Circuit-Think: A Multimodal Reasoning Framework for Automated Circuit-to-Netlist Translation with Trajectory-Guided Reinforcement Learning

Yuqi Jiang[1], **Yupeng Hu**[1], **Jinyuan Deng**[1], **Xiaotian Qiu**[1, 2], **Yucheng Cui**[1], **Xuyang He**[1], **Ruidong Li**[3], **Qi Sun**[1*], **Cheng Zhuo**[1]
[1]Zhejiang University, Hangzhou, China,
[2]Shanghai Innovation Institute, Shanghai, China,
[3]Shandong Yunhai Guochuang Cloud Computing Equipment Industry Innovation Co., Ltd., Jinan, China,
qisunchn@zju.edu.cn

Abstract

Vision Language Models (VLMs) have shown strong performance in multimodal understanding, offering promise for the circuit-to-netlist translation task. However, the diverse component symbols and complex connections in circuit images challenge VLMs in understanding physical layouts and reasoning for electrical connection logic. To address these, we propose **Circuit-Think**, the first multimodal reasoning framework for the automated circuit-to-netlist translation task, which employs a **Trajectory-Guided Reinforcement Learning (TGRL)** paradigm for structured logical reasoning on circuit images. Circuit-Think initializes reasoning capabilities through supervised fine-tuning (SFT) on image-netlist pairs, then optimizes reasoning trajectories and netlist generation decisions using TGRL. Firstly, TGRL introduces a step-by-step reasoning paradigm, which guides the model with stepwise reward functions to simulate the human cognitive trajectory of "identifying ports, recognizing devices, and inferring connections". Secondly, we customize a multi-level reward that maps reasoning and answers into graph structures and node sets, jointly optimizing logical consistency and netlist accuracy via graph similarity and set matching. Thirdly, TGRL contains a reflective learning mechanism for low-scoring samples, which corrects the reasoning trajectory through reference answers as hints, avoiding local optima caused by sparse reward signals or erroneous reasoning paths. Moreover, we construct a circuit image-netlist reasoning dataset with 3,100 samples, offering step-by-step annotations for converting circuit images to netlists. Extensive experiments demonstrate that Circuit-Think achieves SOTA netlist accuracy and significantly improves the accuracy of downstream tasks.

Datasets — https://github.com/7jiangyq/CircuitThink

Introduction

The circuit-to-netlist translation task aims to extract electrical components and interconnections from circuit images and convert them into a standardized SPICE netlist format. Unlike the visual representation of circuit images, the SPICE netlist (Bhandari et al. 2024) structurally describes component types, connections, and electrical properties, eliminating the layout complexity and symbol diversity inherent in images. Therefore, by introducing the SPICE netlist as a unified circuit representation, it provides readable input for downstream tasks, such as circuit QA and optimization (Shi et al. 2024; Jiang et al. 2025), thereby reducing the interpretive bias caused by differentiated visual layouts.

In recent years, many methods have defined the circuit-to-netlist translation task as an image description problem. Computer vision-based methods (Zhang et al. 2022; Shi et al. 2025a; Jin et al. 2025) rely on fixed rules and algorithms to identify component types and connections, but their lack of a global layout perspective makes it difficult to handle scenarios such as component occlusion and symbol variations. Traditional Vision Language Models (VLMs)-based methods (Chen et al. 2024; Jiang et al. 2024) endow them with cross-modal semantic understanding through supervised fine-tuning (SFT) on circuit data, enhancing robustness in recognizing unconventional connection structures. However, this method treats reasoning as a static mapping from circuit images to SPICE netlists, neglecting the dynamic logical constraints and structural verifiability required for circuit decision-making. Therefore, when dealing with circuits involving multi-component functional collaboration, they fail to dynamically model the logical relationships within the circuit layout, limiting their generalization.

Reasoning VLMs are expected to become powerful tools for efficient inference in circuit-to-netlist translation tasks. Their advanced reasoning capabilities are attributed to the integration of Chain-of-Thought (CoT) (Ge et al. 2023; Zhang et al. 2024; Xu et al. 2024), which guides the gradual unfolding of the reasoning process to solve problems, thereby suppressing overfitting to the training data. Despite these advancements, the current training paradigm, such as Group Relative Policy Optimization (GRPO) (Guo et al. 2025), implicitly constrains the reasoning process with answer correctness, which often results in erroneous or self-contradictory reasoning logic, as shown in Figure 1. Particularly when dealing with real-world circuit scenarios, misunderstandings or omissions in node connections divert reasoning from the correct path. Even if the answer is correct by chance, the disconnection between reasoning and answer reduces interpretability. Furthermore, most existing paradigms cannot locate and correct reasoning trajectories, preventing the tracing and adjustment of biases, thus limiting their generalization ability in large-scale complex circuits.

*Corresponding author.

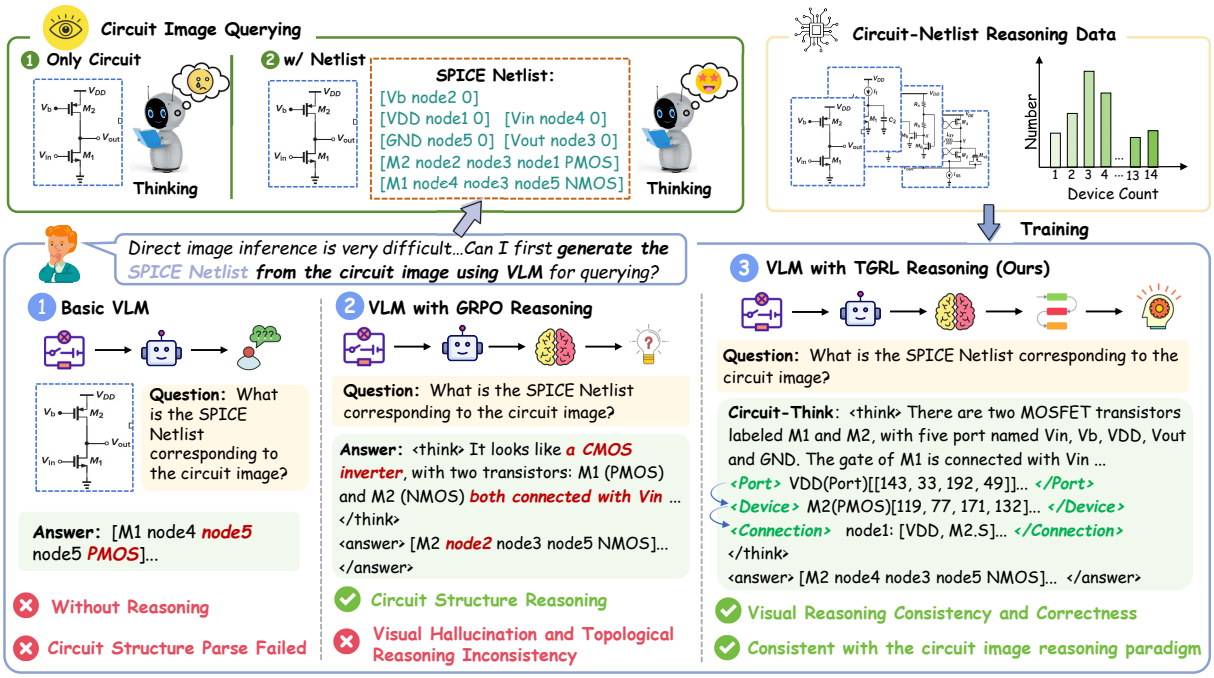

Figure 1: Motivation and advantages of our proposed framework. The top-left corner shows the assistance of SPICE netlist hints in answering circuit image questions. The top-right corner presents our constructed circuit image-netlist dataset and its difficulty distribution. Below are three VLM models: one without reasoning ability, one trained with Group Relative Policy Optimization (GRPO), and one trained with TGRL. Results show that our framework outperforms others in circuit image reasoning.

To overcome these, we propose **Circuit-Think**, the first reasoning VLM framework tailored for automated circuit-to-netlist translation, which is designed to generate structured SPICE netlists from complex circuit images. The framework first employs SFT to warm up the model's reasoning ability, and then adopts the pioneering **Trajectory-Guided Reinforcement Learning (TGRL)** learning paradigm to constrain reasoning logic, generate accurate SPICE netlists, and correct reasoning biases. To activate TGRL, we first introduce a step-by-step reasoning paradigm that sequentially identifies ports, devices and connections. And a stepwise reasoning reward function that utilizes type-region recognition and list element matching is customized to constrain the direction and quality of this reasoning trajectory. Secondly, we develop a netlist accuracy reward based on node set matching, and a graph consistency reward that compares the similarity between reasoning and answer. These reward signals maximize the cumulative precision feedback and guide the model toward high-accuracy SPICE netlists and reasoning paths aligned with ground truth logic. Lastly, to improve the stability and efficiency of TGRL, a reflective learning mechanism adjusts the reward values by actively incorporating reference answers, facilitating the model to converge faster and improve decision-making trajectories in sparse reward signals.

We construct a reasoning dataset with 3,100 circuit image-netlist pairs to support our framework training. It includes images of varying complexity, with reasoning involving ports, device positions, connections, and the corresponding SPICE netlist topology, as shown in Figure 2. Benchmark results show that Circuit-Think, trained with minimal data, outperforms existing models in SPICE netlist accuracy across multiple datasets, demonstrating strong generalization. Our contributions are as follows:

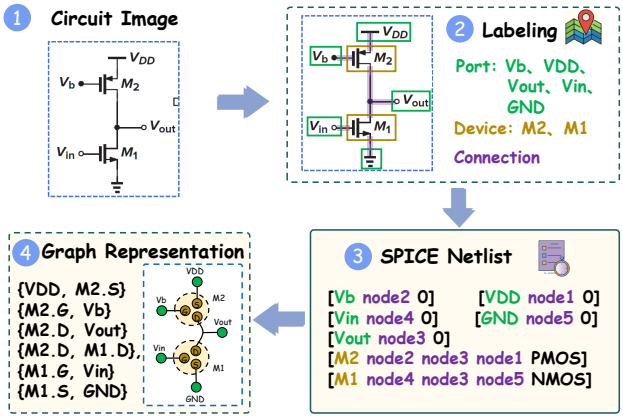

Figure 2: Given a circuit image, the ports, devices, and connections are identified to derive the corresponding SPICE netlist. Each netlist can be uniquely represented as a graph.

- We propose Circuit-Think, the first multimodal reasoning framework based on the Trajectory-Guided Reinforcement Learning training paradigm, applying step-by-step reasoning to automated circuit-to-netlist translation.

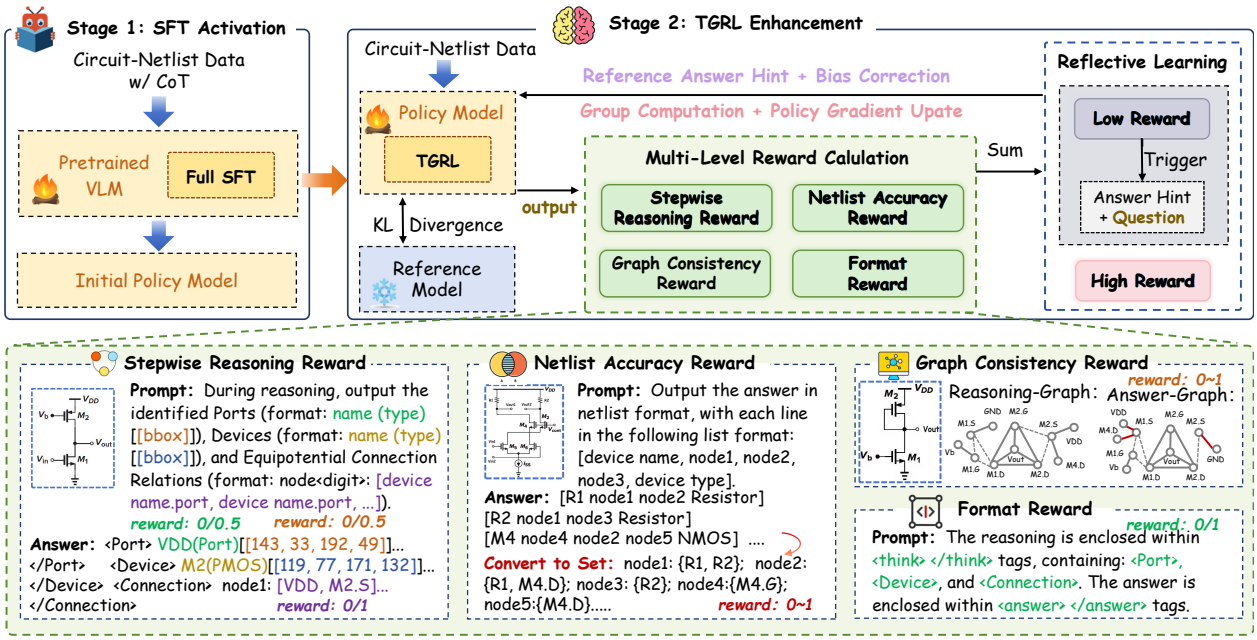

Figure 3: The architecture of Circuit-Think. Pre-train the model with SFT, then apply TGRL to enhance circuit-to-netlist translation abilities through the circuit-specific reward and reflective learning mechanism. Present scores for four reward functions.

- We tailor a multi-level reward mechanism for circuits that progressively evaluates reasoning direction, netlist quality, and logical consistency, filling the gap in dynamic reasoning optimization.
- We introduce reflective learning that corrects reasoning biases through reward feedback and answer hints, handling the challenge of tracing visual reasoning.
- We construct a circuit image-netlist reasoning dataset, improving the circuit-to-netlist translation accuracy by 43.79% and boosting performance in downstream multi-scenario QA tasks by 19.44%.

Preliminaries

Dataset Construction

We systematically construct a dataset of 3,100 high-quality circuit image-netlist pairs from public sources (Gao et al. 2025), covering analog, digital, and mixed-signal designs ranging from simple to complex systems. Subsequently, components within these images were precisely annotated, and the YOLO11 model (Jocher and Qiu 2024) was trained on this annotated dataset for automated component detection and bounding box annotation. Given the complexity of circuit topologies and connections, we incorporate human expert intervention to ensure accuracy. Specifically, experts establish correct connection relationships based on electrical principles and generate their SPICE netlists. Each data entry follows a standardized JSON, containing three main reasoning steps, and undergoes rigorous quality control.

AI for Circuit-to-Netlist Translation

Circuit images contain component layouts and connections, while SPICE netlists provide a more structured description. With deep learning advancements, many studies explore converting circuit images into SPICE netlists for analysis. For example, Hemker et al.(Hemker et al. 2024) decompose circuit images into component detection, wiring detection, and text recognition to convert them into SPICE netlists; Tao et al.(Tao et al. 2024) use a data-driven approach to map circuit images to netlists. However, existing models struggle to understand and reason about complex topologies, particularly when handling circuit images with interwoven elements and intricate connections.

VLMs for Circuit Image Analysis

VLMs demonstrate exceptional multimodal understanding and reasoning capabilities in image analysis tasks. These models combine image and text information, enabling them to parse complex components and connections in circuit images effectively. For example, MAPS (Zhu et al. 2025) combines a physical perception model and simulation process, enhancing VLM's ability in the physical domain through SFT. Auto-SPICE (Bhandari et al. 2024) improves the accuracy of analog circuit netlist generation through circuit labeling, prompt tuning, and netlist verification. However, current VLMs primarily depend on surface image information and lack effective reasoning, often overlooking implicit logic and structural patterns in circuit images, resulting in inaccuracies and inconsistencies in netlist generation.

Our Framework

Figure 3 illustrates the two-stage training pipeline of Circuit-Think. Firstly, Qwen2.5-VL (Bai et al. 2025) is fine-tuned via SFT to enhance reasoning capabilities; in the second stage, TGRL is applied to improve circuit image analysis

and SPICE netlist generation. Next, we provide a detailed explanation of the TGRL learning paradigm.

TGRL: Step-by-Step Reasoning Paradigm

Since the same SPICE netlist may correspond to multiple circuit image layouts, the circuit-to-netlist translation task is inherently a multistep constraint satisfaction problem. Therefore, we propose a step-by-step reasoning paradigm based on stepwise decision-making through "port-device-connection" decoupling, simulating the human cognitive logic of "locating interfaces-identifying components-establishing associations" when analyzing circuit layouts, as shown in Figure 4. This paradigm follows a clear, structured reasoning path that reduces the difficulty of joint multi-attribute learning while preserving the space to explore alternative reasoning pathways.

Specifically, during the reasoning process, the model is required to identify the names, types, and relative coordinates of the ports and devices. Based on the results of these two steps, the model establishes the electrical connections between the device ports. To prevent the continuation of erroneous reasoning paths, if the reward score for any step falls below a predefined threshold, subsequent reasoning steps will not receive any rewards. This dynamic reasoning mechanism can be represented as:

$$\begin{aligned} \text{State}_t &= g(\sum_{i=1}^{t-1} \text{State}_i), \\ R_{\text{think}(t)} &= \begin{cases} f_{\text{think}(t)}(\text{State}_t), & \text{if } R_{\text{think}(t-1)} > \tau_1 \\ 0, & \text{if } R_{\text{think}(t-1)} < \tau_1 \end{cases}, \end{aligned} \tag{1}$$

where State_t is the reasoning state at step $t \in \{0, 1, 2, 3\}$, $g(\cdot)$ and $f_{\text{think}(t)}(\cdot)$ are the reasoning state update and reward functions for step t, τ_1 is the threshold that changes with training steps, starting at 0.5 and gradually increasing to 0.8, and $R_{\text{think}(t)}$ is the reward score for the reasoning step t.

TGRL: Multi-level Reward Mechanism

Traditional single-reward mechanisms typically focus only on the final SPICE netlist quality, neglecting reasoning quality and logical consistency. This may lead to the "gambling optimization" strategy, where the model improves the SPICE netlist quality through random simplifications rather than stable optimization, thereby disrupting the learning of the circuit's physical and electrical properties and reducing the interpretability of the reasoning process. Given that the reasoning chain may be logically inconsistent with the prompt or the final SPICE netlist, this bias exacerbates error accumulation. So we propose a multi-level reward mechanism that decomposes the global reward signal into three fine-grained rewards across "reasoning-answers-logic." We define the multi-level reward score as:

$$R_{\text{total}} = \alpha R_{\text{think}} + \beta R_{\text{answer}} + \gamma R_{\text{logic}} + \delta R_{\text{format}}, \tag{2}$$

where α, β, γ and δ are four parameters set to 0.2, 0.4, 0.2, and 0.2, respectively.

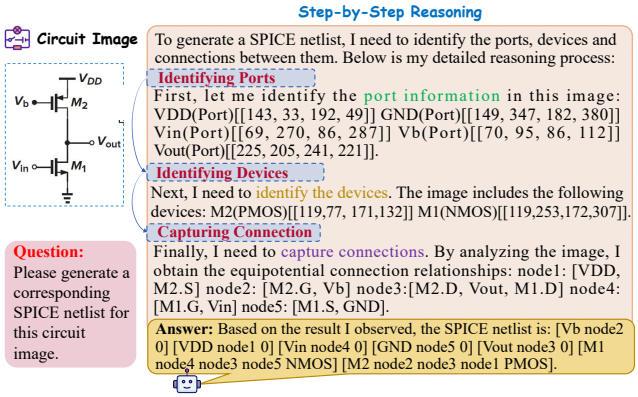

Figure 4: An example of step-by-step reasoning for circuit image to netlist translation.

Stepwise Reasoning Reward (R_{think}) Based on the step-by-step reasoning paradigm, we score the three reasoning steps using structured measurement criteria to quantify the compliance of the reasoning chain. In the port and device recognition stage, the unified format "name (type) [[bbox]]" is adopted, with dual rewards for semantic and geometric space. The semantic reward is awarded 0.5 points for matching the same "name (type)," and the geometric reward is given 0.5 points for calculating the mean Intersection over Union (mIoU $\geq$ 0.5) of normalized bounding boxes, with each total score being 1 point. In the connection relationship validation stage, the connections are described in the format "node<ID>: [device_name.port, . . .]," which is treated as a list of device-port pairs. We design a list reward mechanism, scoring 1 point by matching the list length and aligning the elements. All scores range from 0 to 1. This reward enhances the VLM's spatial understanding and evaluates the reasoning chain's correctness.

Netlist Accuracy Reward (R_{answer}) To constrain the topological accuracy of the generated SPICE netlist and provide more stable reward signals for small-scale sets, we construct a continuous reward function based on set matching. This method applies regularization constraints to map the list of SPICE netlist (e.g., "[VDD node1], [Vb node2], [R1 node1 node2 Resistor]") into the connection sets between devices and ports (e.g., "(VDD, R1), (Vb, R1)"). The reward is derived by comparing the proportion of equivalent connections between the ground truth SPICE netlist set C_{gt} and the generated SPICE netlist set C_{pred}, thus rewarding the correctly matched node sets in the generated SPICE netlist. The reward score can be expressed as:

$$R_{\text{answer}} = \frac{2 \times |C_{\text{gt}} \cap C_{\text{pred}}|}{|C_{\text{gt}}| + |C_{\text{pred}}|}, \tag{3}$$

where $\cap$ is the intersection.

Graph Consistency Reward (R_{logic}) Since the generated reasoning chain differs in format from the SPICE netlist and cannot be directly evaluated, we use a unified graph structure to quantify their similarity, which enforces the logical consistency between reasoning and answers. Specifically, we

employ the Depth-First Search (DFS) algorithm to extract the SPICE netlist and reasoning connection graphs. To eliminate node naming discrepancies (e.g., mapping "node1" in the SPICE netlist to "node5" in the reasoning graph as equivalent nodes), we compare the adjacency relationships of nodes to determine their equivalence. Even if node names differ, they can be considered equivalent as long as they connect to the same node type and share the same connectivity pattern. In the graph, each endpoint represents a port or device in the circuit image, while edges represent the connectivity between endpoints. By comparing the overlapping portions of corresponding edges in the reasoning connection graph G and the SPICE netlist graph $\hat{G}$, we compute the similarity and provide a reward. The reward score can be expressed as:

$$R_{\text{logic}} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{I}(G_{e_i} = \hat{G}_{e_j})}{\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{I}(G_{e_i} = \hat{G}_{e_j}) + \mathbb{I}(G_{e_i} \neq \hat{G}_{e_j})}, \quad (4)$$

where e_i and e_j are the i-th and j-th edges in graphs G and $\hat{G}$, respectively, and $\mathbb{I}$ is the indicator function used to determine whether the two edges are equal or not.

Format Reward (R_{format}) This reward encourages the model to adopt a structured reasoning process, requiring it to output reasoning steps within the <think> and </think> tags, with the reasoning content explicitly including the <port> and </port>, <device> and </device>, and <connection> and </connection> tags. The final answer is between the <answer> and </answer> tags.

$$R_{\text{format}} = \begin{cases} 1, & \text{Format Correct,} \\ 0, & \text{Otherwise.} \end{cases} \quad (5)$$

TGRL: Reflective Learning Mechanism

In the circuit-to-netlist translation task, we observe that the model receives low reasoning rewards during the initial exploration stage. Due to sparse reward feedback and error accumulation, the model struggles to receive sufficient positive feedback and is prone to getting stuck in local optima or halting learning. To address this issue, we propose a reflective learning that, when all reward values R_{total} fall below a certain threshold, actively provides reference answers as hints for low-reward samples, encouraging the model to reverse-engineer the reasoning trajectory based on the connection relationships and electrical properties in the SPICE netlist. Based on the improved reward results, we adjust the final score by calculating the difference between the reflective and original rewards, granting additional rewards only when a positive improvement occurs. If no positive improvement is made, a relative penalty is introduced to guide the model toward further optimizing the reasoning process. The final reward score is:

$$\hat{R}_{\text{total}} = \begin{cases} R_{\text{total}}, & \text{if } R_{\text{total}} > \tau_2 \\ R_{\text{total}} + \lambda_{\text{ref}} \cdot \frac{\max(0, R_{\text{total,hint}} - R_{\text{total}})}{1 - R_{\text{total}} + \epsilon}, & \text{if } R_{\text{total}} \leq \tau_2 \end{cases}, \quad (6)$$

where $R_{\text{total,hint}}$ is the reflective reward from the reference answer, λ_{ref} is the reference reward weight, set to 0.6, ϵ is the stability constant, set to 1e-6, and τ_2 is the threshold, set to 0.7.

Methods	Training Steps			
	50	100	150	200
SFT	37.13	36.85	36.01	34.78 (+0.00%)
DPO	35.30	36.72	39.58	42.15 (+21.19%)
PPO	35.16	37.14	38.63	40.09 (+15.27%)
GRPO	35.94	37.02	38.81	40.23 (+15.67%)
TGRL	**38.73**	**47.51**	**54.36**	**60.17 (+73.00%)**

Table 1: The performance of Qwen2.5-VL 7B on our dataset with TGRL *v.s.* other tuning methods.

TGRL: Overall Optimization Objective

The TGRL training paradigm in Circuit-Think optimizes the reasoning trajectory and answer accuracy based on relative differences within groups. Given a query x, generate N samples $\{\hat{O}_i\}_{i=1}^{N}$ from the old policy $\pi_{\theta_{\text{old}}}$, TGRL objective is:

$$\mathcal{J}_{\text{TGRL}}(\theta) = \mathbb{E}_{x,\{\hat{o}_i\}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i - \beta D_{KL}(\pi_\theta \parallel \pi_{\text{ref}})\right], \quad (7)$$

where the loss $\mathcal{L}_i$ can be defined as:

$$\mathcal{L}_i = \min\left[\frac{\pi_\theta(\hat{o}_i|x)}{\pi_{\text{old}}(\hat{o}_i|x)}\hat{A}_i,\ \text{clip}\left(\frac{\pi_\theta(\hat{o}_i|x)}{\pi_{\text{old}}(\hat{o}_i|x)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right], \quad (8)$$

and the normalized advantage $\hat{A}_i$ is:

$$\hat{A}_i = \frac{\hat{R}_{\text{total},i} - \text{mean}\left(\{\hat{R}_{\text{total},1}, \ldots, \hat{R}_{\text{total},N}\}\right)}{\text{std}\left(\{\hat{R}_{\text{total},1}, \ldots, \hat{R}_{\text{total},N}\}\right)}, \quad (9)$$

where β and ϵ are hyperparameters.

Experiments

Experimental Settings

Implementation Details We use Qwen2.5-VL-7B (Bai et al. 2025) as the base model, starting with 2 epochs of full-parameter SFT followed by 200 steps of TGRL fine-tuning on the pretrained model. During the two-stage training, the batch size is set to 16, with 8 samples per training step, an initial learning rate of 1e-6, and weight decay of 0.01. The training is conducted on 8 NVIDIA A100 GPUs.

Datasets We conduct two-stage training using our circuit image-netlist dataset, which includes 2,500 training and 600 test pairs. During training, SFT is performed on a 1,000-sample subset, and TGRL uses the full training set. We evaluate SPICE netlist accuracy on our test set and the AMSNet dataset (Tao et al. 2024), and assess downstream circuit QA accuracy on the AMSBench dataset (Shi et al. 2025b).

Evaluation Metrics We evaluate circuit-to-netlist translation accuracy from two dimensions: answer accuracy is computed using the Jaccard similarity between the predicted and ground-truth SPICE netlists, while logic consistency is

Models	Params	Thinking	Accuracy ↑	
			Our Dataset	AMSNet Dataset†
GPT-4o	200B	×	39.08	53.83
Qwen2.5-VL	72B	×	35.05	47.15
QVQ-Preview	72B	✓	43.41	25.40
Deepseek-V2	27B	✓	33.56	33.75
GLM-4.1V	9B	✓	31.92	38.44
Qwen2.5-VL	7B	×	37.13	25.04
AMSNet	- *	×	27.14	-
Auto-SPICE	- **	×	29.48	58.90
Circuit-Think	**7B**	✓	**73.27**	**69.32**

† constructed using AMSNet. Therefore, it achieves 100% theoretical performance on itself.
∗ It uses the YOLO-V8 for component detection and generates the netlist pseudocode according to predefined rules.
∗∗ It annotates the images using YOLO-V8 and generates the netlists using GPT-4o.

Table 2: The SPICE netlist accuracy on our circuit image-netlist dataset and the AMSNet dataset.

measured by comparing the predicted SPICE netlist with the intermediate connection reasoning via Jaccard similarity. For circuit QA tasks, Accuracy is the evaluation metric.

Main Experiment Results

TGRL Learning Paradigm Result Table 1 compares different fine-tuning paradigms on Qwen2.5-VL 7B (Bai et al. 2025). We compare SPICE netlist accuracy within 200 steps. The results show that the proposed TGRL paradigm improves accuracy by 25.39% over traditional SFT at 200 steps, while SFT exhibits overfitting in later steps. Additionally, TGRL outperforms the reinforcement learning-based GRPO technique by 19.94% in accuracy. These results show that TGRL steadily improves model performance and significantly enhances the model's generalization and robustness in complex circuit-to-netlist translation tasks. Figure 5 shows the reward scores for format, reasoning, netlist, and logic over 200 steps in TGRL. The results indicate steady improvement in all dimensions as training progresses.

Circuit-to-Netlist Translation Result We compare Circuit-Think with other general VLM models on our dataset and the publicly available AMSNet dataset (Tao et al. 2024) for circuit-to-netlist accuracy. As shown in Table 2, Circuit-Think improves SPICE netlist accuracy by 34.19% over the closed-source GPT-4o with reasoning capabilities on our dataset, and more than 15.49% on the AMSNet dataset. Compared to Auto-SPICE (Bhandari et al. 2024) based on computer vision, Circuit-Think improves by 43.79% on the custom dataset and over 10.42% on the AMSNet dataset. Despite fine-tuning on a small-scale dataset, these results demonstrate that the 7B model outperforms larger models, showcasing state-of-the-art performance and strong generalization capability. Furthermore, Figure 4 illustrates a complete example of the reasoning process from circuit image parsing to SPICE netlist generation.

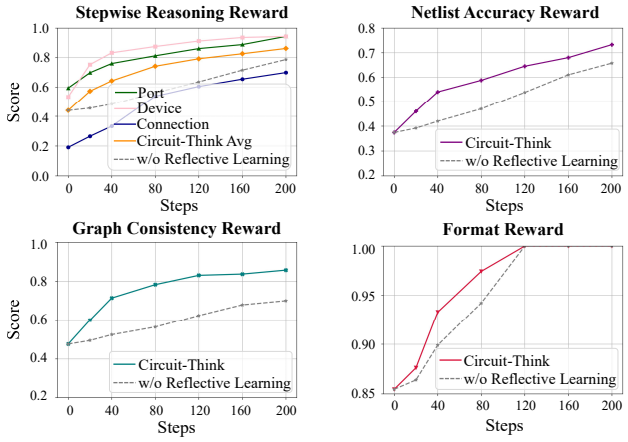

Figure 5: Batch-wise reward comparison during training: Circuit-Think *v.s.* w/o Reflective Learning Mechanism.

Circuit Image QA Result We evaluate QA accuracy across multiple circuit image QA scenarios on the AMS-Bench dataset (Shi et al. 2025b) to assess the effectiveness of our framework on downstream tasks. We compare the results of using only circuit images and using both circuit images and SPICE netlists generated by Circuit-Think. As shown in Table 3, in the Connection Identification scenario, Gemini-2.5-pro (Comanici et al. 2025) achieves a 10.97% improvement, and Qwen2.5-VL 72B (Bai et al. 2025) improves by 9.2%. The results demonstrate that integrating SPICE netlist information significantly enhances performance across multiple circuit image QA scenarios, validating the auxiliary role of the netlist in this task and further confirming the effectiveness of our SPICE netlist generation method in supporting reasoning and improving QA accuracy.

Ablation Results

We conduct ablation studies to validate the effectiveness of reward functions and reflective learning mechanism.

Soft *v.s.* Hard Format Reward We apply three levels of format constraints (w/o: no format constraint, Soft: constraining only think and answer, strict: constraining think, port, device, connection, and answer together) to evaluate their impact on SPICE netlist accuracy and logic consistency. As shown in Table 4, removing constraints reduces accuracy to 39.21%, while stricter constraints significantly improve both metrics. This indicates that the step-by-step reasoning paradigm effectively regulates the reasoning trajectory and enhances task performance.

Design of Stepwise Reasoning Reward We evaluate the effectiveness of stepwise reasoning rewards across three key stages: port recognition, device recognition, and connection reasoning. Port and device recognition involves type classification and location identification. As shown in Table 5, incorporating port and device recognition increases SPICE netlist accuracy from 45.76% to 48.13% and improves logic consistency by approximately 22%. With connection reasoning added, both metrics improve significantly. Results

Models	Input	Element Classification	Connection Identification	Connection Judgment	Function	Total Counting	Type-wise Counting
GPT-4o	Image	87.01	64.36	70.78	81.50	57.73	56.49
	+ Netlist†	**94.33**	**68.70**	**77.78**	**92.86**	**59.51**	**75.93**
Gemini-2.5-pro	Image	92.99	67.09	79.17	91.72	**58.59**	76.13
	+ Netlist†	**93.80**	**78.06**	**87.18**	**93.50**	54.35	**78.95**
Claude-4-Sonnet	Image	91.67	61.80	64.67	78.89	52.72	71.32
	+ Netlist†	**93.75**	**67.45**	**73.05**	**89.80**	**59.30**	**79.03**
Qwen2.5-VL-72B	Image	86.49	57.38	74.83	81.58	33.77	54.33
	+ Netlist†	**88.86**	**66.58**	**78.57**	**89.00**	**47.73**	**67.12**

† indicates that the images are first translated into SPICE netlists by our framework and then fed into the models to enrich the input.

Table 3: The accuracy of representative VLM models across multiple downstream tasks is compared on the AMSBench dataset (Shi et al. 2025b), using our Circuit-Think as a prefix to extract the netlist or not.

Models	Type	Accuracy ↑	
		Answer ↑	Logic ↑
Circuit-Think	w/o	39.21	-
Circuit-Think	Soft	52.69	65.83
Circuit-Think	**Strict**	**73.27**	**85.76**

Table 4: Ablation on various format constraints.

Port	Device	Connection	Accuracy ↑	
			Answer ↑	Logic ↑
×	×	×	45.76	41.45
✓	×	×	45.93	59.39
✓	✓	×	48.13	63.82
✓	✓	✓	**73.27**	**85.76**

Table 5: Ablation on the three steps in the reasoning process.

Operation	Accuracy ↑	
	Answer ↑	Logic ↑
Circuit-Think	**73.27**	**85.76**
w/o port/device_type	47.13	59.21
w/o port/device_bbox	49.91	63.92
w/o Step-by-Step	35.38	-
w/o R_{logic}	63.25	-
w/o Reflective Learning	65.76	70.10

Table 6: Ablation of the important operations independently.

highlight the critical role of stepwise constraints in enhancing netlist quality while maintaining logical consistency.

Design of Bbox and Type Table 6 evaluates the accuracy impact of type classification and location identification. When port and device type classification is removed (w/o port/device_type), SPICE netlist accuracy drops to 47%, and logical consistency is not maintained. It shows that identifying component types and port attributes is fundamental for understanding circuit structures and determining functional relationships during reasoning. When location identification is removed (w/o port/device_bbox), SPICE netlist accuracy drops to 49%. It shows that type classification and location identification play distinct yet complementary roles in circuit interpretation: type information is more critical for constructing the structural framework, while spatial information ensures accurate inference of positional relationships.

Strategy of Reasoning Logic To evaluate the impact of the step-by-step reasoning paradigm and logic consistency constraint, Table 6 reports ablation results. Removing the "port-device-connection" trajectory (w/o step-by-step) reduces SPICE netlist accuracy to 35.38%, indicating that the absence of structured reasoning disrupts the reasoning chain and impairs netlist generation. Omitting the logic consistency reward (w/o R_{logic}) also decreases accuracy, highlighting the importance of maintaining alignment between the reasoning process and netlist output for reliable results.

Strategy of Reflective Learning Mechanism As shown in Table 6, removing the reflective learning (w/o Reflective Learning) reduces SPICE netlist accuracy from 73.27% to 65.76% and logic consistency from 85.76% to 70.10%. Figure 5 further illustrates its impact on reward scores. The results show that reflective learning mechanism effectively reduces early-stage reasoning errors, boosting performance. In later stages, it ensures consistency between the reasoning trajectory and SPICE netlist output, preventing error accumulation and ensuring stable, logical netlist generation.

Conclusion

This work presents Circuit-Think, a multimodal reasoning framework for circuit-netlist translation. Leveraging the TGRL paradigm, the framework integrates step-by-step reasoning, a multi-level reward mechanism, and the reflective learning mechanism to optimize reasoning quality and SPICE netlist generation. Extensive experiments show that Circuit-Think surpasses existing netlist accuracy and reasoning coherence methods. Future work will focus on enhancing generalization and adaptability for broader applications.

Acknowledgments

This work is sponsored by The National Natural Science Foundation of China (Grant No. 62034007), NSFC-FWO No. W2412034, and the Zhejiang University Education Foundation Qizhen Scholar Foundation.

References

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

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 e-prints*, arXiv–2411.

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 *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 1–9.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Gao, J.; Cao, W.; Yang, J.; and Zhang, X. 2025. AnalogGenie: A generative engine for automatic discovery of analog circuit topologies. *arXiv preprint arXiv:2503.00205*.

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*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hemker, D.; Maalouly, J.; Mathis, H.; Klos, R.; and Ravanan, E. 2024. From Schematics to Netlists–Electrical Circuit Analysis Using Deep-Learning Methods. *Advances in Radio Science*, 22: 61–75.

Jiang, Y.; Jin, Q.; Lu, X.; Deng, J.; Geng, H.; Wu, H.; Sun, Q.; and Zhuo, C. 2025. FabThink: A Wafer Analysis Multimodal LLM via Chain-of-Thought-Driven Retrieval Augmentation. In *Proceedings of the 44rd IEEE/ACM International Conference on Computer-Aided Design*, 1–8.

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 *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 1–8.

Jin, Q.; Liu, Y.; Jiang, Y.; Sun, Q.; and Zhuo, C. 2025. Unitho: A Unified Multi-Task Framework for Computational Lithography. In *Proceedings of the 44rd IEEE/ACM International Conference on Computer-Aided Design*, 1–8.

Jocher, G.; and Qiu, J. 2024. Ultralytics YOLO11.

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.; Tao, Z.; Gao, Y.; Zhou, T.; Chang, C.; Wang, Y.; Chen, B.; Zhang, G.; Liu, A.; Yu, Z.; et al. 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.; Zhang, Z.; Wang, H.; Tao, Z.; Li, Z.; Chen, B.; Wang, Y.; 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.; et al. 2024. Amsnet: Netlist dataset for ams circuits. In *2024 IEEE LLM Aided Design Workshop (LAD)*, 1–5. IEEE.

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*.

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*.

Zhu, E.; Liu, Y.; Zhang, Z.; Li, X.; Zhou, J.; Yu, X.; Huang, M.; and Wang, H. 2025. Maps: Advancing multi-modal reasoning in expert-level physical science. *arXiv preprint arXiv:2501.10768*.