RulePlanner: All-in-One Reinforcement Learner for Unifying Design Rules in 3D Floorplanning

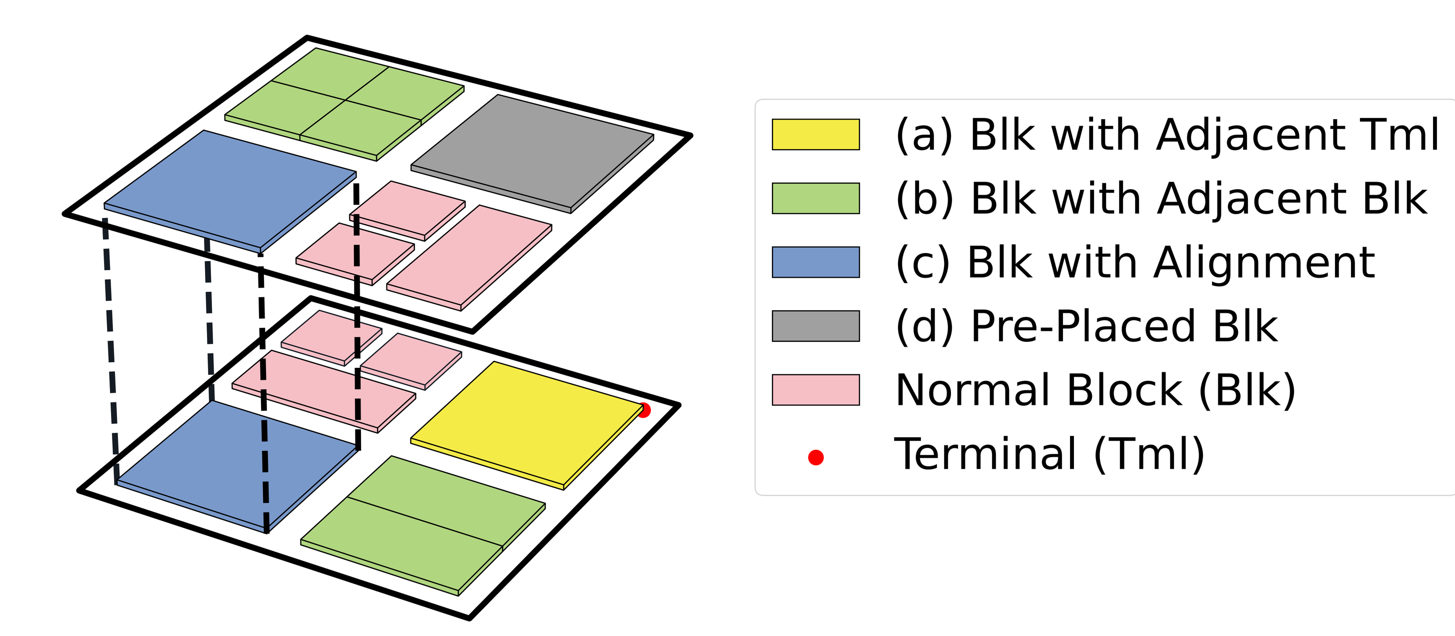
Ruizhe Zhong Xingbo Du Junchi Yan
Shanghai Jiao Tong University | Mohamed bin Zayed University of Artificial Intelligence

Motivation: Design Rules Are the Bottleneck

PROBLEM. 3D floorplanning must place blocks across multiple stacked dies while satisfying heterogeneous hardware design rules. Existing analytical, heuristic, and RL methods usually optimize a narrow objective such as HPWL or a small subset of rules, leaving violations for manual post-processing.

RULES ADDRESSED TOGETHER.

- Boundary constraints: blocks align with edge or corner terminals.
- Grouping constraints: related blocks must be physically abutted.
- Inter-die alignment: paired blocks across dies need projected overlap.
- Pre-placement, non-overlap, outline, and shape constraints.



Design rules in 3D floorplanning: specialized boundary/grouping/alignment rules plus common validity rules.

- Treat rule satisfaction as a first-class objective, not a post-processing repair.
- Encode each rule as a spatial matrix that the policy can see and act on.
- Enforce hard feasibility by masking invalid actions before sampling.

Key Idea

Compile each design rule into the same RL interface.

Rule	Matrix	Metric / Reward
Boundary	Adjacent terminal mask	Block-terminal distance ↓
Grouping	Adjacent block mask	Adjacency length ↑
Inter-die alignment	Alignment mask	Alignment score ↑
Validity rules	Position mask	Overlap / outline violations ↓

This turns heterogeneous geometric and topological constraints into spatial features, feasibility masks, and dense rule-aware feedback for policy learning.

Method Comparison

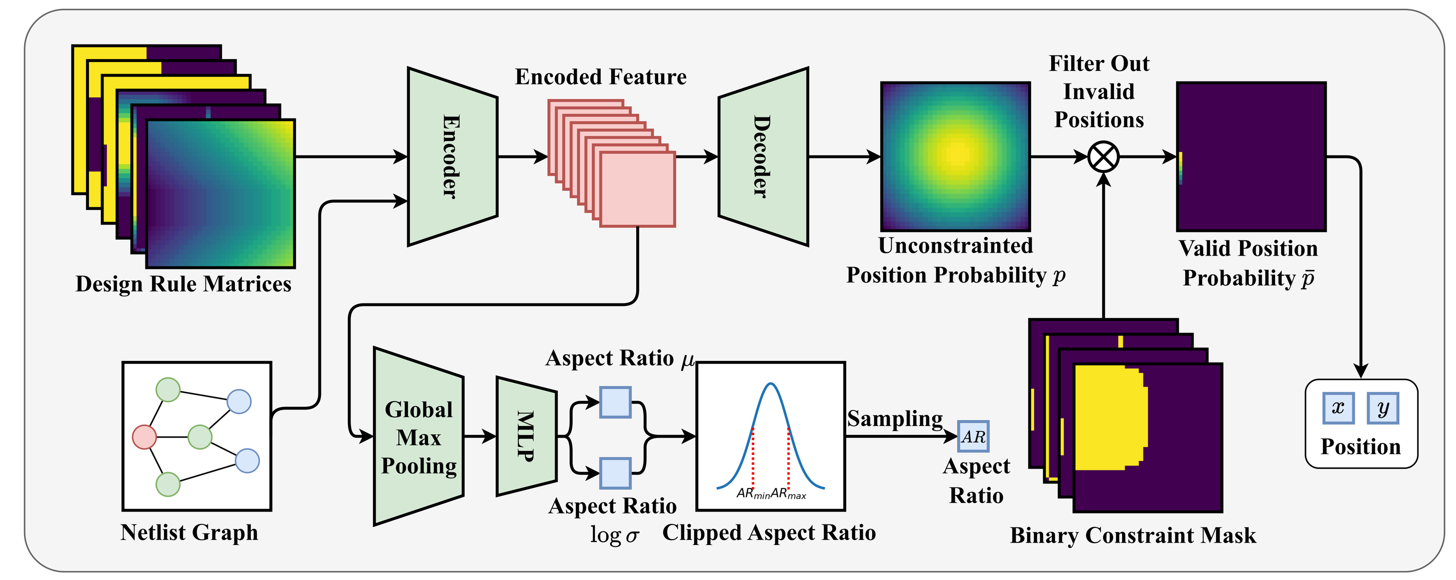
Approach family	Boundary	Grouping	Alignment
Analytical / heuristic	–	–	–
MaskPlace-style RL	–	–	–
FlexPlanner	–	–	✓
RulePlanner	✓	✓	✓

Prior systems usually specialize in one rule family. RulePlanner exposes a reusable representation-feasibility-policy interface, so multiple hard geometric rules can be enforced together.

Evaluation Setup

- Public MCNC and GSRC benchmarks: 8 circuits, from 10 to 300 blocks.
- Three task settings compose boundary, grouping, and inter-die alignment rules with common validity constraints.
- Baselines cover analytical, heuristic, black-box, and RL floorplanning methods; each experiment uses 5 seeds.
- Metrics cover rule satisfaction, HPWL, overlap, zero-shot transfer, and component ablations.

RulePlanner Pipeline



Stacked rule matrices and netlist graph feed an actor-critic policy. Invalid placements are removed before sampling.

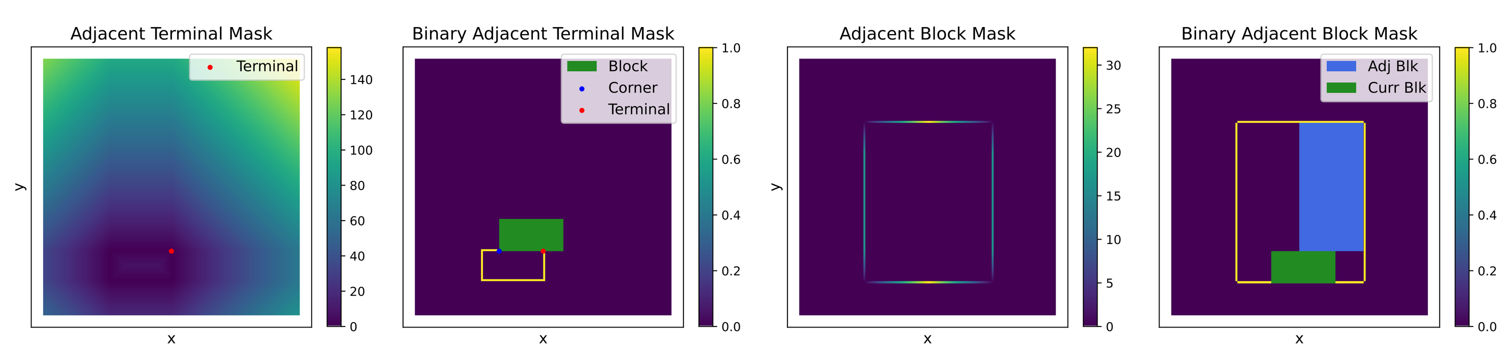
HYBRID ACTION. At step t, the policy chooses a discrete block location (x, y) and a continuous aspect ratio AR.

AVAILABILITY MASK. Rule-specific binary masks are combined:

$$\mathbf{M} = \bar{\mathbf{T}} \odot \bar{\mathbf{B}} \odot \bar{\mathbf{A}} \odot \bar{\mathbf{P}}, \qquad \bar{p}_\theta = \mathrm{softmax}(p_\theta + (\mathbf{M} - 1) \cdot 10^8).$$

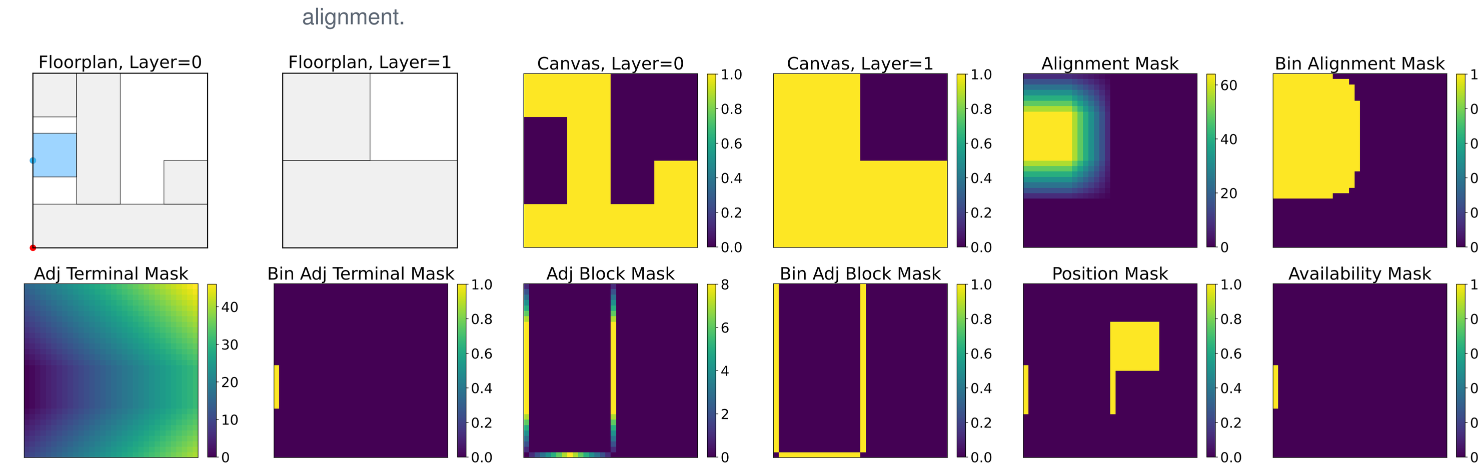
Only feasible positions remain in the action distribution; soft-block aspect ratios are clipped to the legal range.

Rule Matrices



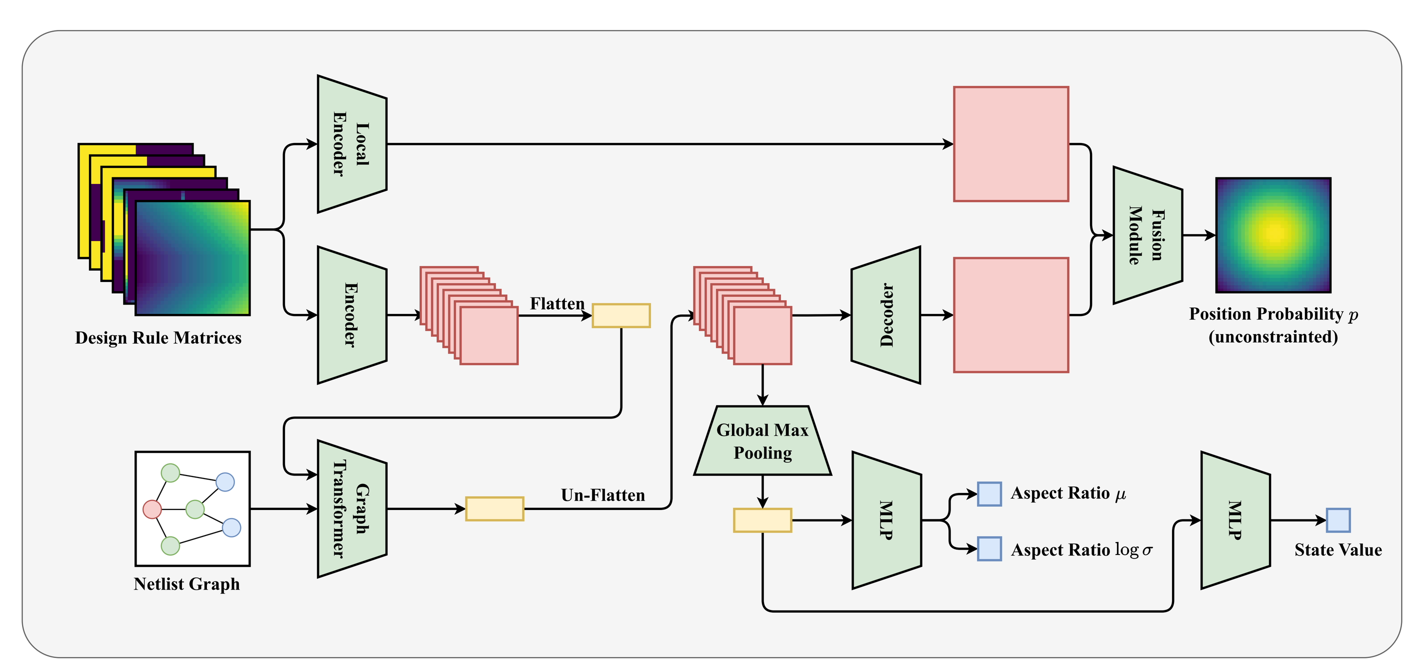
Adjacent terminal mask encodes where a block can satisfy boundary alignment.

Adjacent block mask encodes valid abutment regions for grouped modules.



All rule masks are fused into the availability mask used by the policy.

Model and Training



Vision features from rule matrices are fused with Transformer-encoded netlist features.

Training uses actor-critic Hybrid PPO. Dense rewards combine alignment, adjacency, terminal distance, overlap, and HPWL with circuit-aware normalization, which stabilizes transfer across circuits with different metric scales.

Main Results

TASK 3: BOUNDARY + GROUPING + INTER-DIE ALIGNMENT.

0.000 Block-terminal distance — full boundary satisfaction

0.162 Block-block adjacency — 3.0x best baseline avg.

0.938 Alignment score — best average alignment

Method	Tml Dist.↓	Blk Adj.↑	Align.↑	HPWL↓
Analytical	0.169	0.003	0.061	303,239
B*-3D-SA	0.328	0.042	0.484	313,819
WireMask-BBO	0.672	0.054	0.065	303,693
FlexPlanner	0.327	0.011	0.749	285,446
RulePlanner	**0.000**	**0.162**	**0.938**	**275,334**

RulePlanner improves rule satisfaction while retaining competitive HPWL; the method prioritizes layouts that are legal under complex hardware rules instead of requiring expensive manual legalization afterward.

Generalization and Ablation

Zero-shot transfer

Trained on *n100*, the policy transfers to unseen circuits in Task 3:

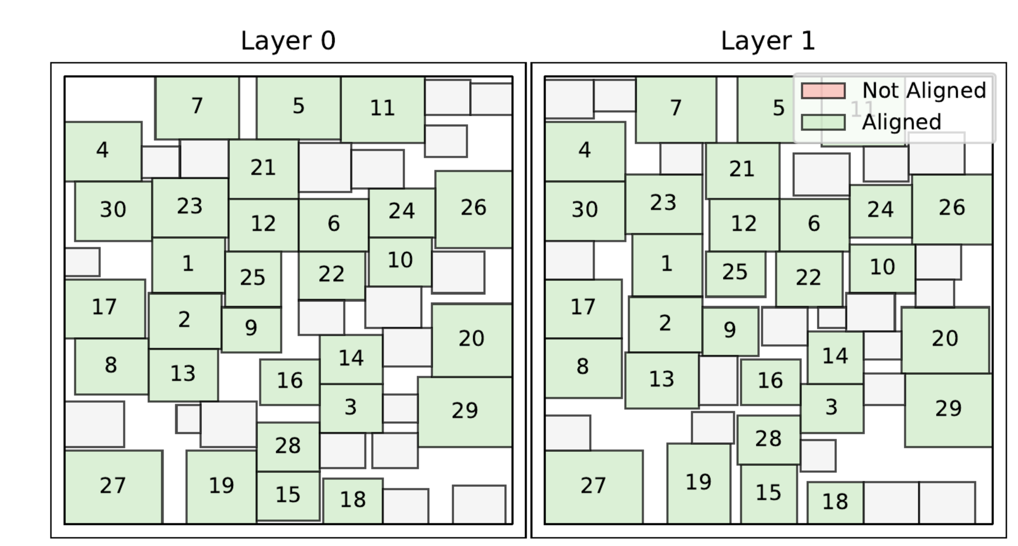
- terminal distance remains **0.000** across all evaluated circuits;
- alignment ratios stay near training performance: **0.946–1.026** on larger/unseen circuits;
- overlap remains **0.000**.

Ablation

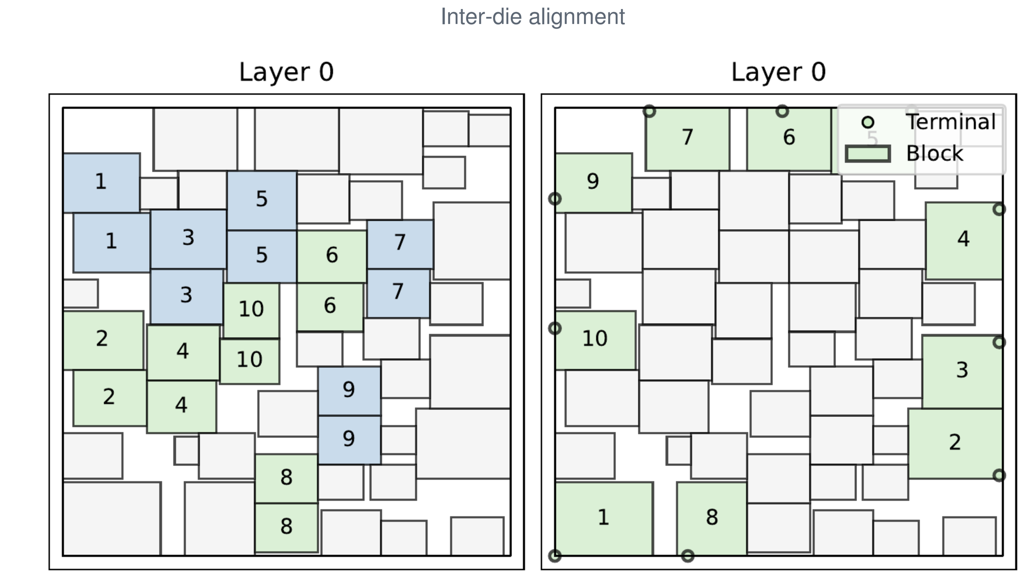
Feat.	Constr.	Adj.↑	Dist.↓
–	–	0.057	0.145
✓	–	0.153	0.058
–	✓	0.143	0.045
✓	✓	**0.169**	**0.000**

Rule matrices help representation; action constraints prune infeasible choices. Both are needed.

Task 3 Layout Example



Inter-die alignment



Grouped blocks Boundary terminals

Takeaway

- **Unified:** more than seven practical 3D floorplanning rules are handled in one RL framework.
- **Feasible by construction:** action-space masks remove illegal placements before the policy samples.
- **Extensible:** new geometric/topological rules can be added by defining a matrix, feasibility condition, and metric.
- **Practical scope:** current work focuses on geometric/positional rules; thermal, routability, and expert-tuned industrial validation are important future directions.