CIRCUIT-R1: CIRCUIT-GROUNDED REINFORCEMENT LEARNING FOR VERIFIABLE CIRCUIT2NETLIST

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent multimodal large language models have shown strong visual reasoning capabilities, yet reliable circuit-to-netlist translation remains challenging. Existing circuit-oriented reasoning methods use reinforcement learning to reward only a subset of intermediate reasoning stages, leaving the grounding of circuit connections insufficiently constrained. We investigate this limitation from two complementary perspectives. From an explicit perspective, our analysis shows that recognizing devices and ports alone is insufficient: wire tracing provides essential geometric evidence for recovering circuit topology. From an implicit perspective, generated netlists frequently violate basic structural and electrical principles, demonstrating the need for circuit-specific priors beyond visible image evidence. Motivated by these findings, we introduce **CIRCUIT-R1** (Circuit-Grounded Reinforcement Learning for Verifiable Circuit-to-Netlist Translation) . CIRCUIT-R1 decomposes netlist generation into five reasoning stages and defines continuous reward signals over bounding boxes, terminal centers, wire polylines, and endpoint assignments. We further propose the Circuit Physics Prior Verifier (CPPV), which represents each predicted netlist as a typed terminal hypergraph and detects structural and electrical violations without consulting the ground-truth netlist. We will release **Cir-GCoT**, a dataset of 2,440 circuit schematics with fine-grained grounding and topology annotations. Our results show that explicit visual evidence and implicit circuit priors jointly guide policy optimization toward accurate and reliable netlist generation.

1 INTRODUCTION

Circuit-to-netlist translation aims to recover electrical components and their interconnections from schematic images and express them as standardized SPICE netlists. Unlike circuit images, whose appearance varies with layout, drawing style, and symbol convention, a SPICE netlist represents device types, terminal connections, and electrical properties in a structured textual form (Bhandari et al. 2024). This unified representation abstracts away irrelevant visual variation and exposes the underlying circuit topology, providing a stable interface for downstream tasks such as circuit question answering, retrieval, analysis, and design optimization (Shi et al. 2024, 2025b). Converting heterogeneous schematic layouts into a common netlist representation therefore reduces ambiguity caused by visual differences and makes circuit diagrams directly accessible to subsequent reasoning and electronic-design-automation workflows.

Existing circuit-parsing methods provide only part of the required solution. Computer-vision pipelines combine component detection, net segmentation, and hand-crafted geometric post-processing (Zhang et al. 2022; Shi et al. 2025a; Jin et al. 2025). These systems benefit from task-specific visual modules, but can be brittle to symbol variations, occlusion, fragmented wires, and complex global layouts. General language and multimodal models have also been introduced into hardware-oriented workflows (Chen et al. 2024; Jiang et al. 2024). Their cross-modal knowledge improves semantic recognition, yet direct image-to-netlist generation provides little control over how a connection is obtained or whether the resulting circuit is structurally valid.

Chain-of-thought (CoT) reasoning provides a natural mechanism for decomposing this structured prediction process into intermediate decisions (Wei et al. 2022). Extensions such as zero-shot CoT, self-consistency, Tree of Thoughts, and Program of Thoughts improve reasoning through explicit

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

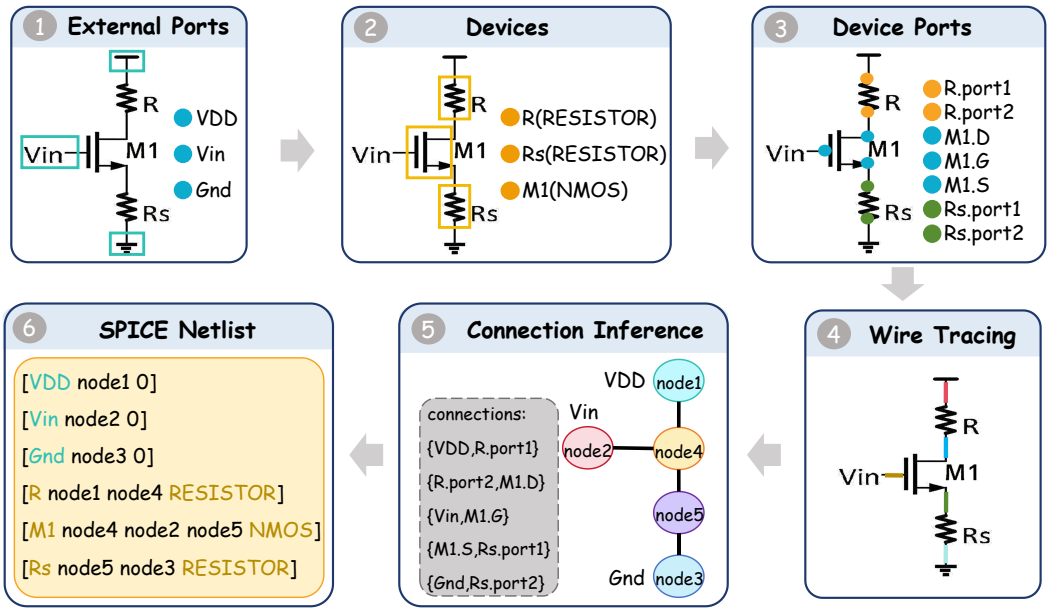

Figure 1: The model progressively identifies external ports, devices, device ports, wire trajectories, and connection groups before producing the final structured netlist.

intermediate computation or multiple reasoning paths (Kojima et al. 2022; Wang et al. 2023; Yao et al. 2023; Chen et al. 2022). In vision–language models, Visual CoT and LLaVA-CoT organize reasoning around localized visual evidence and structured stages (Ge et al. 2023; Zhang et al. 2024; Shao et al. 2024a; Xu et al. 2024), while Grounded CoT explicitly links intermediate decisions to image regions or coordinates (Xia et al. 2025; Wu et al. 2026c).

Reinforcement learning offers a complementary way to optimize such structured reasoning. RLHF aligns model behavior with human preferences (Ouyang et al. 2022), whereas GRPO estimates relative advantages from groups of sampled responses without a separate critic (Shao et al. 2024b; DeepSeek-AI et al. 2025). Circuit-Think (Jiang et al. 2026b) combines stepwise circuit reasoning with trajectory-guided reinforcement learning for external-port, device, and connection prediction. However, existing circuit reasoning still leaves two important forms of supervision underexploited. Device-port locations and the wire trajectories supporting each endpoint group are not explicitly supervised, so a plausible symbolic connection can receive credit without following the visible wires. Moreover, reference-based rewards do not ensure that a predicted circuit is internally valid: a netlist may still contain hallucinated terminals, duplicate assignments, undersized nets, or physically implausible internal shorts.

These limitations motivate a unified view of circuit reasoning through visual evidence and physics-derived verification. We therefore introduce CIRCUIT-R1, which couples a five-stage grounded reasoning process with continuous visual-evidence rewards and reference-independent circuit-physics verification. This design encourages predicted connections to follow the visible schematic evidence while remaining structurally and electrically valid. By coupling what the model sees with what the circuit permits, our framework turns connection prediction from unconstrained text generation into a verifiable, evidence-conditioned topology reconstruction process. It also provides localized learning signals when the final netlist is only partially correct.

Our contributions are: (1) we conduct a detailed analysis of circuit-to-netlist translation, showing that wire tracing is critical for topology recovery and that circuit-rule violations are prevalent in current multimodal models; (2) we propose CIRCUIT-R1, a circuit-grounded reinforcement learning framework that couples five-stage visual reasoning with continuous grounding rewards and reference-independent circuit-physics verification; and (3) extensive experiments validate the effectiveness of CIRCUIT-R1, and we will release **Cir-GCoT**, a dataset of 2,440 circuit schematics with fine-grained annotations.

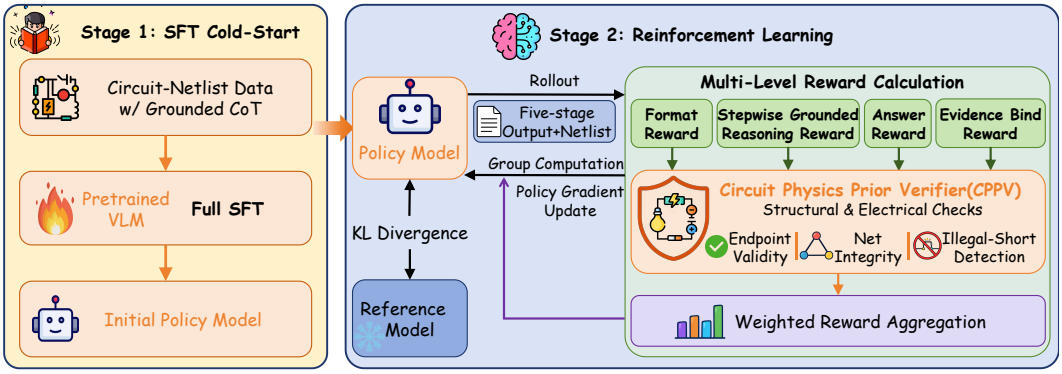

Figure 2: Overview of the Circuit-R1 training pipeline.

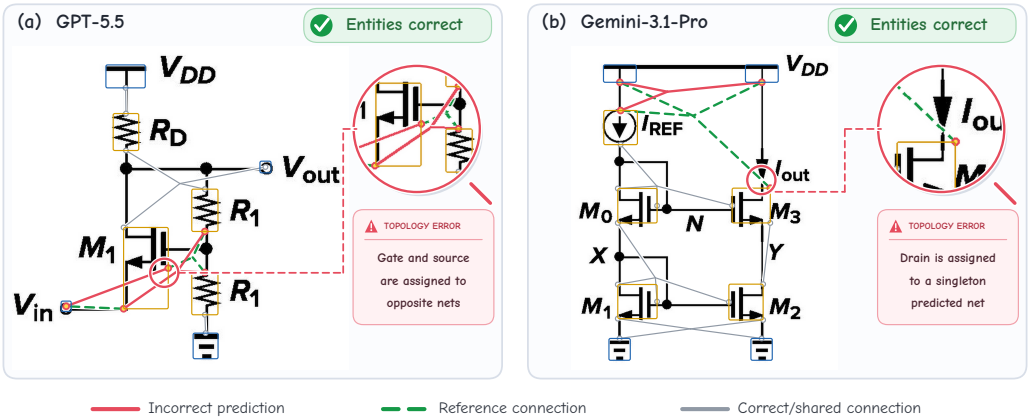

Figure 3: Representative connectivity failures of current multimodal models on AMSNet despite correct entity recognition.

2 Connectivity Bottleneck Analysis

We first test whether current multimodal-model failures stem only from entity recognition. On AMSNet, Gemini-3.1-Pro, Kimi-K3, and GPT-5.5 jointly predict devices, external ports, their bounding boxes, and the final SPICE netlist. Figure 3 shows two failures despite correct entity recognition: GPT-5.5 swaps the net assignments of an NMOS gate and source, while Gemini-3.1-Pro places an NMOS drain in a spurious isolated net. The errors therefore arise from topology recovery rather than missing entities.

Figure 4 quantifies this effect on AMSNet. Even when all devices and external ports are correctly classified and localized, exact topology fails for 63.5% of Gemini-3.1-Pro, 48.9% of Kimi-K3, and 57.8% of GPT-5.5 predictions; Answer Accuracy remains only 77.7–80.9%. Thus, terminal-to-net assignment remains a major bottleneck, motivating us to introduce device-port localization and wire tracing into the reasoning process.

2.1 Oracle Analysis of Wire Evidence

Using the cold-start SFT model, we replace preceding stages with ground-truth prefixes and continue generation from the next stage. This stage-wise oracle analysis estimates the upper-bound contribution of each intermediate representation.

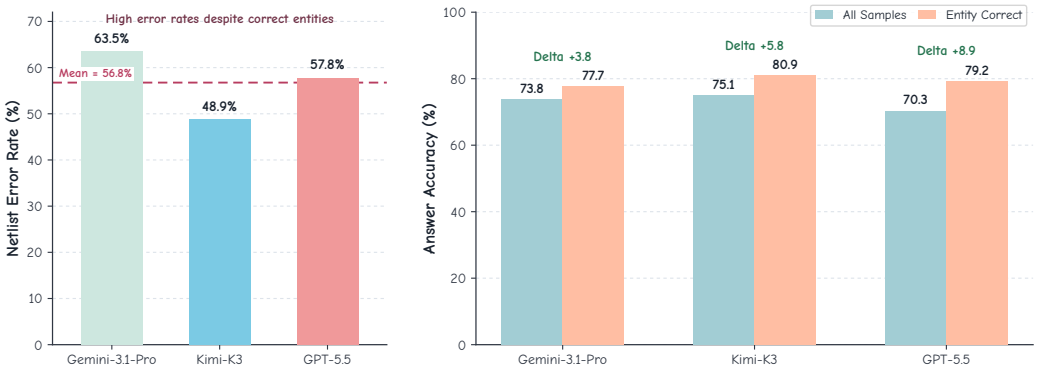

Figure 4: Left: the percentage of entity-correct predictions that still contain an incorrect final topology. Right: Answer Accuracy over all samples and over the subset with correct entity recognition.

Condition	Ground-truth prefix supplied	Model continues from	Answer Accuracy	Δ
C0	None	Full response	57.63	–
C1	External ports	Device recognition	64.01	+5.38
C2	+ devices	Device-port localization	69.96	+6.95
C3	+ device-ports	Wire tracing	75.82	+5.86
C4	+ grouped wire polylines	Connection inference	95.39	**+19.57**
C5	+ endpoint groups	Final answer	98.62	+3.23

Table 1: Cumulative stage-wise oracle analysis on AMSNet. The same cold-start SFT model receives all ground-truth stages up to each row.

Table 1 shows moderate gains from oracle ports, devices, and device ports, but a 19.57-point jump when grouped wire polylines are added (75.82% to 95.39%). These polylines encode only sparse wire geometry and trace grouping, without endpoint names or net identifiers. Endpoint groups add only 3.23 further points. The result highlights wire polylines as a valuable intermediate representation and motivates our wire-tracing stage and visual grounding rewards.

2.2 Circuit-Rule Violations in Predicted Netlists

Answer Accuracy measures reference-topology similarity, not internal circuit validity. A prediction may score highly while omitting a declared terminal, inventing an endpoint, duplicating an assignment, or shorting terminals within a component. We call these *circuit-rule violations*; unlike ordinary topology mismatches, they can be detected from the prediction alone.

Figure 5 shows such a failure. GPT-5.5 recognizes the components but assigns both terminals of an inductor and a capacitor to `net4`, self-shorting both components. Thus, correct entity recognition and substantial topology overlap do not imply electrical validity.

For each AMSNet prediction, we construct a typed terminal hypergraph and apply the five reference-independent CPPV checks in Section 3.1.2: missing terminals, singleton nets, undeclared endpoints, duplicate assignments, and internal shorts. A sample violates the rules if any check fails. To test whether violations persist as topology quality improves, let $\mathcal{S}_t$ contain predictions with Answer Accuracy at least t; we compute

$$\mathrm{CPVR}(t) = 100 \times \frac{\sum_{i \in \mathcal{S}_t} \mathbb{I}[C^{(i)}_{\mathrm{rule}} > 0]}{|\mathcal{S}_t|}. \tag{1}$$

Here, $t = 0$ includes all predictions, while larger t retains increasingly accurate ones. If topology similarity guaranteed validity, CPVR would approach zero as t increases.

Figure 6 confirms that topology quality and validity are related but distinct. For Gemini-3.1-Pro, Kimi-K3, and GPT-5.5, CPVR decreases from 32.19%, 29.08%, and 34.94% at $t = 0$ to 25.79%, 20.82%, and 27.33% at $t = 0.6$. Yet at $t = 0.8$, it remains 12.07%, 7.60%, and 9.22%. These residual violations motivate CPPV as a lightweight, reference-independent training signal.

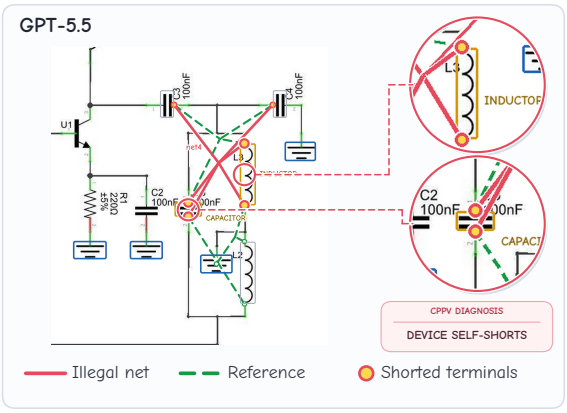

Figure 5: An Illegal-Short example in a GPT-5.5 prediction.

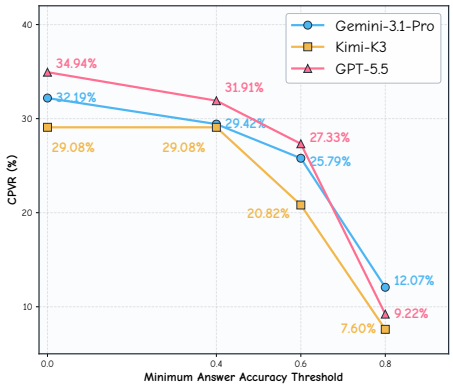

Figure 6: Answer-Accuracy-conditioned CPVR on AMSNet.

3 METHOD

3.1 CIRCUIT-R1

To emulate the progressive process by which humans interpret circuit schematics and strengthen the model's ability to recover connectivity, we organize each training response as a five-stage grounded reasoning trajectory: (1) external-port recognition, (2) device recognition, (3) device-port localization, (4) wire-trajectory tracing, and (5) connection inference by grouping endpoints into electrical nets. This data format establishes an explicit transition from object-level perception through fine-grained visual evidence to the final symbolic topology. Compared with Circuit-Think, our trajectory introduces device-port localization and wire-trajectory tracing as two explicit intermediate stages. Based on this, we design dense rewards for visual trajectories, together with CPPV for reference-independent circuit-physics verification. Finally, we aggregate all rewards and apply GRPO to optimize the policy model.

3.1.1 REWARD DESIGN

Circuit-R1 supervises both the final netlist Y and a grounded reasoning trajectory

$$Z = (P, D, T, W, C), \tag{2}$$

where P, D, T, W, and C denote external ports, devices, typed device terminals, wire trajectories, and endpoint groups, respectively. We assign continuous rewards to external-port and device recognition, terminal localization, wire geometry, wire-to-terminal grounding, and connection topology. We further introduce the Circuit Physics Prior Verifier (CPPV), which penalizes structurally and electrically invalid predictions without consulting the reference netlist.

Format reward. We use a binary reward to enforce the output schema. $R_{\text{format}} = 1$ when `<think>` contains the ordered sections `<external port>`, `<device>`, `<device port>`, `<trace>`, and `<connection>`, followed by a valid `<answer>`, with parseable entries and consistent evidence identifiers; otherwise, $R_{\text{format}} = 0$.

Stepwise grounded reasoning reward. To provide dense supervision throughout circuit interpretation, we introduce the Stepwise Grounded Reasoning Reward R_{SGR}, which combines four stage-level rewards.

External port reward. An external port is represented as `port_name(type)` followed by `[[x1,y1,x2,y2]]`. We construct a maximum-weight one-to-one matching $\mathcal{M}_P$ between predicted and reference ports of the same type, using bounding-box IoU as the matching weight. The reward is

$$R_{\text{port}} = \frac{\sum_{(i,j)\in\mathcal{M}_P} \text{IoU}(b_i, b_j)}{\max(|P_{\text{pred}}|, |P_{\text{gt}}|)}. \tag{3}$$

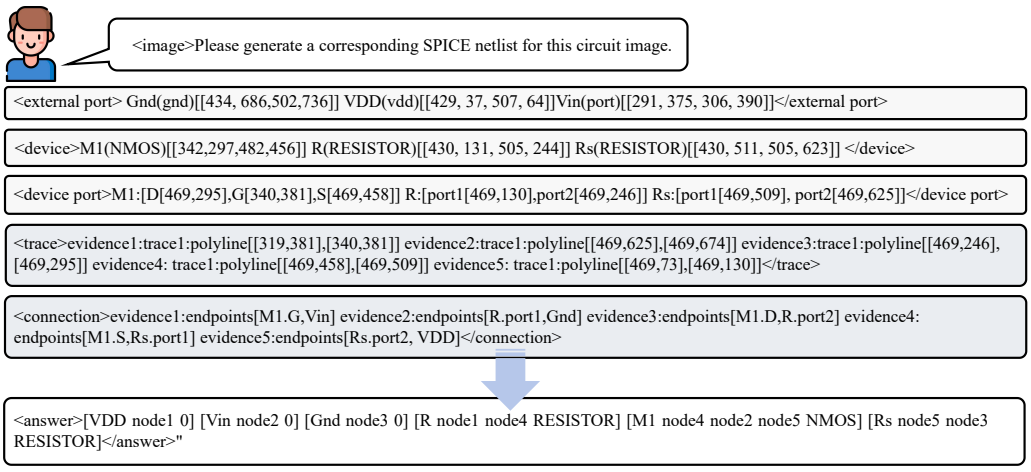

Figure 7: In CIRCUIT-R1, the first three stages establish object and port-level grounding, while the final two stages trace visible wire evidence and group endpoints into electrical nets.

Device reward. Device detection follows the same type-constrained, maximum-weight one-to-one IoU matching used for external ports:

$$R_{\text{det}} = \frac{\sum_{(i,j)\in\mathcal{M}_D} \text{IoU}(b_i, b_j)}{\max(|D_{\text{pred}}|, |D_{\text{gt}}|)}. \tag{4}$$

The model predicts the role and center of each device terminal. Let $\mathcal{M}_T$ be the role-compatible one-to-one terminal matching within each matched device, d_{ij} the terminal-center distance, and s the spatial tolerance. The terminal reward is

$$R_{\text{term}} = \frac{\sum_{(i,j)\in\mathcal{M}_T} \max\left(0, 1 - \frac{d_{ij}}{s}\right)}{\max(|T_{\text{pred}}|, |T_{\text{gt}}|)}. \tag{5}$$

Unmatched terminals are penalized through cardinality normalization. The final device reward is

$$R_{\text{device}} = R_{\text{det}} + R_{\text{term}}. \tag{6}$$

Visual reward. Each wire trace is represented by a sparse *polyline* with keypoints at endpoints, bends, and junctions. We interpolate each segment at approximately one-pixel spacing, yielding predicted and reference point sets A and B. Let $d(x, S) = \min_{y\in S} \|x - y\|_2$. With distance tolerance δ, geometric precision and recall are

$$P_{\text{wire}} = \frac{1}{|A|} \sum_{a\in A} \mathbb{I}[d(a, B) \le \delta]\,. \qquad R_{\text{wire}} = \frac{1}{|B|} \sum_{b\in B} \mathbb{I}[d(b, A) \le \delta]\,. \tag{7}$$

Their harmonic mean is computed globally over merged traces and locally after one-to-one Hungarian matching of evidence items. The visual reward is

$$R_{\text{visual}} = F_{\text{global}} + R_{\text{evidence}}. \tag{8}$$

where the two terms measure overall coverage and evidence-level path consistency, respectively.

Connection reward. Each connection is represented as an evidence-associated unordered endpoint set, for example `evidence1:endpoints[VDD,R1.Pos]`. After predicted entities are mapped to reference identities by the one-to-one spatial matches, we measure the similarity between predicted endpoint set E_i and reference endpoint set E_j as

$$q_{ij}^{\text{conn}} = \frac{2|E_i \cap E_j|}{|E_i| + |E_j|}. \tag{9}$$

Given the maximum-weight one-to-one connection assignment $\mathcal{M}_C$, the connection reward is

$$R_{\text{conn}} = \frac{\sum_{(i,j)\in\mathcal{M}_C} q_{ij}^{\text{conn}}}{\max(|C_{\text{pred}}|, |C_{\text{gt}}|)}. \tag{10}$$

Combining the four stages, the Stepwise Grounded Reasoning Reward is

$$R_{\mathrm{SGR}} = \lambda_{\mathrm{port}} R_{\mathrm{port}} + \lambda_{\mathrm{device}} R_{\mathrm{device}} + \lambda_{\mathrm{visual}} R_{\mathrm{visual}} + \lambda_{\mathrm{conn}} R_{\mathrm{conn}}, \tag{11}$$

The weights are selected empirically; see Appendix C.2.1 for details.

Evidence binding reward. Geometry alone does not reveal whether a trace is assigned to the correct terminals. For predicted evidence i and reference evidence j, we compute geometry F1 g_{ij} and endpoint-set F1 e_{ij} after mapping predicted entity identities through the one-to-one spatial matches. Their joint quality is the harmonic mean

$$b_{ij} = \frac{2 g_{ij} e_{ij}}{g_{ij} + e_{ij}}, \tag{12}$$

with $b_{ij} = 0$ if either term is zero. A single Hungarian assignment directly maximizes b_{ij}, and the binding reward is

$$R_{\mathrm{binding}} = \frac{\sum_{(i,j)\in\mathcal{M}} b_{ij}}{\max(|W_{\mathrm{pred}}|, |W_{\mathrm{gt}}|)}. \tag{13}$$

This formulation provides a smooth reward signal: improving either geometric alignment or endpoint-set agreement increases the score, while a high reward requires both to be accurate.

Answer reward. The final answer is scored independently from the stepwise connection groups. We convert the predicted and reference netlists into multisets of canonicalized unordered endpoint sets. Let $c_{\mathrm{pred}}(S)$ and $c_{\mathrm{gt}}(S)$ denote the multiplicities of network signature S in the predicted and reference netlists, respectively. The answer reward is their exact-net multiset Jaccard similarity:

$$R_{\mathrm{answer}} = J^{A}_{\mathrm{net}} = \frac{\sum_{S\in\mathcal{U}} \min(c_{\mathrm{pred}}(S), c_{\mathrm{gt}}(S))}{\sum_{S\in\mathcal{U}} \max(c_{\mathrm{pred}}(S), c_{\mathrm{gt}}(S))}, \tag{14}$$

where $\mathcal{U}$ is the union of predicted and reference network signatures.

The reference-based reward before circuit-physics verification is

$$R_{\mathrm{base}} = \alpha R_{\mathrm{SGR}} + \beta R_{\mathrm{binding}} + \gamma R_{\mathrm{answer}} + \delta R_{\mathrm{format}}. \tag{15}$$

The weights are selected empirically; see Appendix C.2.2 for details.

3.1.2 Circuit Physics Prior Verifier

Reference matching alone cannot express whether a prediction is internally well formed. We therefore parse the predicted entities and nets into a typed terminal hypergraph $H = (V, E, \tau)$, where V is the set of declared terminals, E is the set of predicted nets represented as hyperedges, and τ records device and terminal types. The verifier counts five types of violations:

1. c_{miss}: declared external and device terminals missing from all predicted nets;
2. c_{dangling}: dangling endpoints that form singleton predicted nets;
3. c_{exist}: endpoint occurrences not declared by the predicted entities;
4. c_{dup}: repeated occurrences of the same endpoint;
5. c_{short}: internal short circuits within a component. This includes placing both terminals of a resistor, capacitor, eligible inductor, or voltage source in the same predicted net. For a MOSFET, a gate–source short is also counted because it forces $V_{GS} = 0$, leaving the device in the off state.

$$C_{\mathrm{rule}} = c_{\mathrm{miss}} + c_{\mathrm{dangling}} + c_{\mathrm{exist}} + c_{\mathrm{dup}} + c_{\mathrm{short}}. \tag{16}$$

The CPPV penalty and final reward are

$$P_{\mathrm{CPPV}} = \min(P_{\max}, \eta C_{\mathrm{rule}}), \qquad R = R_{\mathrm{base}} - P_{\mathrm{CPPV}}, \tag{17}$$

where η is the cost per violation and $P_{\max}$ caps the total deduction. We use $\eta = 0.1$ and $P_{\max} = 0.5$. Thus, additional violations are penalized linearly until the cap is reached, preventing severely malformed predictions from dominating optimization.

4 EXPERIMENTS

4.1 SETUP

IMPLEMENTATION. We train Qwen3-VL-4B with two epochs of full-parameter SFT in LLaMA-Factory, followed by five epochs of reinforcement learning in EasyR1, on four NVIDIA A800 GPUs. Full hyperparameters are provided in Appendix C.1.

DATASETS. Our **Cir-GCoT** dataset contains 2,440 circuit schematics with fine-grained reasoning and netlist annotations. We reserve 426 samples for testing, use a random 1,000-sample subset of the remaining 2,014 for SFT, and use all 2,014 for reinforcement learning. We additionally evaluate cross-dataset generalization on AMSNet.

METRICS. Answer Accuracy is computed as the Jaccard similarity between the predicted and ground-truth SPICE netlists. Connection F1 measures pairwise connectivity after expanding each net into unordered terminal pairs. Circuit Physics Violation Rate (CPVR) is the percentage of samples with detectable circuit-rule violation.

4.2 MAIN RESULTS

We compare CIRCUIT-R1 with general-purpose VLMs and a specialized computer-vision pipeline on our benchmark and the publicly available AMSNet dataset (Tao et al. 2024). As shown in Table 2, CIRCUIT-R1 achieves SPICE netlist accuracies of 76.60 and 77.67 on the two benchmarks, respectively. It outperforms Kimi-K3 by 13.77 and 2.60 percentage points. It also surpasses GPT-5.5 by 15.41 and 7.35 points. Compared with the vision-based Auto-SPICE pipeline (Bhandari et al. 2024), the corresponding gains are 24.27 and 7.43 points. These results demonstrate state-of-the-art netlist generation performance with a substantially smaller model.

Method	Params	Thinking	Cir-GCoT	AMSNet
Qwen3-VL-4B	4B	✓	5.72	13.08
Qwen3-VL-32B	32B	✓	16.83	29.15
Gemini-3.1-pro	–	×	62.24	73.81
GPT-5.5	1.5T	×	61.19	70.32
Kimi-K3	2.8T	×	62.83	75.07
Auto-SPICE*	–	×	52.33	70.24
Circuit-Think†	7B	✓	58.54	69.32
CIRCUIT-R1	4B	✓	**76.60**	**77.67**

* It annotates the images using YOLO-V8 and generates the netlists using GPT-5.5.
† We reproduce Circuit-Think because its data and code are unavailable.

Table 2: The SPICE netlist accuracy on our benchmark and the AMSNet dataset.

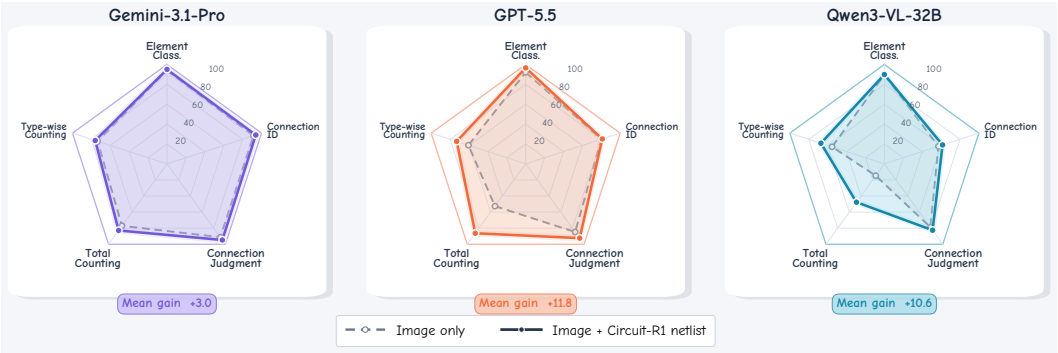

Figure 8: Downstream circuit understanding on AMSBench. Each spoke reports accuracy (%). Each model is evaluated using the circuit image alone and with the netlist generated by CIRCUIT-R1.

To assess whether the generated netlists benefit downstream circuit understanding, we further evaluate three general-purpose VLMs on five AMSBench tasks. Each model is tested with the schematic alone and with the same schematic augmented by the netlist generated by CIRCUIT-R1. As shown in

Figure 8, the generated netlist improves all five tasks across the three models. The gains are particularly pronounced for counting and connection judgment, while the already strong Gemini-3.1-Pro also benefits consistently from the additional circuit representation. This consistent improvement indicates that the generated topology provides a compact structural representation that complements the image rather than merely repeating visible component labels. It is especially useful for tasks that require multi-hop reasoning over component connectivity. The netlist therefore serves as an explicit intermediate representation of relations that a downstream model would otherwise need to recover repeatedly from image pixels.

Visible Evidence and Physics Verification. Table 3 presents a controlled 2×2 analysis of wire tracing and CPPV under identical settings. Wire tracing raises Connection F1 from 67.98 to 79.04 and Answer Accuracy from 62.13 to 72.37, while CPPV reduces CPVR from 41.57 to 18.13 and increases Answer Accuracy to 66.28. Combining both achieves the highest Connection F1 (84.68) and Answer Accuracy (76.60), while reducing CPVR to 20.91, demonstrating their complementary benefits. Wire tracing primarily improves topology reconstruction, whereas CPPV targets malformed outputs that reference similarity alone may overlook. Their combination therefore improves both topology agreement and internal validity.

Variant	Wire	CPPV	Connection F1	CPVR	Accuracy
Neither	×	×	67.98	41.57	62.13
Wire only	✓	×	79.04	28.42	72.37
CPPV only	×	✓	71.37	18.13	66.28
CIRCUIT-R1	✓	✓	84.68	20.91	76.60

Table 3: Complementarity of explicit wire evidence and circuit-physics verification.

Three-stage†	Device port	Wire	Accuracy
×	×	×	47.73
✓	×	×	63.54
✓	✓	×	66.28
✓	✓	✓	**76.60**

Table 4: Ablation of the Reasoning Design.

† In Table 4, three-stage denotes the port–device–connection trajectory introduced by Circuit-Think.

4.3 Ablation Studies

Grounded Reasoning Stages. Table 4 progressively augments direct netlist generation with three-stage reasoning, device-terminal localization, and wire tracing. Each row cumulatively introduces a finer-grained representation of circuit structure, allowing us to assess the contribution of increasingly explicit visual evidence. Answer Accuracy increases from 47.73% to 63.54%, 66.28%, and 76.60%, respectively, showing that increasingly explicit terminal and wire representations improve topology recovery.

Reward design	Accuracy
w/o R_{term}	74.81
w/o R_{visual}	70.04
w/o R_{binding}	72.16
CIRCUIT-R1	**76.60**

Table 5: Ablation of the reward design.

Grounding Rewards. Table 5 separately ablates terminal localization, wire geometry, and evidence-to-endpoint binding rewards. Removing R_{term}, R_{visual}, and R_{binding} lowers Answer Accuracy by 1.79, 6.56, and 4.44 points, respectively. The larger drops without R_{visual} and R_{binding} confirm the importance of both wire geometry and its binding to circuit endpoints. The decrease without R_{binding} further shows that accurate wire geometry must be associated with the correct endpoint set rather than rewarded in isolation.

5 Conclusion

We presented CIRCUIT-R1, an electrically verifiable grounded chain-of-thought framework for circuit-to-netlist translation. The method couples continuous visual grounding rewards with a consistency verifier over typed terminal hypergraphs. Experiments results show that these signals improve netlist answer accuracy using a Qwen3-VL-4B backbone.Nevertheless, CPPV incorporates circuit knowledge only through a limited set of physical consistency checks. Future work could incorporate device-level physical parameters and SPICE simulation to further strengthen the integration of electrical characteristics into the verification and learning process.

REFERENCES

Arcuschin, I.; Janiak, J.; Krzyzanowski, R.; Rajamanoharan, S.; Nanda, N.; and Conmy, A. 2025. Chain-of-Thought Reasoning in the Wild Is Not Always Faithful. In *ICLR Workshop on Reasoning and Planning for Large Language Models*.

Bhandari, J.; Bhat, V.; He, Y.; Garg, S.; Rahmani, H.; and Karri, R. 2024. Auto-SPICE: Leveraging LLMs for Dataset Creation via Automated SPICE Netlist Extraction from Analog Circuit Diagrams. *arXiv preprint arXiv:2411.14299*.

Chen, L.; Wu, Y.; Wen, C.; Wang, S.; Zhang, L.; Yu, B.; Sun, Q.; and Zhuo, C. 2024. An Agile Framework for Efficient LLM Accelerator Development and Model Inference. In *ICCAD*.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2022. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *arXiv preprint arXiv:2211.12588*.

Chen, X.; Zhou, S.; Liang, K.; Zhou, T.; and Liu, X. 2026. ImgCoT: Compressing Long Chain of Thought into Compact Visual Tokens for Efficient Reasoning of Large Language Models. In *ICML*.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Diao, X.; Liu, Z.; Zhang, C.; Wu, W.; Kong, K.; Shi, L.; Ding, K.; Vosoughi, S.; and Gui, J. 2026. Addressing Overthinking in Large Vision-Language Models via Gated Perception-Reasoning Optimization. *arXiv preprint arXiv:2601.04442*.

Du, C.; Niu, Y.; Shen, D.; and Xu, L. 2026. CoTZero: Annotation-Free Human-Like Vision Reasoning via Hierarchical Synthetic CoT. *arXiv preprint arXiv:2602.08339*.

Gan, Z.; Liao, Y.; and Liu, Y. 2025. Rethinking External Slow-Thinking: From Snowball Errors to Probability of Correct Reasoning. In *ICML*.

Ge, J.; Luo, H.; Qian, S.; Gan, Y.; Fu, J.; and Zhang, S. 2023. Chain of Thought Prompt Tuning in Vision Language Models. *arXiv preprint arXiv:2304.07919*.

Gu, J.; Hao, Y.; Wang, H. W.; Li, L.; Shieh, M. Q.; Choi, Y.; Krishna, R.; and Cheng, Y. 2026. ThinkMorph: Emergent Properties in Multimodal Interleaved Chain-of-Thought Reasoning. In *ICLR*.

Gui, R.; Wang, Z.; Wang, J.; Ma, C.; Zhen, H.; Yuan, M.; Hao, J.; Lian, D.; Chen, E.; and Wu, F. 2025. HyperTree Planning: Enhancing LLM Reasoning via Hierarchical Thinking. In *ICML*.

Hao, S.; Sukhbaatar, S.; Su, D.; Li, X.; Hu, Z.; Weston, J.; and Tian, Y. 2025. Training Large Language Models to Reason in a Continuous Latent Space. In *ICLR*.

He, H.; Geng, Z.; and Peng, Y. 2026. Fine-R1: Make Multimodal LLMs Excel in Fine-Grained Visual Recognition by Chain-of-Thought Reasoning. In *ICLR*.

Hu, L.; Ma, X.; Liao, Z.; and Liu, Y. 2026. TVI-CoT: Text-Visual Interleaved Chain-of-Thought Reasoning for Multimodal Understanding. In *ICML*.

Hua, J.; Yin, Y.; Wu, Y.; Wang, T.; Huang, Y.; and Liu, M. 2026. Unleashing Spatial Reasoning in Multimodal Large Language Models via Textual Representation Guided Reasoning. In *ACL*.

Huang, M.; Shi, Y.; Peng, D.; Lai, S.; Xie, Z.; and Jin, L. 2026. OCR-Reasoning Benchmark: Unveiling the True Capabilities of MLLMs in Complex Text-Rich Image Reasoning. In *ICLR*.

Jiang, S.; Liao, Y.; Zhang, Y.; Wang, Y.; and Wang, Y. 2026a. Overthinking Reduction with Decoupled Rewards and Curriculum Data Scheduling. In *ICLR*.

Jiang, Y.; Lu, X.; Jin, Q.; Sun, Q.; Wu, H.; and Zhuo, C. 2024. FabGPT: An Efficient Large Multimodal Model for Complex Wafer Defect Knowledge Queries. In *ICCAD*.

Jiang, Y.; Hu, Y.; Deng, J.; Qiu, X.; Cui, Y.; He, X.; Li, R.; Sun, Q.; and Zhuo, C. 2026b. CircuitThink: A Multimodal Reasoning Framework for Automated Circuit-to-Netlist Translation with Trajectory-Guided Reinforcement Learning. In *AAAI*.

Jin, B.; Xie, C.; Zhang, J.; Roy, K. K.; Zhang, Y.; Li, Z.; Li, R.; Tang, X.; Wang, S.; Meng, Y.; and Han, J. 2024. Graph Chain-of-Thought: Augmenting Large Language Models by Reasoning on Graphs. *arXiv preprint arXiv:2404.07103*.

Jin, Q.; Liu, Y.; Jiang, Y.; Sun, Q.; and Zhuo, C. 2025. UniTHO: A Unified Multi-Task Framework for Computational Lithography. In *ICCAD*.

Jin, Z.; Zhu, K.; Yuan, H.; Hao, Y.; Cao, P.; Chen, Y.; Liu, K.; and Zhao, J. 2026. Look Light, Think Heavy: What Multimodal Chain-of-Thought Reasoning Can and Cannot Do. *arXiv preprint arXiv:2606.22565*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models Are Zero-Shot Reasoners. *arXiv preprint arXiv:2205.11916*.

Kong, J.; Fang, H.; Liao, S.; Li, J.; Chen, B.; Wu, H.; Xia, S.-T.; and Zhang, M. 2026. Reasoning Matters: Mitigate Hallucination in Multimodal Large Reasoning Models via Reasoning-Conditioned Preference Optimization. *arXiv preprint arXiv:2605.27906*.

Li, A.; Wang, C. L.; Fu, D.; Yue, K.; Cai, Z.; Zhu, W. B.; Liu, O.; Guo, P.; Neiswanger, W.; Huang, F.; Goldstein, T.; and Goldblum, M. 2026a. Zebra-CoT: A Dataset for Interleaved Vision-Language Reasoning. In *ICLR*.

Li, L.; Wang, Z.; Wu, Y.; Cai, J.; and Yang, X. 2026b. CoT Vectors: Transferring and Probing the Reasoning Mechanisms of LLMs. In *ICLR*.

Li, X.; and Ma, J. 2025. AIM-CoT: Active Information-Driven Multimodal Chain-of-Thought for Vision-Language Reasoning. *arXiv preprint arXiv:2509.25699*.

Li, Y.; Chen, Z.; Wu, Z.; Zhou, K.; Luo, R.; Zhang, C.; He, Z.; Zhan, Y.; Zhao, W. X.; and Qiu, M. 2026c. Unleashing Perception-Time Scaling to Multimodal Reasoning Models. In *ICLR*.

Li, Z.; Ma, Z.; Li, M.; Li, S.; Rong, Y.; Xu, T.; Zhang, Z.; Zhao, D.; and Huang, W. 2026d. STAR-R1: Multi-View Spatial Transformation Reasoning by Reinforcing Multimodal LLMs. In *CVPR*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training Language Models to Follow Instructions with Human Feedback. In *NeurIPS*.

Qin, L.; Gong, J.; Sun, Y.; Li, T.; Pan, H.; Yang, M.; Yang, X.; Qu, C.; Tan, Z.; and Li, H. 2026. Uni-CoT: Towards Unified Chain-of-Thought Reasoning across Text and Vision. In *ICLR*.

Shao, H.; Qian, S.; Xiao, H.; Song, G.; Zong, Z.; Wang, L.; Liu, Y.; and Li, H. 2024a. Visual CoT: Advancing Multi-Modal Language Models with a Comprehensive Dataset and Benchmark for Chain-of-Thought Reasoning. *arXiv preprint arXiv:2403.16999*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024b. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Shen, X.; Wang, Y.; Zhou, Y.; Shi, X.; Zhao, P.; Wang, Y.; and Gu, J. 2025. Efficient Reasoning with Hidden Thinking. *arXiv preprint arXiv:2501.19201*.

Shi, Y.; Tao, Z.; Gao, Y.; Zhou, T.; Chang, C.; Wang, Y.; Chen, B.; Zhang, G.; Liu, A.; Yu, Z.; Lin, T.-J.; and He, L. 2024. AMSnet-KG: A Netlist Dataset for LLM-based AMS Circuit Auto-Design Using Knowledge Graph RAG. *ACM Transactions on Design Automation of Electronic Systems*.

Shi, Y.; Tao, Z.; Gao, Y.; Huang, L.; Wang, H.; Yu, Z.; Lin, T.-J.; and He, L. 2025a. AMSnet 2.0: A Large AMS Database with AI Segmentation for Net Detection. *arXiv preprint arXiv:2505.09155*.

Shi, Y.; Zhang, Z.; Wang, H.; Tao, Z.; Huang, L.; Li, Z.; Chen, B.; Wang, Y.; Huang, Z.; Liu, X.; Chen, Q.; Yu, Z.; Lin, T.-J.; and He, L. 2025b. AMSBench: A Comprehensive Benchmark for Evaluating MLLM Capabilities in AMS Circuits. *arXiv preprint arXiv:2505.24138*.

Tao, Z.; Shi, Y.; Huo, Y.; Ye, R.; Li, Z.; Huang, L.; Wu, C.; Bai, N.; Yu, Z.; Lin, T.-J.; and He, L. 2024. AMSNet: Netlist Dataset for AMS Circuits. In *IEEE LLM Aided Design Workshop*.

Tian, X.; Zou, S.; Yang, Z.; He, M.; Waschkowski, F.; Wesemann, L.; Tu, P.; and Zhang, J. 2026. More Thought, Less Accuracy? On the Dual Nature of Reasoning in Vision-Language Models. In *ICLR*.

von Recum, A.; Girrbach, L.; and Akata, Z. 2026. Are Reasoning LLMs Robust to Interventions on Their Chain-of-Thought? In *ICLR*.

Wang, H.; Xu, Q.; Wang, C.; Xue, T.; Peng, C.; Chen, W.; and Lin, F. 2026a. Bad Seeing or Bad Thinking? Rewarding Perception for Vision-Language Reasoning. In *ICML*.

Wang, W.; Gao, Z.; Chen, L.; Chen, Z.; Zhu, J.; Zhao, X.; Liu, Y.; Cao, Y.; Ye, S.; Zhu, X.; Lu, L.; Duan, H.; Qiao, Y.; Dai, J.; and Wang, W. 2026b. VisualPRM400K: An Effective Dataset for Training Multimodal Process Reward Models. In *ICLR*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *ICLR*.

Wang, Y.; Wu, S.; Zhang, Y.; Yan, S.; Liu, Z.; Luo, J.; and Fei, H. 2025. Multimodal Chain-of-Thought Reasoning: A Comprehensive Survey. *arXiv preprint arXiv:2503.12605*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*.

Wei, X.; Liu, X.; Zang, Y.; Dong, X.; Cao, Y.; Wang, J.; Qiu, X.; and Lin, D. 2026. SIM-CoT: Supervised Implicit Chain-of-Thought. In *ICLR*.

Wu, J.; Zhang, X.; Yuan, H.; Zhang, X.; Huang, T.; He, C.; Deng, C.; Zhang, R.; Wu, Y.; and Long, M. 2026a. Visual Generation Unlocks Human-Like Reasoning through Multimodal World Models. In *ICLR Workshop on World Models*.

Wu, J.; Chen, H.; Tu, H.; Tang, X.; Shi, F.; Liu, H.; Lu, H.; Xie, C.; and Zhou, Y. 2026b. From Seeing to Thinking: Decoupling Perception and Reasoning Improves Post-Training of Vision-Language Models. In *ICML*.

Wu, Q.; Yang, X.; Zhou, Y.; Fang, C.; Song, B.; Sun, X.; and Ji, R. 2026c. Grounded Chain-of-Thought for Multimodal Large Language Models. In *CVPR*.

Wu, T.; Lan, J.; Yuan, W.; Jiao, J.; Weston, J.; and Sukhbaatar, S. 2025. Thinking LLMs: General Instruction Following with Thought Generation. In *ICML*.

Xia, J.; Tong, B.; Zang, Y.; Shao, R.; and Zhou, K. 2025. Bootstrapping Grounded Chain-of-Thought in Multimodal LLMs for Data-Efficient Model Adaptation. In *ICCV*.

Xiao, T.; Xu, X.; Huang, Z.; Gao, H.; Liu, Q.; Liu, Q.; and Chen, E. 2026. Perception-R1: Advancing Multimodal Reasoning Capabilities of MLLMs via Visual Perception Reward. In *ICLR*.

Xu, G.; Jin, P.; Li, H.; Song, Y.; Sun, L.; and Yuan, L. 2024. LLaVA-CoT: Let Vision Language Models Reason Step-by-Step. *arXiv preprint arXiv:2411.10440*.

Yamin, K.; Gupta, S.; Ghosal, G.; Lipton, Z.; and Wilder, B. 2026. LLMs Struggle to Balance Reasoning and World Knowledge in Causal Narrative Understanding. In *ICLR*.

Yang, J.; Yang, S.; Gupta, A. W.; Han, R.; Fei-Fei, L.; and Xie, S. 2025. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall Spaces. In *CVPR*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint arXiv:2305.10601*.

Yao, X.; Ren, R.; Liao, Y.; Ding, L.; and Liu, Y. 2026. Compositional Generalization from Learned Skills via CoT Training: A Theoretical and Structural Analysis for Reasoning. In *ICLR*.

Ye, X.; Mao, Y.; Zhang, J.; Liu, Y.; Li, H.; Wu, F.; Li, Z.; Liao, Y.; Wang, Z.; Yin, Z.; Yuan, L.; Zeng, X.; Liu, Z.; Wang, M.; Cong, L.; Torr, P.; Sun, H.; Gao, S.; and Tang, X. 2026. LatentChem: From Textual CoT to Latent Thinking in Chemical Reasoning. In *ICLR Workshop on Latent and Implicit Thinking*.

Yeo, E.; Tong, Y.; Niu, X.; Neubig, G.; and Yue, X. 2025. Demystifying Long Chain-of-Thought Reasoning in LLMs. In *ICLR Workshop on Data Problems for Foundation Models*.

Yin, B.-W.; Yang, Q.; Sun, B.; Wei, X.; and Hou, Q. 2026. Discriminative Visual Process Rewards for Scaling Thinking at Test-Time with Images. In *ICML*.

Yu, Q.; Weber, M.; Deng, X.; Shen, X.; Cremers, D.; and Chen, L.-C. 2024. An Image Is Worth 32 Tokens for Reconstruction and Generation. In *ECCV*.

Zhang, J.; Tong, J.; Lin, J.; Wu, H.; Sun, Y.; Ma, Y.; and Shen, X. 2026a. Think-as-You-See: Streaming Chain-of-Thought Reasoning for Large Vision-Language Models. *arXiv preprint arXiv:2603.02872*.

Zhang, J.; Zhang, C.; Chen, S.; Wang, X.; Huang, Z.; Zheng, P.; Yuan, S.; Zheng, S.; Sun, Q.; Zou, J.; Lee, L.-H.; and Yang, Y. 2026b. Learning Global Hypothesis Space for Enhancing Synergistic Reasoning Chain. In *ICLR*.

Zhang, K.; Hua, Z.; Li, Y.; Chen, Y.; and Zhou, Y. 2022. AMS-Net: Adaptive Multi-Scale Network for Image Compressive Sensing. *IEEE Transactions on Multimedia*, 25:5676–5689.

Zhang, R.; Zhang, B.; Li, Y.; Zhang, H.; Sun, Z.; Gan, Z.; Yang, Y.; Pang, R.; and Yang, Y. 2024. Improve Vision Language Model Chain-of-Thought Reasoning. *arXiv preprint arXiv:2410.16198*.

Zhang, X.; Jia, C.; Li, S.; He, D.; Xiong, X.; Sun, Z.; He, H.; Wu, Y.; Yu, B.; Sun, L.; Tan, C.; and Wei, J. 2026c. How Reinforcement Learning Unlocks the Aha Moment in Geometric Interleaved Reasoning. In *ICML*.

Zhang, Y.; Wu, Z.; Zhao, J.; Fang, J.; Li, Z.; Wang, K.; Wen, Q.; and Shao, Y. 2026d. See First, Reason Later: Mutual Information-Guided Reinforcement Learning for Vision-Language Models. In *ICML*.

Zhao, Q.; Lu, Y.; Kim, M. J.; Fu, Z.; Zhang, Z.; Wu, Y.; Li, Z.; Ma, Q.; Han, S.; Finn, C.; Handa, A.; Lin, T.-Y.; Wetzstein, G.; Liu, M.-Y.; and Xiang, D. 2025. CoT-VLA: Visual Chain-of-Thought Reasoning for Vision-Language-Action Models. In *CVPR*.

Zhu, M.; Zhong, H.; Zhao, C.; Du, Z.; Liu, M.; Huang, Z.; Li, A.; Chen, H.; Zou, C.; Chen, J.; Yang, M.; and Shen, C. 2026. ACTIVE-o3: Empowering Multimodal Large Language Models with Active Perception via Pure Reinforcement Learning. In *ICML*.

A AI USAGE DISCLOSURE

AI-assisted tools were used during manuscript preparation to improve the clarity, fluency, and presentation of the writing. All AI-assisted revisions were reviewed, edited, and verified by the authors, who take full responsibility for the manuscript and its scientific claims.

B CONNECTIVITY BOTTLENECK ANALYSIS: PROMPT AND EVALUATION PROTOCOL

B.1 EVALUATION PROMPT

We use the same prompt for Gemini-3.1-Pro, Kimi-K3, and GPT-5.5 in the connectivity bottleneck analysis. Each model receives only the unannotated circuit schematic together with the following system and user prompts.

System prompt.

```
You are an expert in circuit schematics.Return only the requested <answer> JSON.
```

User prompt.

```
Analyze the circuit schematic and return three outputs:
1. Every external port with its bounding box.
2. Every physical circuit device with its type and bounding box.
3. A complete SPICE netlist.

Bounding-box rules:
- bbox order is [x1,y1,x2,y2], with (0,0) at the top-left.
- A device bbox must tightly enclose the complete graphical
  device symbol, including its body, plates, and terminal
  strokes. Do not box only a center line, arrow, or text label,
  and do not include the nearby instance name.
- An external-port bbox must enclose the terminal marker or
  hollow endpoint, not the adjacent signal-name text.

SPICE rules:
- Output one line for every physical device.
- MOS: Mname drain_node gate_node source_node bulk_node
  NMOS|PMOS
- Resistor: Rname node_a node_b [optional value]
- Capacitor: Cname node_a node_b [optional value]
- Inductor: Lname node_a node_b [optional value]
- Voltage source: Vname positive_node negative_node
  [optional value]
- Current source: Iname positive_node negative_node
  [optional value]
- Use node 0 for ground. Use VDD for an explicit VDD rail.
- Do not output .MODEL, .OP, .END, comments, Markdown, or
  explanations.

Return exactly this schema:
<answer>
{
  "external_ports": [
    {"id":"P1","type":"PORT","bbox":[x1,y1,x2,y2]}
  ],
  "devices": [
    {"id":"M1","type":"NMOS","bbox":[x1,y1,x2,y2]}
```

```
  ],
  "netlist": [
    "M1 node1 node2 0 0 NMOS"
  ]
}
</answer>
```

B.2 Entity-Correct Evaluation Protocol

A prediction is considered entity-correct only when all devices and ordinary external ports are successfully matched, with no missing or extra entities. We use maximum-weight one-to-one matching, so each predicted entity and each reference entity can participate in at most one match. Device matches require identical types and a bounding-box IoU of at least 0.5. External-port matches require identical types and either a bounding-box IoU above 0.5 or a center distance below 20 in the normalized 0–1000 coordinate system. Predicted entity names are not used for matching. External-port markers generally occupy small image regions, so minor localization shifts can cause a disproportionately large drop in IoU even when the predicted center is accurate. The center-distance criterion therefore provides tolerance to small variations in bounding-box extent while preserving spatial specificity.

B.3 Additional Connectivity and Circuit-Rule Failure Examples

Figure 9 presents two prediction-only visualizations with a unified annotation style. In the Gemini-3.1-Pro example, the predicted `net3` contains only one endpoint, creating a dangling connection detectable without the ground-truth topology. The GPT-5.5 panel is a controlled case-study visualization in which two declared NMOS gate terminals are omitted from all predicted nets, illustrating the missing-endpoint rule.

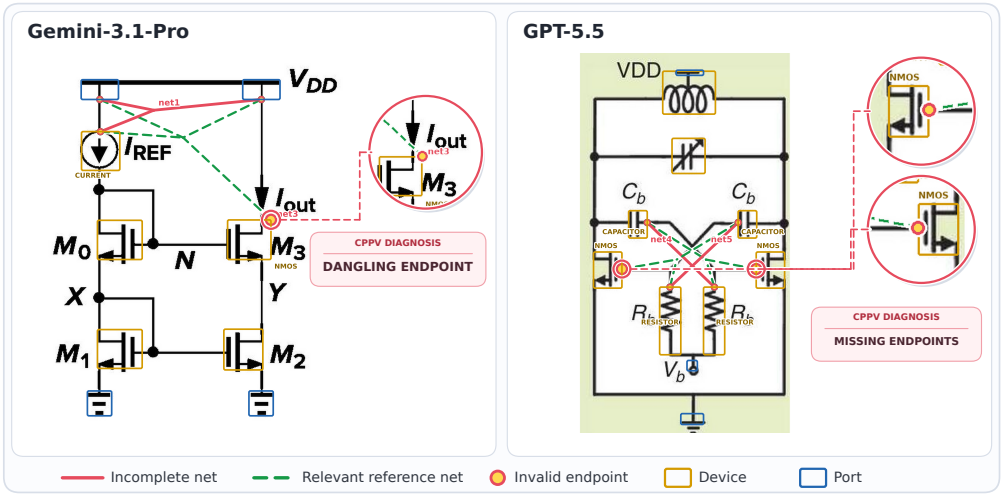

Figure 9: Additional qualitative examples of dangling and missing endpoints.

C Experiments

C.1 Implementation Details

We initialize Circuit-R1 from Qwen3-VL-4B and use full-parameter optimization in both training stages. SFT is implemented with LLaMA-Factory, GRPO is implemented with EasyR1, and all experiments are conducted on four NVIDIA A800 GPUs. Tables 6 and 7 report the detailed configurations.

Configuration	Value	Configuration	Value
Base model	Qwen3-VL-4B	Training samples	1,000
Training type	Full-parameter	Training epochs	2
Global batch size	8	Maximum sequence length	8,192
Learning rate	1×10^{-6}	Optimizer	8-bit AdamW
LR scheduler	Cosine	Warmup ratio	0.1
Weight decay	0.01	Precision	BF16
Distributed strategy	DeepSpeed ZeRO-3	Maximum image pixels	589,824

Table 6: Implementation details of supervised fine-tuning.

Configuration	Value	Configuration	Value
Initialization	Cold-start SFT	Training samples	2,014
Training epochs	5	Rollouts per prompt	8
Rollout batch size	16	Actor global batch size	16
Maximum prompt length	1,024	Maximum response length	5,120
Learning rate	5×10^{-7}	Optimizer	AdamW
LR scheduler	Cosine	Weight decay	0.01
KL coefficient	0.001	Maximum gradient norm	1.0
Sampling temperature	0.8	Top-p	0.95

Table 7: Implementation details of GRPO.

C.2 Sensitivity to Reward Weights

We study the sensitivity of CIRCUIT-R1 to the coefficients in R_{SGR} and R_{base}. We vary one reward family at a time rather than evaluating their Cartesian product. The coefficients in each configuration sum to one, and all other training settings, including the CPPV penalty, are held fixed. Because the scalar reward itself changes across configurations, we compare the resulting task metrics rather than the aggregate training reward. All metrics are reported as percentages on the same benchmark split.

C.2.1 Weights in the Stepwise Grounded Reasoning Reward

We first vary $(\lambda_{\mathrm{port}}, \lambda_{\mathrm{device}}, \lambda_{\mathrm{visual}}, \lambda_{\mathrm{conn}})$ in R_{SGR} while keeping all other settings fixed. The evaluated configurations and results are shown in Table 8.

C.2.2 Weights in the Base Reward

We next vary $(\alpha, \beta, \gamma, \delta)$, corresponding to R_{SGR}, R_{binding}, R_{answer}, and R_{format} in R_{base}. The weights within R_{SGR} and all other settings are kept fixed. The evaluated configurations and results are shown in Table 9.

$(\lambda_p, \lambda_d, \lambda_v, \lambda_c)$	Ext.-Port F1	Device F1	Ans. Acc.
$(0.25, 0.25, 0.25, 0.25)$	96.94	95.99	73.82
$(0.30, 0.30, 0.20, 0.20)$	96.96	96.34	74.27
$(0.20, 0.20, 0.40, 0.20)$	97.06	95.82	75.58
$(0.20, 0.20, 0.20, 0.40)$	96.06	95.75	75.41
$(0.20, 0.20, 0.30, 0.30)$	**96.16**	**96.31**	**76.60**

Table 8: Sensitivity to the weights in R_{SGR}.

$(\alpha, \beta, \gamma, \delta)$	Format Valid	Ans. Acc.
$(0.70, 0.10, 0.15, 0.05)$	100	75.59
$(0.50, 0.30, 0.15, 0.05)$	99.5	74.17
$(0.50, 0.15, 0.30, 0.05)$	99.57	75.79
$(0.55, 0.15, 0.20, 0.10)$	100	73.33
$(0.60, 0.15, 0.20, 0.05)$	**99.86**	**76.60**

Table 9: Sensitivity to the weights in R_{base}.

C.3 Continuous Entity-Localization Reward

We compare our continuous localization reward with a thresholded alternative in Circuit-Think. For a type-compatible device or external port, the thresholded variant assigns fixed spatial credit once the bounding-box IoU reaches 0.5, whereas our design uses the IoU value itself. Both variants use the same one-to-one entity matching, cardinality normalization, data, and other training settings.

Entity-localization reward	Device F1	External-Port F1	Answer Acc.
Thresholded IoU ($\mathrm{IoU} \geq 0.5$)	85.65	83.79	58.54
Continuous IoU (Ours)	**93.38**	**92.32**	**62.89**

Table 10: Ablation of thresholded and continuous localization rewards for devices and external ports. All other reward components are held fixed.

As shown in Table 10, replacing the thresholded objective with continuous IoU improves Device F1 from 85.65 to 93.38, External-Port F1 from 83.79 to 92.32, and Answer Accuracy from 58.54 to 62.89. In the thresholded design, a correct type already contributes half of the entity reward, even when the predicted box is substantially displaced. Moreover, boxes below the IoU threshold receive no progressively increasing localization signal. The model can therefore obtain considerable credit from type recognition while learning comparatively coarse spatial grounding. The continuous objective instead distinguishes different degrees of overlap and rewards every improvement in localization. The 4.35-point gain in Answer Accuracy further indicates that more accurate entity localization propagates to terminal association and final topology recovery.