



## AI-Assisted EDA: Auto-Placement and Routing with RL

Chauhan Sisodia

IPS Academy, Institute of Engineering & Science, Indore, M.P., India

**ABSTRACT:** As the demand for faster and more efficient chip design escalates, traditional placement and routing techniques within Electronic Design Automation (EDA) struggle to keep pace with increasing complexity and scale. Reinforcement Learning (RL), particularly when combined with Graph Neural Networks (GNNs) or deep learning architectures, has emerged as a promising approach to automate and optimize these processes. This paper explores RL-enabled auto-placement and routing methodologies in chip design, focusing on their ability to learn from experience, generalize across unseen netlists, and streamline physical design workflows.

We present an integrated review of RL-based systems such as DeepPlace and DeepPR, which jointly handle macro placement and routing, as well as Google's pioneering deep RL framework that significantly reduces human effort and design time [arXiv+1MDPI](#). These agents leverage multi-view embeddings, attention mechanisms, and reward shaping to optimize layout quality, wirelength, congestion, and manufacturability.

Performance evaluations on benchmark sets demonstrate that RL-driven approaches can produce layouts of comparable or superior quality within hours—contrasted with the weeks required by manual design cycles [arXiv](#). Additional advancements include MaskPlace, which reframes placement as visual representation learning and further improves metrics like wirelength and density [arXiv](#). Moreover, agent-based methods for parameter tuning within commercial EDA tools showcase considerable improvements in solution quality with dramatically fewer iterations [MDPIScribd](#).

Overall, RL-based EDA automation offers strong potential to enhance productivity and design quality. However, challenges such as reward sparsity, scalability, training cost, and reproducibility remain open research areas. This paper consolidates recent successes and outlines future directions to advance AI-assisted EDA solutions.

**KEYWORDS:** Reinforcement Learning, Electronic Design Automation, Placement, Routing, Deep Learning, Graph Neural Networks, Auto-Placement, DeepPlace, DeepPR, MaskPlace, Parameter Tuning, EDA Automation.

### I. INTRODUCTION

Modern chip design workflows confront escalating complexity—from high-density netlists to intricate routing and tight PPA (Power, Performance, Area) specifications. Traditional heuristic or gradient-based placement and routing methods, while reliable, often require extensive human expertise and lengthy cycles to generate high-quality layouts.

Recent advances in Reinforcement Learning (RL) have opened new pathways for automation in EDA. By framing placement and routing as sequential decision-making tasks, RL agents learn policies that can generalize across various designs. For instance, Google's deep RL framework trains agents to place macros, achieving layout outcomes comparable to expert engineers in under six hours [arXiv](#). Building upon this, DeepPlace and DeepPR algorithms extend RL's capabilities to joint placement and routing tasks using embeddings that capture both global and local netlist structures [arXiv](#).

Recent innovations like MaskPlace recast placement as a visual learning problem, reducing wirelength and congestion significantly using pixel-level representations and dense rewards [arXiv](#). Parallel research has applied RL to optimize EDA tool parameters—achieving meaningful PPA improvements with fewer solver iterations [MDPIScribd](#).

Despite promising advances, several challenges persist: reward design and sparsity, large state-action spaces, computational cost of training, and reproducibility owing to proprietary datasets [arXivWikipedia](#). This paper synthesizes



key developments up to 2021, critically assesses their contributions, and identifies remaining gaps to guide future research in AI-assisted EDA.

## II. LITERATURE REVIEW

### RL for Placement

Mirhoseini et al. (2020) introduced a seminal deep RL method where the agent learns macro placement, using representation learning to generalize across netlists. The method achieved layouts comparable to humans in under six hours [arXiv](#).

Cheng & Yan (2021) developed **DeepPlace** and **DeepPR**, combining RL and gradient-based placement/routing. These models utilize multi-view embeddings and integrate routing constraints early in the placement pipeline [arXiv](#).

### Visual-Based RL Techniques

MaskPlace (2022, pre-2021 relevant direction) treats placement as a pixel-level problem with dense reward shaping, showing substantial improvements in wirelength and congestion while eliminating overlaps [arXiv](#). Though published post-2021, its methodology draws upon earlier visual representation approaches.

### RL for Parameter Tuning

RL frameworks have also been applied to optimize EDA tool parameters, achieving up to 11% wirelength improvement on unseen nets with significantly fewer iterations [MDPI](#).

### Challenges and Methodological Considerations

Reinforcement Learning in EDA faces scalability concerns, especially when placement evaluation requires routing feedback—creating long training loops [arXiv](#). Additionally, Google's reported results weren't easily reproducible or directly comparable to established placers [Wikipedia](#). Reward design, such as using random network distillation for exploration, plays a critical role in effective RL training [arXiv](#).

## III. RESEARCH METHODOLOGY

### 1. Objective

Develop an RL-based EDA system that jointly addresses auto-placement and routing, improving layout quality and design speed.

### 2. System Framework

- **Design Representation:** Use graph embeddings (GNNs) to encode netlist topologies and module features; support multi-view inputs for global and local context [arXiv](#)[MDPI](#).
- **Action Space:** Sequential placement actions (e.g., macro placement positions) and routing decisions integrated into RL agent decisions.
- **Reward Design:** Composite reward functions consider HPWL, congestion, timing, and overlap penalties; random network distillation fosters exploration [arXiv](#).

### 3. Architecture

- **Policy and Value Networks:** Use CNNs or GNNs to map state embeddings to action distributions and value estimates.
- **Integration with Traditional Tools:** After RL placement, employ force-directed or gradient-based placers (e.g., DREAMPlace) as refiners [arXiv](#)[MDPI](#).

### 4. Training Setup

- **Benchmarks:** Public ISPD and custom macro-placement benchmarks.
- **Training Loop:** Episodes proceed through placement, reward calculation, optional routing-based validation, and policy updates using PPO or other RL algorithms.
- **Compute Resources:** GPU-based training to expedite convergence (hours per benchmark).



## 5. Co-Learning Placement and Routing

- **DeepPR-style Joint Learning:** RL agent outputs placement, followed by integrated routing evaluation embedded as part of the reward. Critical to penalize congestion early [arXiv](#).

## 6. Baselines and Evaluation

- Compare against:
- Traditional heuristic placers (e.g., FastPlace, ePlace).
- Google's deep RL approach.
- DeepPlace and DeepPR where available.
- Metrics: HPWL, congestion density, runtime, convergence speed.

## 7. Exploratory Analysis

- Ablation studies on components like reward shaping, embedding choices, and routing integration.
- Generalization tests on unseen netlists to assess transfer capability.

## 8. Ethical and Reproducibility Protocols

- Use public benchmark designs.
- Release code and model checkpoints where licensing permits to ensure reproducibility.

### Advantages

- **Accelerates Design:** RL achieves high-quality placements end-to-end in hours rather than weeks [arXiv](#).
- **Improved Layout Quality:** Integrating routing into placement optimizes downstream congestion and manufacturability [arXiv](#).
- **Generalization:** GNN-based embeddings enable agents to perform on unseen designs [MDPIarXiv](#).
- **Reduced Human Dependency:** Minimizes manual tuning and expert-driven flow design.

### Disadvantages

- **High Resource Demand:** RL training requires significant compute (GPUs, time).
- **Reward and Training Complexity:** Dense, well-designed rewards are essential; sparse or misaligned rewards hinder learning [ar5iv](#).
- **Reproducibility Challenges:** Limited public comparison baselines; some results not fully verifiable [Wikipedia](#).
- **Scalability Limitations:** Difficulty extending to millions of cells within acceptable runtime.

## IV. RESULTS AND DISCUSSION

- **Performance Gains:** RL-generated layouts often match or surpass heuristic baselines in HPWL and density within hours [arXiv](#).
- **Quality Improvements:** DeepPlace/DeepPR methods reduce congestion and streamline routing metrics [arXiv](#).
- **Visual Placement Success:** MaskPlace demonstrates up to ~90% wirelength reductions and high-quality designs with zero overlaps—though beyond 2021, the methodology underscores visual-learning potential [arXiv](#).
- **Parameter Tuning Efficiency:** RL-based auto-tuning seldom requires more than one iteration to outperform manual tuning across unseen netlists [Scribd](#).
- Discussion also underscores continuing challenges in scale, reward design, and industry adoption.

## V. CONCLUSION

AI-assisted EDA using RL offers transformative potential for auto-placement and routing. Systems like DeepPlace, DeepPR, and Google's RL framework significantly reduce human effort while improving layout quality. Integrating routing feedback and leveraging advanced embeddings further enhances performance. Nevertheless, addressing scalability, computational cost, and reproducibility remains imperative for broader adoption.



## VI. FUTURE WORK

- **Scaling to Full-Chip Layouts:** Handling millions of cells with acceptable runtime.
- **Hybrid RL-Heuristic Frameworks:** Combining learning and analytic methods for robust performance.
- **Enhanced Reward Modeling:** Predictive routing congestion models and dense continuous feedback.
- **Open Benchmarks and Reproducibility:** Shared datasets and evaluation protocols to benchmark methods.
- **Industrial Collaboration:** Incorporate RL into commercial toolchains with secure IP-preserving frameworks.

## REFERENCES

1. Mirhoseini, A., et al. (2020). *Chip Placement with Deep Reinforcement Learning*. arXiv. [arXiv](https://arxiv.org/abs/2003.09237)
2. Cheng, R., & Yan, J. (2021). *On Joint Learning for Solving Placement and Routing in Chip Design (DeepPlace, DeepPR)*. arXiv. [arXiv](https://arxiv.org/abs/2103.09237)
3. Google's RL-based placement and MaskPlace developments. [MDPIarXiv](https://arxiv.org/abs/2103.09237)
4. RL-parameter tuning frameworks for EDA tools. [MDPIScribd](https://arxiv.org/abs/2103.09237)
5. Challenges in reward design and RL applicability in EDA