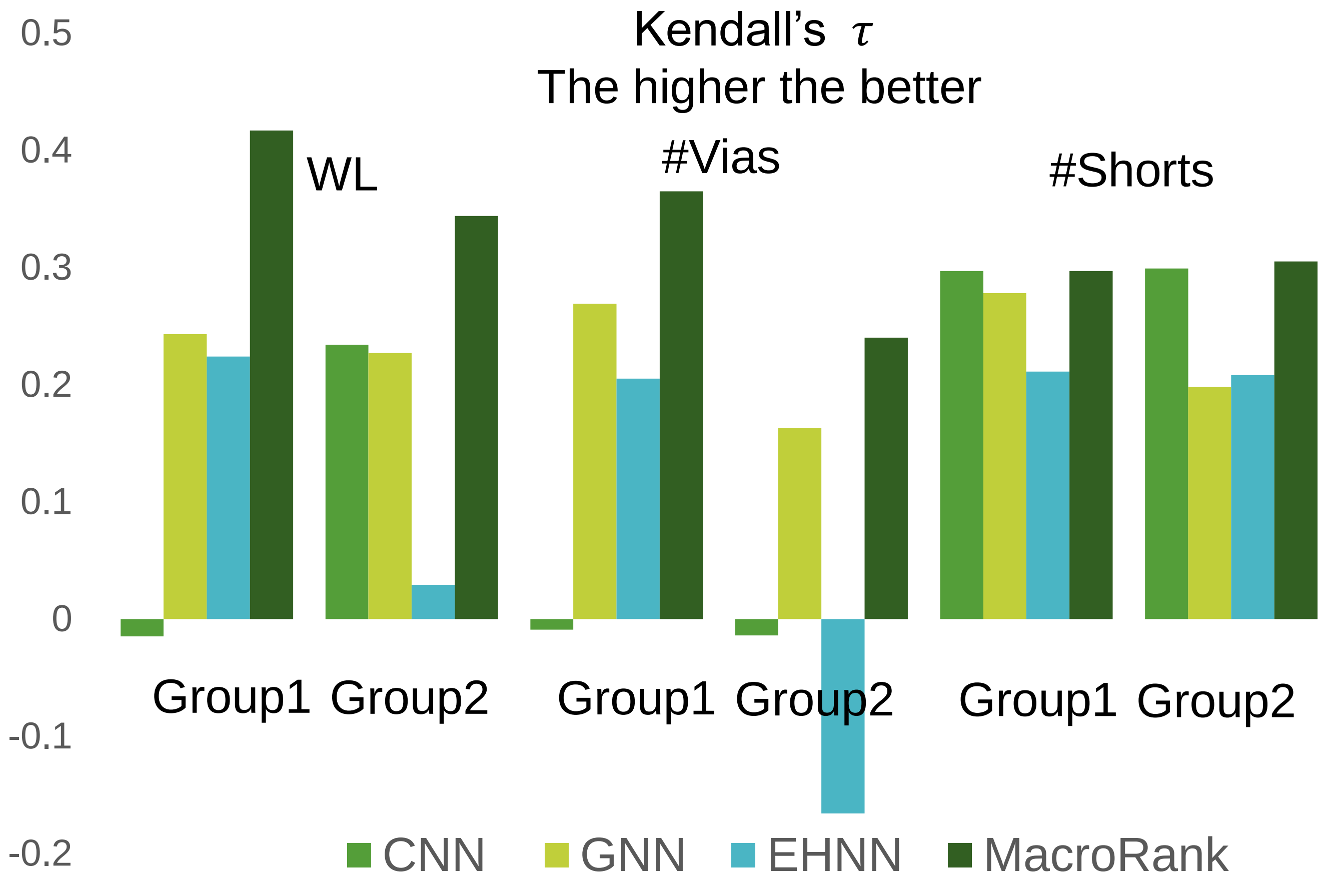
- Training:
  - Implemented by Pytorch Geometric.
  - A Nvidia 2080Ti
  - 400 epochs, ~6 hours

- Code Release:
  - https://github.com/PKU-IDEA/MacroRank

Experiment: MRE and Kendall's τ

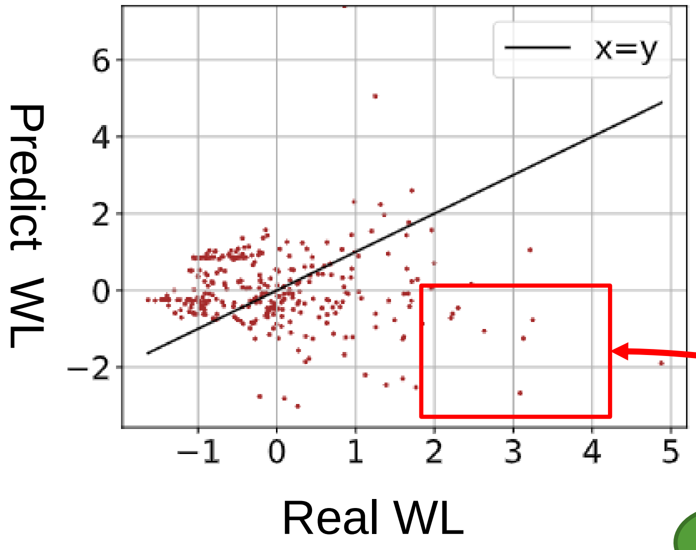
- EHNN dominates GNN in all groups (MRE).
- MacroRank (=EHNN + LTR) achieves the best Kendall's τ on all the groups,
  - 49.5% better than CNN.



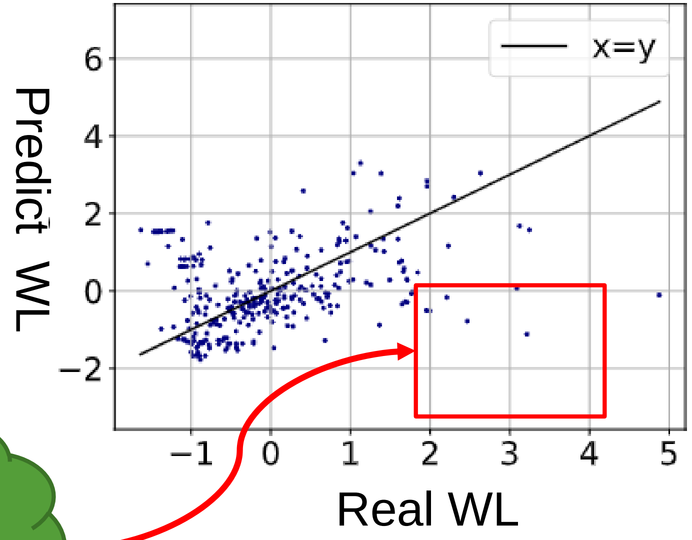


Experiment: MRE and Kendall's τ

CNN

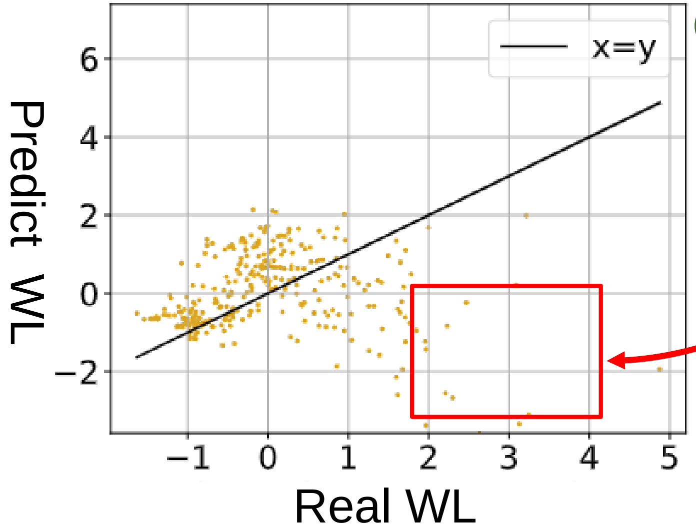


GNN

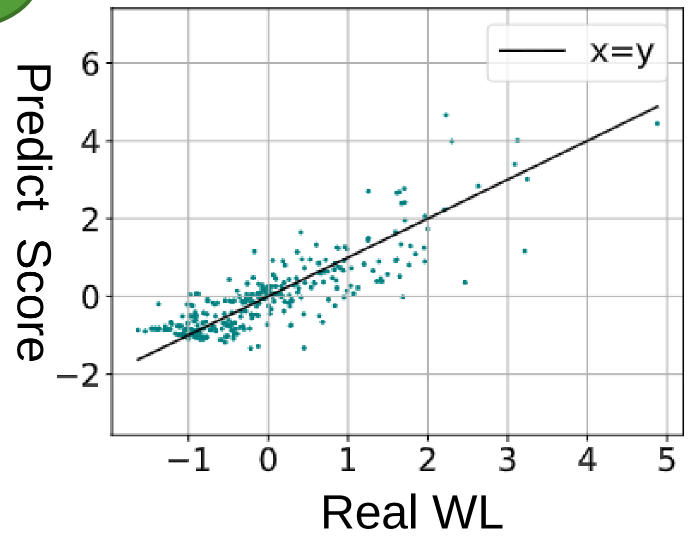


Obvious error !

EHNN

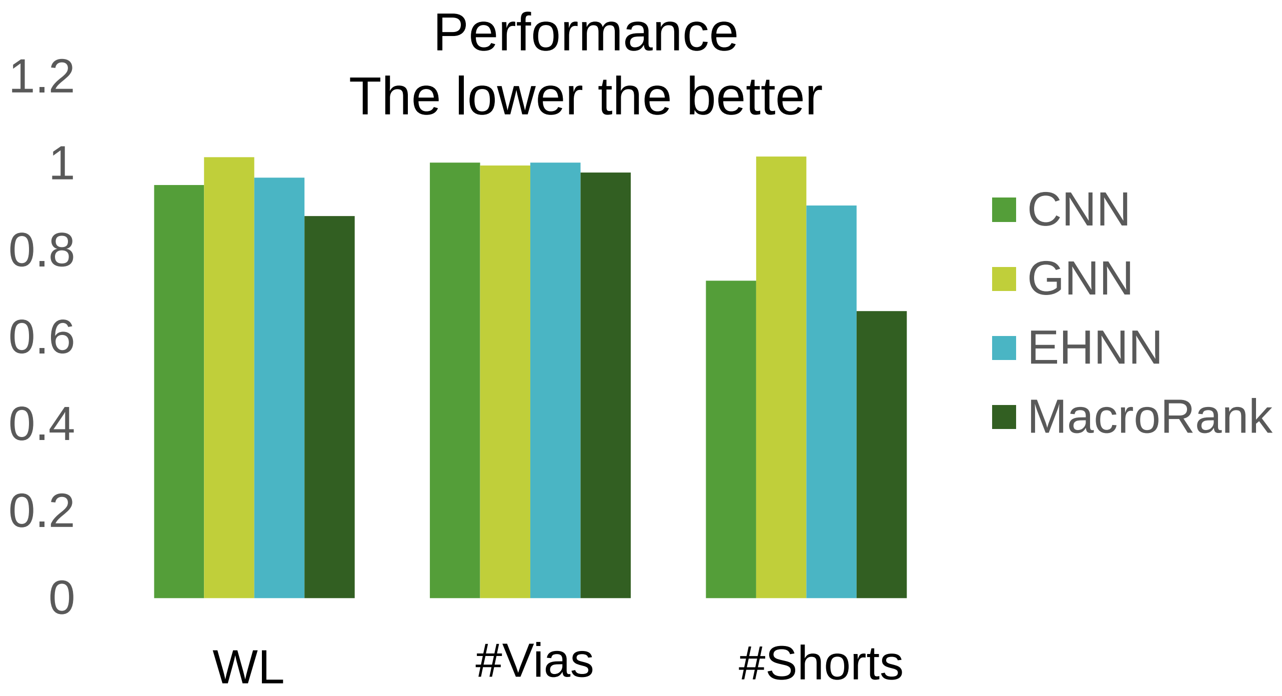


MacroRank = EHNN+LTR



Experiment: Top 30 Prediction

	MEAN	CNN	GNN	EHNN	MacroRank
WL	1	0.951	1.015	0.968	**0.88**
#Vias	1	1.003	0.996	1.003	**0.98**
#Shorts	1	0.731	1.017	0.904	**0.661**



Conclusion

- MacroRank: translation equivariance & LTR.
- Accurately predict the relative order of the quality of macro placement solutions.
- Improve the Kendall's τ by 49.5%
- Improve the average performance of top-30 prediction by 8.1%, 2.3%, and 10.6% on wirelength, vias, and shorts, respectively.

- Future Work
  - Integrate the model in macro placement algorithm.

Thanks!
Questions are welcome!

Email: chenyifan2019@pku.edu.cn