

# Toward Building AI Analog IC Design Engineers

Bin Hu (UIUC), Elyse Rosenbaum (UIUC), Pavan Hanumolu (UIUC), Huan Zhang (UIUC)

EDPS 2025

October 2, 2025

# The Era of Large Language Models (LLMs)

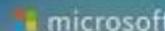
 OpenAI

 deepseek

 Claude

 Meta AI

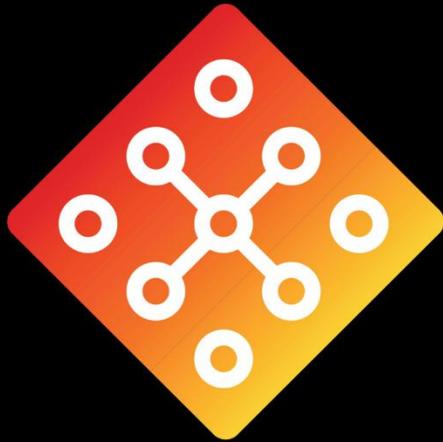
 Gemini

 microsoft  
/Phi-4-reasoning

 Grok

 Qwen

- Generate texts beyond average human writing ability
- Generate functional code across languages and accelerate software development
- Power user-friendly search
- Assist in scientific discovery
- Reason through complex tasks across various technical domains
- “Sparks of AGI”
- **AI engineers?**



# CAEML

CENTER FOR ADVANCED ELECTRONICS  
THROUGH MACHINE LEARNING



PennState



ILLINOIS

NC STATE  
UNIVERSITY

AN NSF INDUSTRY/UNIVERSITY COOPERATIVE RESEARCH CENTER

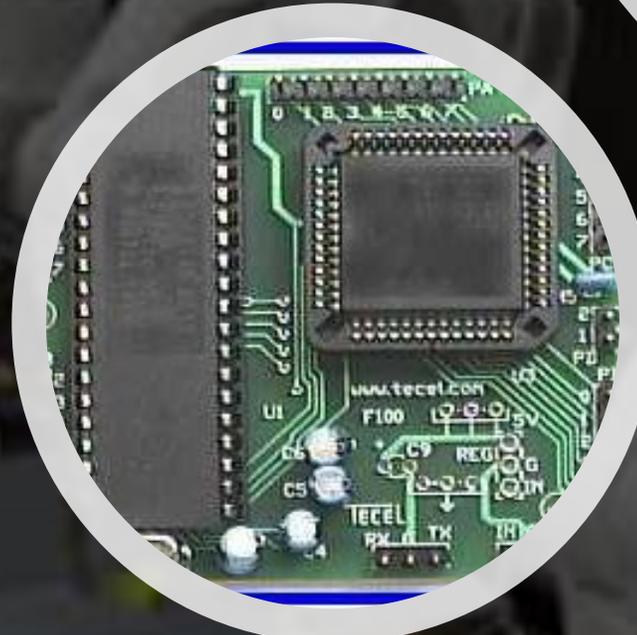
## **P25-13: Reinforcement Learning and Test-Time Scaling for Continuously Training and Improving LLMs for Analog/Mixed-Signal Circuit Design (Starting from 08/01/2025)**

**Bin Hu (PI), Pavan Hanumolu (Co-PI), Elyse Rosenbaum (Co-PI), Huan Zhang (Co-PI), University of Illinois.**

# Outline

- Toward Engineering AGI: EngDesign Benchmarking the engineering design capabilities of LLMs
- Past successful story I: LLM agents with test-time scaling for control design
- Past successful story II: Reinforcement learning for finetuning VLM for robotics
- Toward Building LLM-based Analog IC Design Engineers: Where are we right now?

- Engineering: Cornerstone of Modern Society
- Definition: Application of scientific principles to design or develop structures, machines, or manufacturing processes



A simple procedure for measuring surface convection heat transfer coefficients involves coating the surface with a thin layer of material having a very low thermal conductivity. The surface is then heated and, by determining the time required for the coating to melt, the convection coefficient is determined. The following experimental arrangement uses the procedure to determine the convection coefficient for gas flow normal to a surface. Specifically, a long copper rod is initially at 25°C and gas flow for which  $h = 200 \text{ W/m}^2\text{K}$  is applied. What is the melting point temperature of the coating if it is observed to melt at  $t = 400 \text{ s}$ ? ASSUMPTIONS: (1) One-dimensional conduction in the rod, (2) Rod approximated as semi-infinite medium, (3) Negligible thermal resistance of the coating, (4) Constant properties, (5) Negligible thermal resistance of the rod (Given:  $k = 400 \text{ W/m}\cdot\text{K}$ ,  $\alpha = 10^{-4} \text{ m}^2/\text{s}$ ).



# DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

research@deepseek.com

## Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.



Calculated coating's melting point through thermal transfer analysis

### Determining the Melting Point Temperature

I'll solve this problem using heat transfer principles for a semi-infinite solid with convection at the surface, the temperature is given by:

#### Given:

- Copper rod initially at  $T_i = 25^\circ\text{C}$
- Gas flow with  $h = 200 \text{ W/m}^2\text{K}$  and  $T_\infty = 300^\circ\text{C}$
- Melting observed at  $t = 400 \text{ s}$
- Copper properties:  $k = 400 \text{ W/m}\cdot\text{K}$ ,  $\alpha = 10^{-4} \text{ m}^2/\text{s}$

#### Solution:

For a semi-infinite solid with convection at the surface, the temperature is given by:

$$\frac{T(0, t) - T_i}{T_\infty - T_i} = 1 - \exp\left(-\frac{h^2 \alpha t}{k^2}\right) \times \text{erfc}\left(\frac{h\sqrt{\alpha t}}{k}\right)$$

Let's calculate each term:

$$\frac{h^2 \alpha t}{k^2} = \frac{(200)^2 \times 10^{-4} \times 400}{(400)^2} = 0.01$$

$$\frac{h\sqrt{\alpha t}}{k} = \frac{200 \times \sqrt{10^{-4} \times 400}}{400} = 0.1$$

The value of  $\exp(0.01) \approx 1.01$  and  $\text{erfc}(0.1) \approx 0.887$

Substituting these values:

$$\frac{T(0, 400) - 25}{300 - 25} = 1 - 1.01 \times 0.887 = 0.104$$

Therefore:

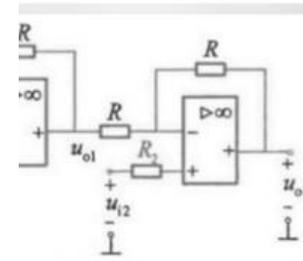
$$T(0, 400) - 25 = 0.104 \times 275$$

#### Answer:

The melting point temperature of the coating is approximately 53.6°C.

cascaded stages, each having an op-amp with infinite resistor,  $R$ , with the negative input of the op-amp is the output voltage  $u_o$ , I'll need to deduce the exact structure either a summing or differential amplifier. I'm focusing on key components.

more details. For the op-amp: I think the negative input is a resistor  $R$  inside the left block – could it be the infinity symbol about the labels though... there's a plus sign at the left side. That makes a huge difference in how I



here are plus and minus signs near the input terminals, the minus on the top right. Typically, an op-amp is drawn in a triangle. I'm trying to make sure the labeling is accurate. Check out the image again—hopefully, that helps clarify.

the SOTA using reinforcement

# LLMs are far from the level of practicing engineers!

## Advanced Analog IC Design Project: Pipelined ADC design

The objective of this project is to design the first stage of a 1.5b/stage 12-bit pipelined ADC. The primary goal is to design an ADC and switched capacitor MDAC for the first stage to meet the performance requirements in Table 1 with minimum power consumption. To simulate the ADC, the first stage will be modeled in a circuit simulator. The course website, and the source in your design using transistors.

## Solid-State Drive Design Problem

In this problem, you will need to finalize a set of Solid-State Drive hardware that meet the performance requirement of a specific workload type with several design constraints.

- 1 Background Knowledge
  - 1.1 Basic SSD Architecture

## Flexible Drone Design Task (from Prof. Gregory Elliot)

### SCENARIO

The project involves developing a transportable hand launched mini-UAV (MUAV). The MUAV should be easily transportable in a SUV (2020 Toyota Highlander). The mission of the MUAV is create a ¼ scale prototype to evaluate the aerodynamics of the full-scale aircraft. The MUAV must be able to transport a cylinder 0.75" in diameter, 1.5" long and a mass of 5 grams (0.176 oz) which represent a scaled down version of the payload which will be transported to the full-scale aircraft. Although in reality the aircraft

Existing LLMs (or agents) cannot solve any of such design tasks.

Resolution	12
Conversion rate	20MS
SNDR	$\geq$
SFDR	$\geq$
DNL	$\leq 0$
INL	$\leq$
Power consumption	Min

has multiple flash chips. Within each chip, there are multiple planes and each block has multiple flash pages. The size of a flash page varies in different SSDs. When a free flash page is written on, it is no longer available for future writes until that page is erased. Her operation is expensive and performed at block granularity. As each block has limited endurance, it is important for blocks to age uniformly (leveling). Modern SSD controllers employ out-of-place write, garbage collection, and wear leveling to overcome these shortcomings and maintain indirect addressing for translation in their flash translation layer (FTL).

In this problem, our aim is to design SSDs that allow developers to design SSD hardware according to the application needs.

## 2 Problem Definition

We define the entire possible configuration set as  $S$ , the baseline configuration as  $c_{base}$ , the configuration constraints as  $C_{cons}$ , and the performance

**Wing:** Balsa reinforced foam with 1 oz fiberglass covering. Separation from fuselage preferred but not required as long as it is transportable. Keep thickness less than 1.75".

**Wingspan:** less than 70"

**Size:** Transportable by single person and can fit in a 2020 Toyota Highlander SUV

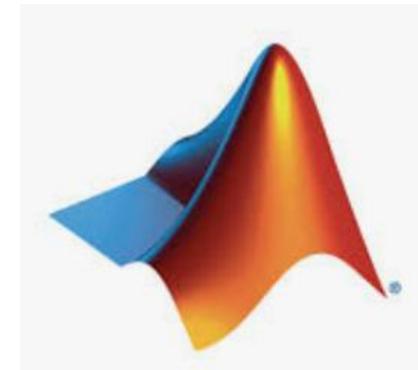
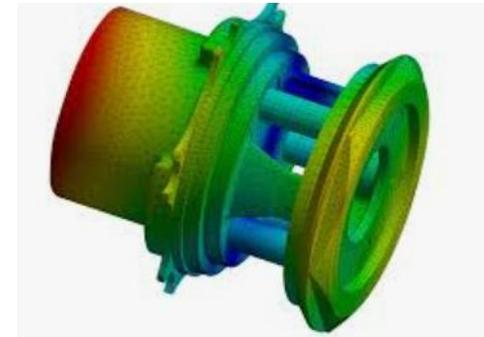
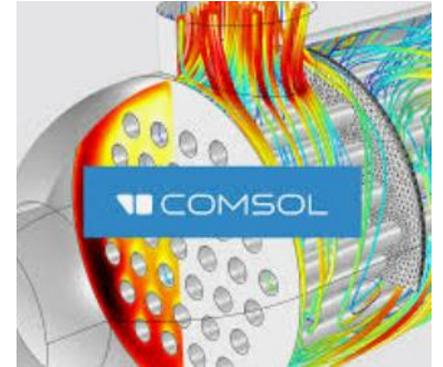
**Construction:** Reinforced foam, carbon fiber or plywood structural members, fiberglass skin, balsa and bass wood, 3D printed PLA and foam fuselage skin, single boom to tail. For initial wing weight estimates use 0.033 oz/in<sup>2</sup> (based on planform area of wing), for tails use 0.030 oz/in<sup>2</sup> (based on planform area of tail), for winglets use 0.060 oz/in<sup>2</sup> (based on planform area of winglet)

**Configuration:**  
Tractor tailless (flying wing) aircraft

**Weight:** Less than 3 pounds take-off weight

# From Textbook to Practical Design Problems

- Practical engineering design problems are diverse and challenging!
- Human practicing engineers rely on experiences which go beyond textbooks
- Many problems require domain specific engineering tools (e.g., SPICE, Simulink, Ansys, COMSOL, Matlab toolboxes, etc) and involve real physical data
- **How far away are LLMs from practicing engineers?**



# Toward Engineering AGI: Benchmarking the Engineering Design Capabilities of LLMs



*Xingang Guo, Yaxin Li, Xiangyi Kong, YILAN JIANG, Xiayu Zhao, Zihua Gong, Yufan Zhang, Daixuan Li, Tianle Sang, Beixiao Zhu, Gregory Jun, Yingbing Huang, Yiqi Liu, Yuqi Xue, Rahul Dev Kundu, Qi Jian Lim, Yizhou Zhao, Luke Alexander Granger, Mohamed Badr Younis, Darioush Keivan, Nippun Sabharwal, Shreyanka Sinha, Prakhar Agarwal, Kojo Vandyck, Hanlin Mai, Zichen Wang, Aditya Venkatesh, Ayush Barik, Jiankun Yang, Chongying Yue, Jingjie He, Libin Wang, Licheng Xu, Hao Chen, Jinwen Wang, Liujun Xu, Rushabh Shetty, Ziheng Guo, Dahui Song, Manvi Jha, Weijie Liang, Weiman Yan, Bryan Zhang, Sahil Bhandary Karnoor, Jialiang Zhang, Rutva Pandya, Xinyi Gong, Mithesh Ballae Ganesh, Feize Shi, Ruiling Xu, Yifan Zhang, Yanfeng Ouyang, Lianhui Qin, Elyse Rosenbaum, Corey Snyder, Peter Seiler, Geir Dullerud, Xiaojia Shelly Zhang, Zuofu Cheng, Pavan Kumar Hanumolu, Jian Huang, Mayank Kulkarni, Mahdi Namazifar, Huan Zhang, Bin Hu (hide authors)*

Published: 18 Sept 2025, Last Modified: 18 Sept 2025 [NeurIPS 2025 Datasets and Benchmarks Track poster](#) [Datasets and Benchmarks Track, Senior Area Chairs, Area Chairs, Reviewers, Authors](#) [Revisions](#) [BibTeX](#) [CC BY 4.0](#)

Edit ▾

**Keywords:** Engineering Design, LLM Benchmark

**Abstract:**

Modern engineering, spanning electrical, mechanical, aerospace, civil, and computer disciplines, stands as a cornerstone of human civilization and the foundation of our society. Today, industry pioneers dream of developing AI systems capable of designing and building humanity's most ambitious projects—from starships that will carry us to distant worlds to Dyson spheres that harness stellar energy. Yet engineering design represents a fundamentally different challenge for large language models (LLMs) compared to traditional textbook-style problem solving or factual question answering. Although existing benchmarks have driven progress in areas such as factual question answering, code synthesis, and scientific problem solving, real-world engineering design demands synthesis of domain knowledge, navigation of complex trade-offs, and management of the tedious processes that consume much of practicing engineers' time. Despite these shared challenges across engineering disciplines, no benchmark currently captures the unique demands of engineering design work. In this work, we introduce EngDesign, an Engineering Design benchmark that evaluates LLMs' abilities to perform practical design tasks across nine engineering domains: Operating System Design, Computer Architecture Design, Control System Design, Mechanical Systems, Structural Design, Digital Hardware Design, Analog Integrated Circuit Design, Robotics, and Signal Processing. Unlike existing benchmarks that focus on factual recall or question answering, EngDesign uniquely emphasizes LLMs' ability to synthesize domain knowledge, reason under constraints, and generate functional, objective-oriented designs. Each task in EngDesign represents a real-world engineering design problem, accompanied by a detailed task description specifying design goals, constraints, and performance requirements. We pioneer a simulation-based evaluation paradigm where LLM-generated designs undergo rigorous testing through executable, domain-specific simulations—from circuit SPICE simulations to structural finite element analysis, from control system validation to robotic motion planning. EngDesign establishes a new benchmark paradigm that moves beyond textbook knowledge to assess genuine engineering capability and shifts evaluation from static answer checking to dynamic, simulation-driven functional verification, marking a crucial step toward realizing the vision of engineering AGI.

**Dataset Submission:** This submission includes a dataset.

**Croissant File:** [↓ json](#)

**Dataset URL:** <https://huqqinqface.co/datasets/opt1zer/EnqDesign>

To Appear in **NeurIPS 2025**



## Toward Engineering AGI: Benchmarking the Engineering Design Capabilities of LLMs

Xingang Guo\*†, Yaxin Li\*, Xiangyi Kong\*, Yilan Jiang\*, Xiayu Zhao\*, Zihua Gong\*, Yufan Zhang\*,  
Daixuan Li, Tianle Sang, Beixiao Zhu, Gregory Jun, Yingbing Huang, Yiqi Liu, Yuqi Xue, Rahul Dev Kundu,  
Qi Jian Lim, Yizhou Zhao, Luke Alexander Granger, Mohamed Badr Younis, Darioush Keivan,  
Nippun Sabharwal, Shreyanka Sinha, Prakhar Agarwal, Kojo Vandyck, Hanlin Mai, Zichen Wang,  
Aditya Venkatesh, Ayush Barik, Jiankun Yang, Chongying Yue, Jingjie He, Libin Wang, Licheng Xu,  
Hao Chen, Jinwen Wang, Liujun Xu, Rushabh Shetty, Ziheng Guo, Dahui Song, Manvi Jha, Weijie Liang,  
Weiman Yan, Bryan Zhang, Sahil Bhandary Karnoor, Jialiang Zhang, Rutva Pandya,  
Xinyi Gong, Mithesh Ballae Ganesh, Feize Shi, Ruiling Xu, Yifan Zhang,  
Yanfeng Ouyang, Lianhui Qin, Elyse Rosenbaum, Corey Snyder, Peter Seiler, Geir Dullerud,  
Xiaojia Shelly Zhang, Zuofu Cheng, Pavan Kumar Hanumolu, Jian Huang,  
Mayank Kulkarni, Mahdi Namazifar, Huan Zhang, Bin Hu†

### EngDesign Team

\*Core Contributors

†Corresponding to: [xingang2@illinois.edu](mailto:xingang2@illinois.edu), [binhu7@illinois.edu](mailto:binhu7@illinois.edu),

 arXiv

 Dataset

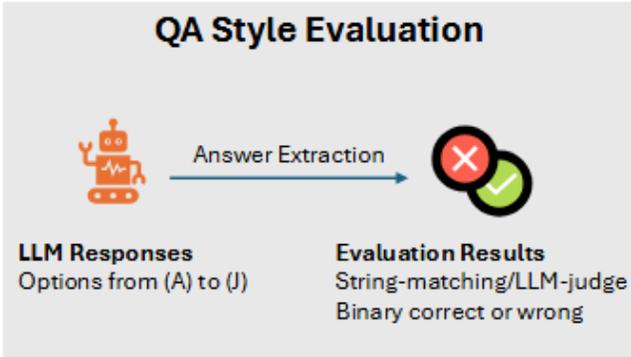
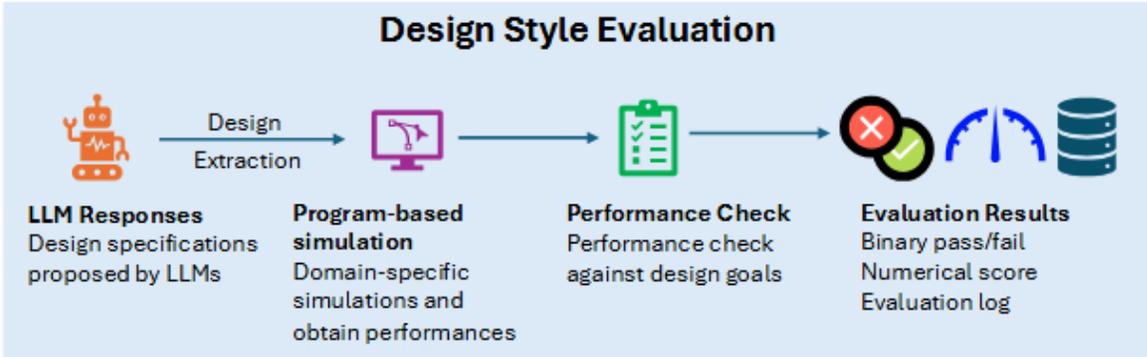
 Code

 Leaderboard

<https://agi4engineering.github.io/Eng-Design/>

# EngDesign

Table 1: Statistics of ENGDESIGN. Tokens are calculated with Tiktoken using o200k\_base encoding.

Engineering Area	# of query tokens	
Operating System Design	<p><b>QA Style Question</b></p> <p>A sample question from MMLU-Pro</p> <p>Two long parallel conductors carry 100 A. If the conductors are separated by 20 mm, the force per meter of length of each conductor will be</p> <p>(A) 100 N (B) 0.01 N (C) 10 N (D) 0.5 N (E) 0.001 N (F) 5 N (G) 1 N (H) 50 N (I) 0.1 N (J) 0.0001 N</p>	<p><b>Design Style Task</b></p> <p>A sample task from EngDesign (Ours)</p> <p>In this task, you are required to <b>design a suitable thickness (<math>T_h</math>) of a rectangular steel beam subjected to a dual-point load</b>, applied at the quarter-span locations ... You are given the following fixed parameters:</p> <ul style="list-style-type: none"> <li>- 'L = 1000' mm (Total span length)</li> <li>- 'w = 40' mm (Beam width, constant)</li> <li>- Two vertical downward forces: <math>F = 1000</math> N each, applied at quarter-span locations</li> <li>- Material properties:</li> <li>- Young's modulus: 210,000 MPa</li> <li>- Poisson's ratio: 0.3</li> <li>...</li> </ul> <p>Your task is to:</p> <ul style="list-style-type: none"> <li>- Propose a structurally sound value for <math>T_h</math> (thickness of the beam, in mm)</li> <li>- Provide a brief justification for your choice of thickness, considering stiffness, loading, and geometric constraints.</li> </ul>
Computer Arch Design	<p><b>QA Style Evaluation</b></p>  <p><b>LLM Responses</b> Options from (A) to (J)</p> <p><b>Evaluation Results</b> String-matching/LLM-judge Binary correct or wrong</p>	<p><b>Design Style Evaluation</b></p>  <p><b>LLM Responses</b> Design specifications proposed by LLMs</p> <p><b>Program-based simulation</b> Domain-specific simulations and obtain performances</p> <p><b>Performance Check</b> Performance check against design goals</p> <p><b>Evaluation Results</b> Binary pass/fail Numerical score Evaluation log</p>
Control Design		
Mechanical Systems		
Structure Design		
Digital Hardware Design		
Analog IC Design		
Robotics		
Signal Processing		
Overall		

# EngDesign

Domain: Control Design (XG\_05)



Evaluation Pipeline (with MATLAB)

Domain: Analog IC-Design (TB\_04)



Evaluation Pipeline (with SPICE Simulator)

Domain: Structure Design (RK\_03)



Evaluation Pipeline (with MATLAB)

Domain: Computer Architecture Design (DL\_01)



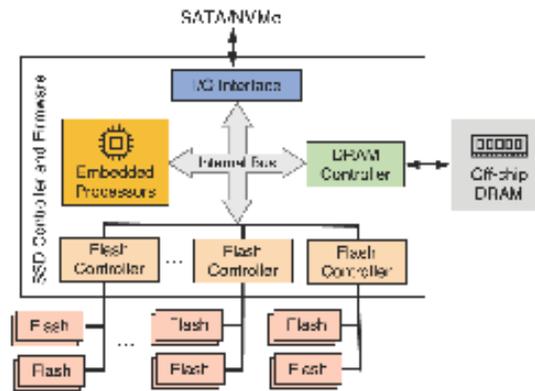
Evaluation Pipeline (with MQSim Simulator)

In this problem, you will finalize a set of Solid-State Drive (SSD) designs that meet the performance requirements for a specific workload type. Initially, you must determine which parameters to tune, and to what extent, to satisfy the given performance criteria ...

### Task Description

SSD customers typically evaluate SSD ... If we want to get 20% performance improvement on I/O throughput or ... Please provide your solution for each workload mentioned below:

Real-world Workloads. These workloads represents typical storage-intensive application patterns.



Workload Category	Description
Big Data Analytics	Data center MapReduce workloads .
Cloud Storage	Data center cloud storage workloads.
...	...
WebSearch	WebSearch services trace collected by UMass.
Advertisement	Advertisement workloads running on servers.

```
def evaluate_llm_response(llm_response):
    # Load ground truth obtained via MQSim simulation
    ground_truth = json.load(open(ground_truth_file, "r"))
    # Check performance for each workload
    passed, score, details = False, 0, {}
    for i, workload_name in
    enumerate(llm_response.config.workload_names):
        # Performance check for workload I
        # Check performance for each workload
        for workload in tuning_result.keys():
            ...
            # Scoring the LLM responses based on ground_truth
            for name in tuning_result[workload]:
                ...
    return passed, eval_details, score
```

MQSim simulation

```
<Execution_Parameter_Set>
<Device_Parameter_Set>
  <Data_Cache_Capacity>26
  8435456</Data_Cache_Cap
  acity>
```

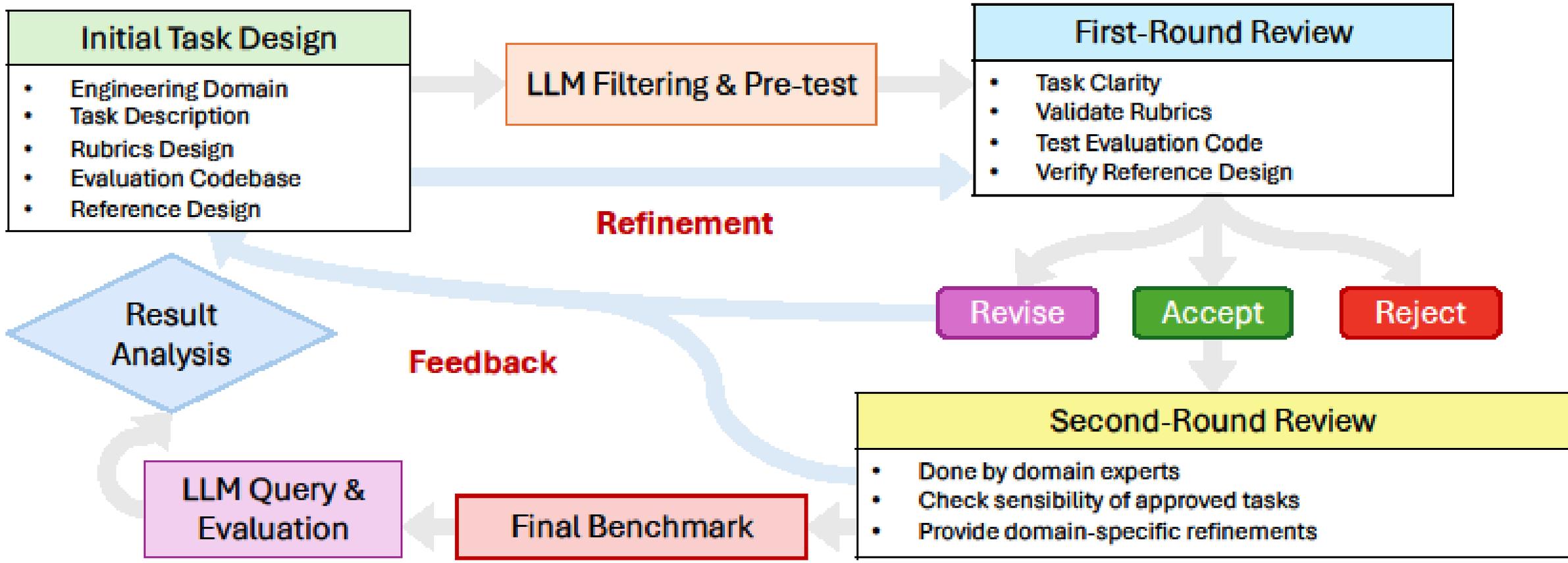
SSD Config files



```
{
  "LiveMapsBackend": {
    "Data_Cache_Capacity": [
      [Tuning_Result,
        Goal_Reached_flag,...
      ],
      ...
      ],
      ...
    ]
  }
}
```

Performance JSON

# EngDesign: Benchmark Construction

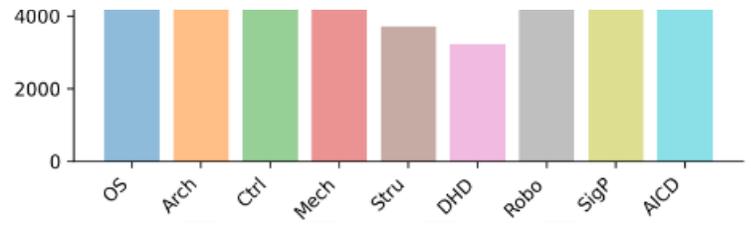
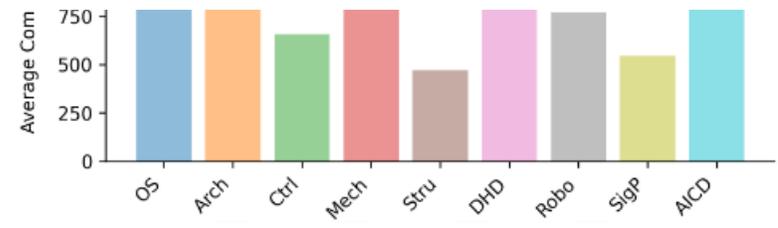
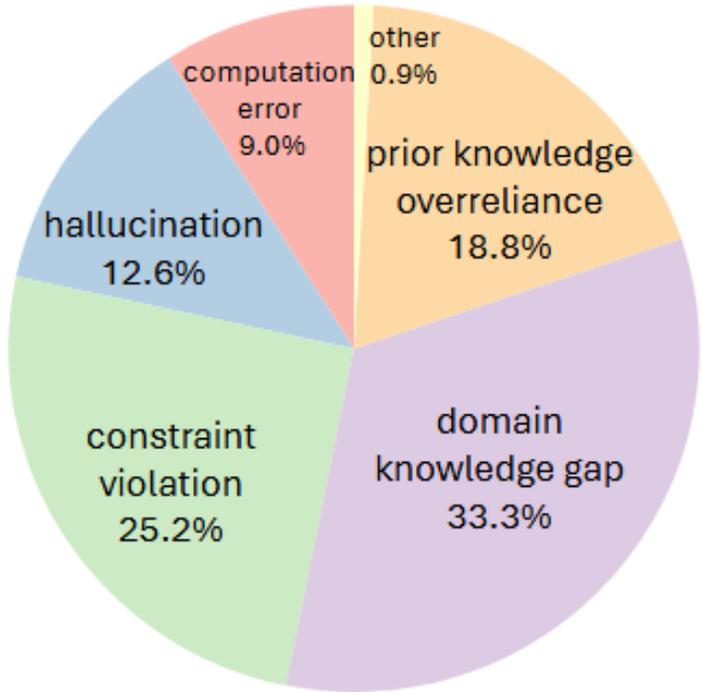
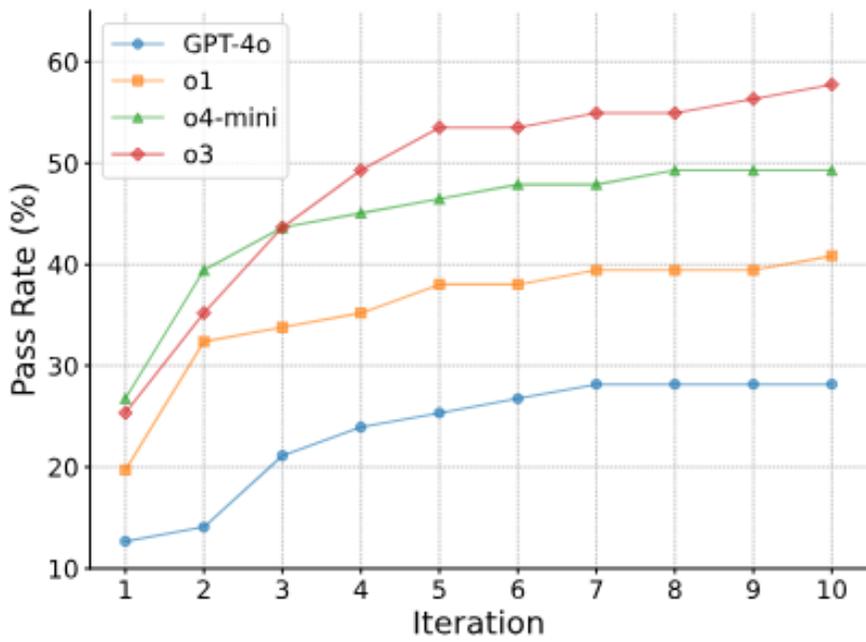
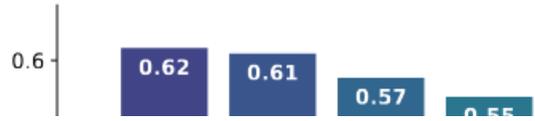


# EngDesign: Results

Table 2: Average pass rate (%) results (each task evaluated over 3 independent trials). \* Model is not multi-modal and was evaluated only on the text-only subset. We report text-only results for all models in Appendix B.4. The domain abbreviations are: AICD = Analog Integrated Circuit Design, Arch = Computer Architecture Design, Ctrl = Control Design, DHD = Digital Hardware Design, Mech = Mechanical Systems, OS = Operating System Design, Robo = Robotics, SigP = Signal Processing, Stru = Structure Design. The best results for each column are highlighted in bold.

Model	Overall	AICD	Arch	Ctrl	DHD	Mech	OS	Robo	SigP	Stru
<b>Chat Models</b>										
GPT-4o	15.68	0.00	20.00	18.52	10.26	0.00	4.17	26.67	17.65	25.64
DeepSeek-v3*	17.92	0.00	0.00	27.27	25.64	0.00	0.00	16.67	12.82	38.89
Gemini-2.0-Flash	14.16	0.00	0.00	15.15	16.67	4.76	0.00	20.00	5.13	36.11
Claude-3.7-Sonnet	22.61	0.00	<b>40.00</b>	16.67	33.33	14.29	0.00	33.33	21.57	30.77
<b>Reasoning Models</b>										
o1	29.17	0.00	<b>40.00</b>	24.07	41.03	14.29	<b>37.50</b>	50.00	25.49	23.08
o3	<b>34.38</b>	0.00	<b>40.00</b>	35.19	20.51	23.81	25.00	63.33	<b>41.18</b>	30.77
o3-high	33.57	0.00	<b>40.00</b>	27.78	20.51	<b>28.57</b>	30.43	<b>68.97</b>	35.29	41.03
o4-mini	31.60	0.00	33.33	24.07	41.03	19.05	29.17	43.33	33.33	41.03
o4-mini-high	34.04	0.00	<b>40.00</b>	27.78	<b>47.22</b>	19.05	<b>37.50</b>	46.67	35.29	35.90
Gemini-2.5-Pro	29.54	0.00	0.00	33.33	43.59	0.00	9.52	56.67	12.82	<b>50.00</b>
DeepSeek-R1*	25.53	0.00	36.36	<b>36.36</b>	38.46	4.76	5.26	26.67	20.51	41.67
Claude-3.7-Thinking	20.07	0.00	33.33	18.52	17.95	9.52	0.00	40.00	19.61	28.21

# EngDesign: More Results



# EngDesign: Key Take-aways

- Currently, it is difficult to develop a unified engineering AGI system.
- End-to-end analog IC design (topology selection and sizing) is very difficult for SOTA LLMs.
- Question 1: Instead of engineering AGI, can we do better for one area via developing specialized LLM agents?
- Question 2: Can we develop much smaller LLMs (<10B) with special engineering expertise?

# Outline

- Toward Engineering AGI: EngDesign Benchmarking the engineering design capabilities of LLMs
- Past successful story I: LLM agents with test-time scaling for control design
- Past successful story II: Reinforcement learning for finetuning VLM for robotics
- Toward Building LLM-based Analog IC Design Engineers: Where are we right now?

# ControlAgent

## ControlAgent: Automating Control System Design through the Integration of LLM Agents and Domain Expertise

Xingang Guo<sup>1</sup> Darioush Keivan<sup>1</sup> Usman Syed<sup>1</sup> Lianhui Qian<sup>1</sup>  
 Geir Dullerud<sup>1</sup> Peter Seiler<sup>3</sup> Bin Han<sup>2</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign

<sup>2</sup>University of California, San Diego

<sup>3</sup>University of Michigan

### Abstract

Control system design is a crucial aspect of modern engineering with far-reaching diverse sectors including aerospace, automotive systems, power grids, and more. Despite the advances made by Large Language Models (LLMs) in various domains, their application in control system design remains limited due to the complexity and specificity of control tasks. To bridge this gap, we introduce **ControlAgent**, a new paradigm that automates control system design through the integration of LLM agents and control-oriented domain expertise. ControlAgent integrates domain knowledge and emulates human iterative design processes by gradually tuning controller parameters to meet user-specified requirements for stability, performance (e.g. settling time), and margin). Specifically, ControlAgent integrates multiple collaborative LLM agents: a central agent responsible for task distribution and task-specific agents dedicated to details for various types of systems and requirements. In addition to LLM agents, ControlAgent includes a CoDesigner, a Python computation agent that performs complex control gain calculations and simulations based on standard design information (e.g. crossover frequency, etc) provided by the task-specific agents. Combined with a history and feedback module, the task-specific LLM agents refine controller parameters based on real-time feedback from prior designs. Overall, ControlAgent automates design processes used by (human) practicing engineers, but removes all the human intervention, running in a fully automated way to give end-to-end solutions for control system design. To validate ControlAgent's effectiveness, we develop **ControlEval** that comprises 500 control tasks with various specific design goals. The effectiveness of ControlAgent is demonstrated via extensive comparative evaluations between LLM-based and traditional toolbox-based baselines. Our numerical experiments show that ControlAgent can automate control design tasks, marking a significant step towards fully automated control system design. Our code is available at <https://github.com/ControlAgent/ControlAgent.git>

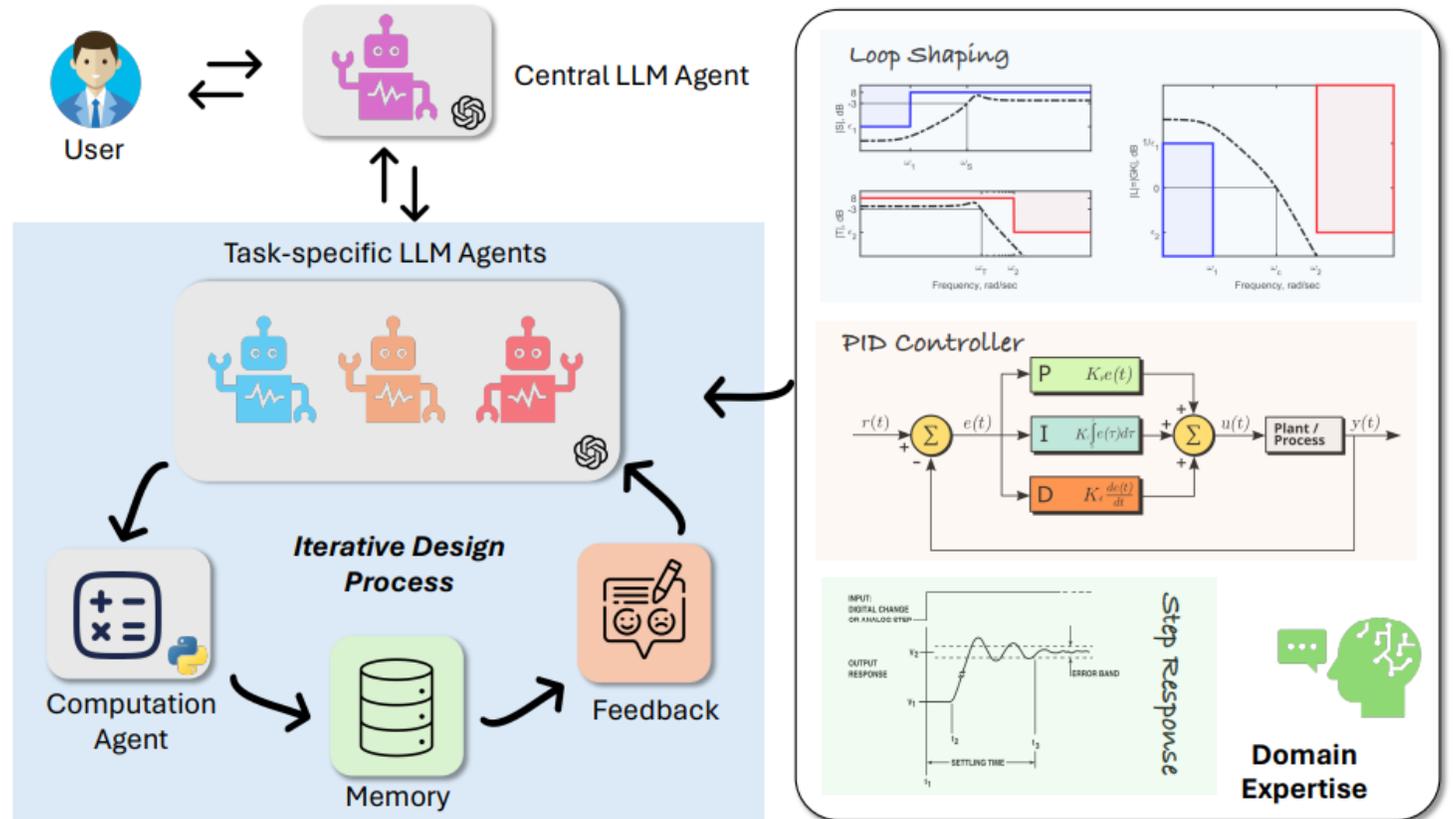
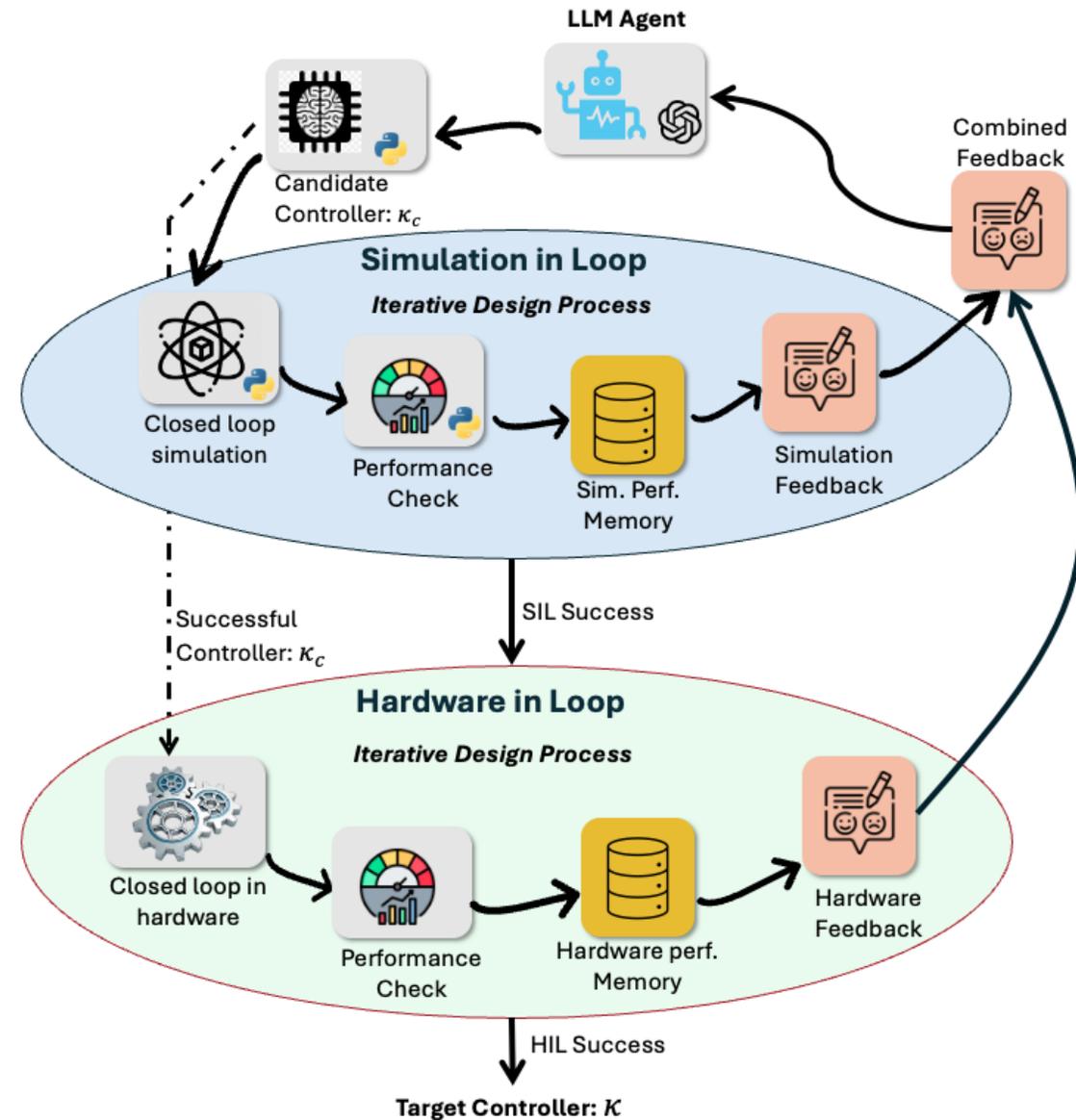


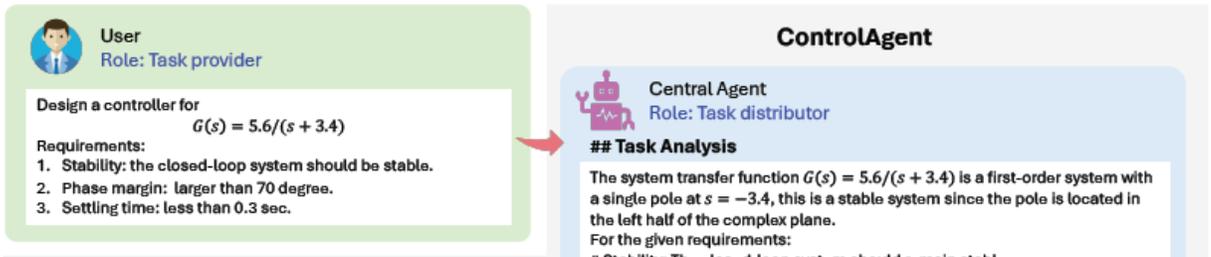
Figure 1: General ControlAgent framework.

# ControlAgent

- Multi-loop design process
- Starting design with simplified linear model
- Simulation in Loop (SIL) is the next level
- Hardware in Loop (HIL) is the outer loop
- Any failure leads to re-design again from inner loop
- Test-time scaling with multiple reasoning paths
- The above process mimics the human design process



# ControlAgent



Method	Metric	1st f	1st m	1st s	2nd f	2nd m	2nd s	1st u	2nd u	w dly	hgr	real-app
<b>Loop Shaping Prompt</b> (o1)	pass@1	0.30	0.36	0.02	0.51	0.24	0.02	0.04	0.20	0.19	0.26	0.04
	pass@3	0.62	0.73	0.05	0.88	0.45	0.07	0.13	0.32	0.42	0.43	0.12
	pass@5	0.78	0.86	0.08	0.98	0.54	0.10	0.22	0.38	0.54	0.50	0.20
<b>Loop Shaping Prompt</b> (o1-preview)	pass@1	0.72	0.60	0.14	0.65	0.43	0.15	0.31	0.42	0.41	0.31	0.08
	pass@3	0.98	0.93	0.37	0.97	0.78	0.38	0.62	0.71	0.75	0.58	0.21
	pass@5	<b>1.00</b>	<b>1.00</b>	0.56	<b>1.00</b>	0.88	0.56	0.78	0.82	0.86	0.68	0.30
<b>PIDtune</b>	pass@1	0.51	0.95	0.87	0.83	0.97	0.83	0.22	0.11	0.98	0.47	0.24
	pass@3	0.87	0.99	1.00	0.96	1.00	0.99	0.53	0.27	<b>1.00</b>	0.54	0.36
	pass@5	0.98	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.69	0.41	<b>1.00</b>	0.57	0.44
<b>ControlAgent</b> (Llama-3.3-70b)	pass@1	0.93	<b>1.00</b>	0.99	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.93	<b>0.96</b>	<b>1.00</b>	<b>0.91</b>	0.87
	pass@3	0.98	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.99	<b>1.00</b>	<b>1.00</b>	<b>0.97</b>	0.99
	pass@5	0.99	<b>1.00</b>	<b>0.99</b>	<b>1.00</b>							
<b>ControlAgent</b> (GPT-4o)	pass@1	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	<b>0.98</b>	0.91	0.97	0.82	<b>0.93</b>
	pass@3	<b>1.00</b>	0.99	<b>1.00</b>	0.95	<b>0.99</b>						
	pass@5	<b>1.00</b>	0.96	<b>1.00</b>								

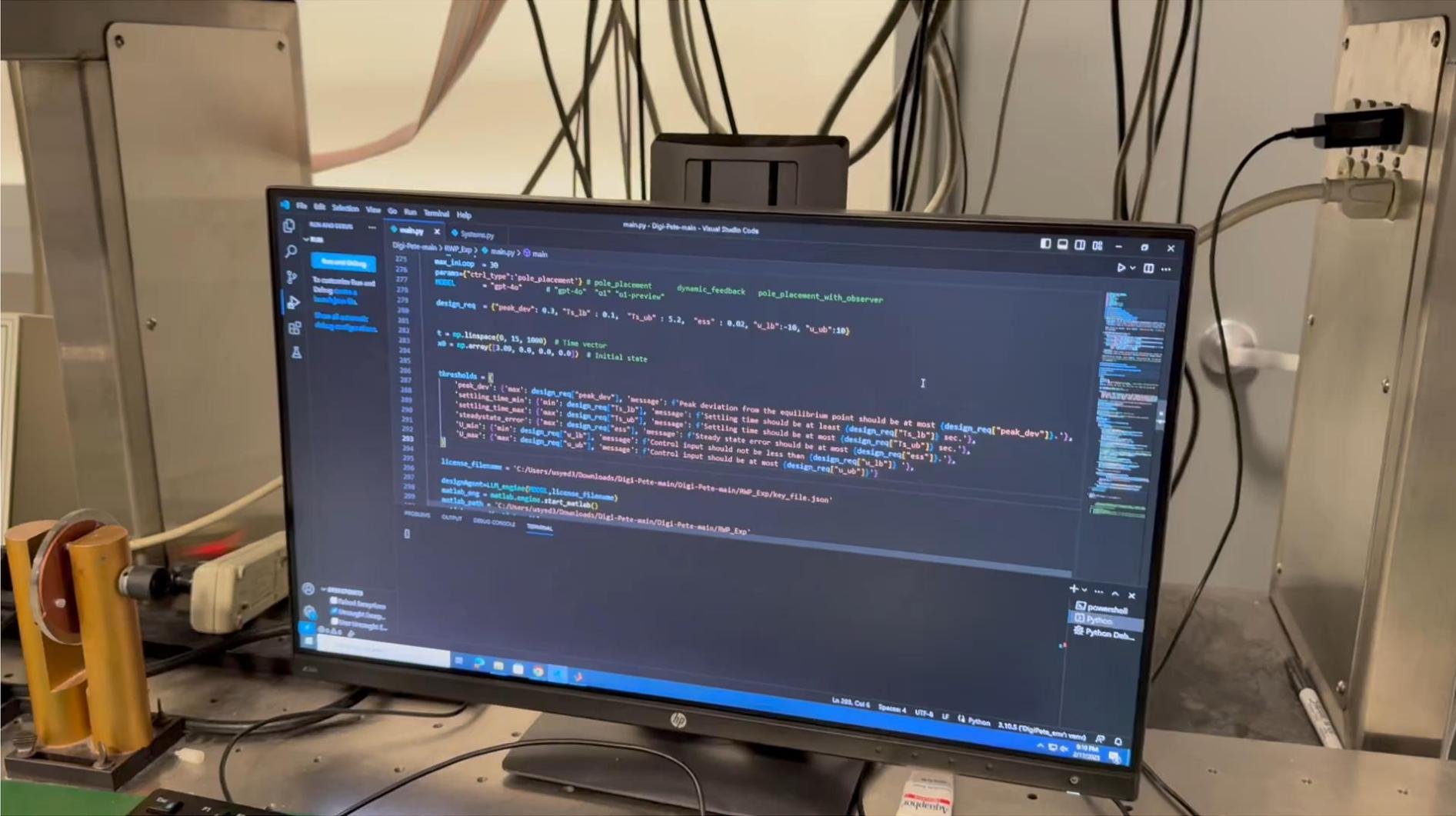
```

L = G * K
sys = ctrl.feedback(L, 1)
# Get the step response info of the closed-loop system
T(s)
info = ctrl.step_info(sys)
# Gain margin and phase margin
gm, pm, wg, wp = ctrl.margin(L)
# Steady-state error (assuming unit step input)
ess = 1 / (1 + np.abs(ctrl.dcgain(L)))
return pm, info['SettlingTime'], ess

### Design L
Parameters:  $\omega_L = 20$ ,  $\beta_b = 3.162$ 
Performance:
Steady state error = 0.0, hence the closed-loop system is stable;
Phase margin= 82.22 deg;
Settling time = 0.43 sec.

## Feedback
Feedback: Settling time should be less than 0.3 sec. Please design a faster controller.
    
```

# Demo: LLM-based Design for Inverted Pendulum



# ControlAgent: Key Take-aways

- For control engineering, integrating LLMs with analysis/design tools used by practicing engineers can lead to some initial preliminary successes.
- It is important to select the right space for the LLMs to reason over and scale up multiple inner/outer-loop reasoning paths with different tools.
- Analog IC design is the most difficult area for LLMs that we have tested so far. What to do?

# Outline

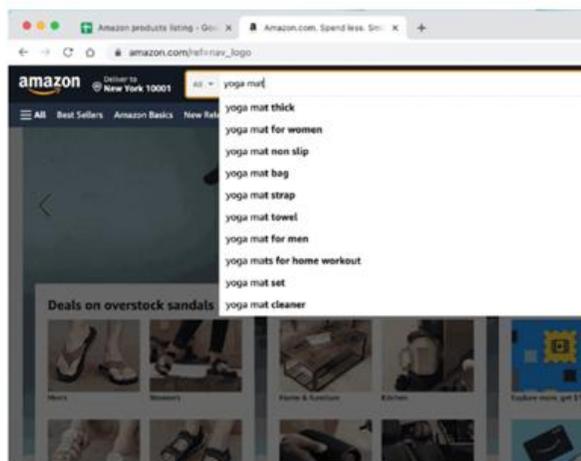
- Toward Engineering AGI: EngDesign Benchmarking the engineering design capabilities of LLMs
- Past successful story I: LLM agents with test-time scaling for control design
- Past successful story II: Reinforcement learning for finetuning VLM for robotics
- Toward Building LLM-based Analog IC Design Engineers: Where are we right now?

# Recent trend in RL-finetuning for LLMs/VLMs

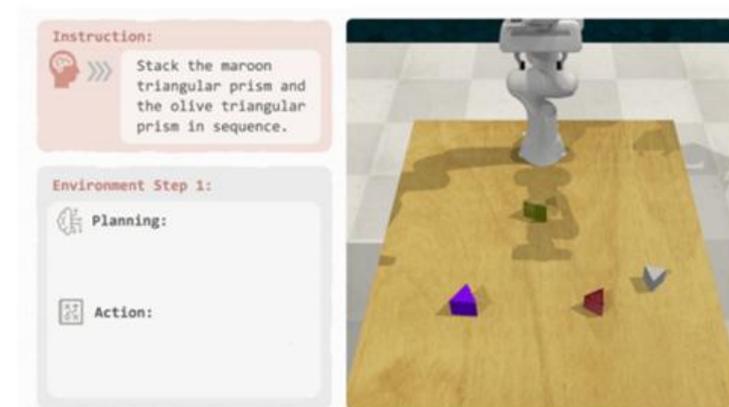
- RL paradigm popularized by DeepSeek-R1 (the GRPO algorithm)
- RL encourages reasoning and thinking in LLMs/VLMs
- Impressive performance improvements on various domains

```
[user]: write simple python code that creates neural network  
[assistant]:
```

Coding LLMs

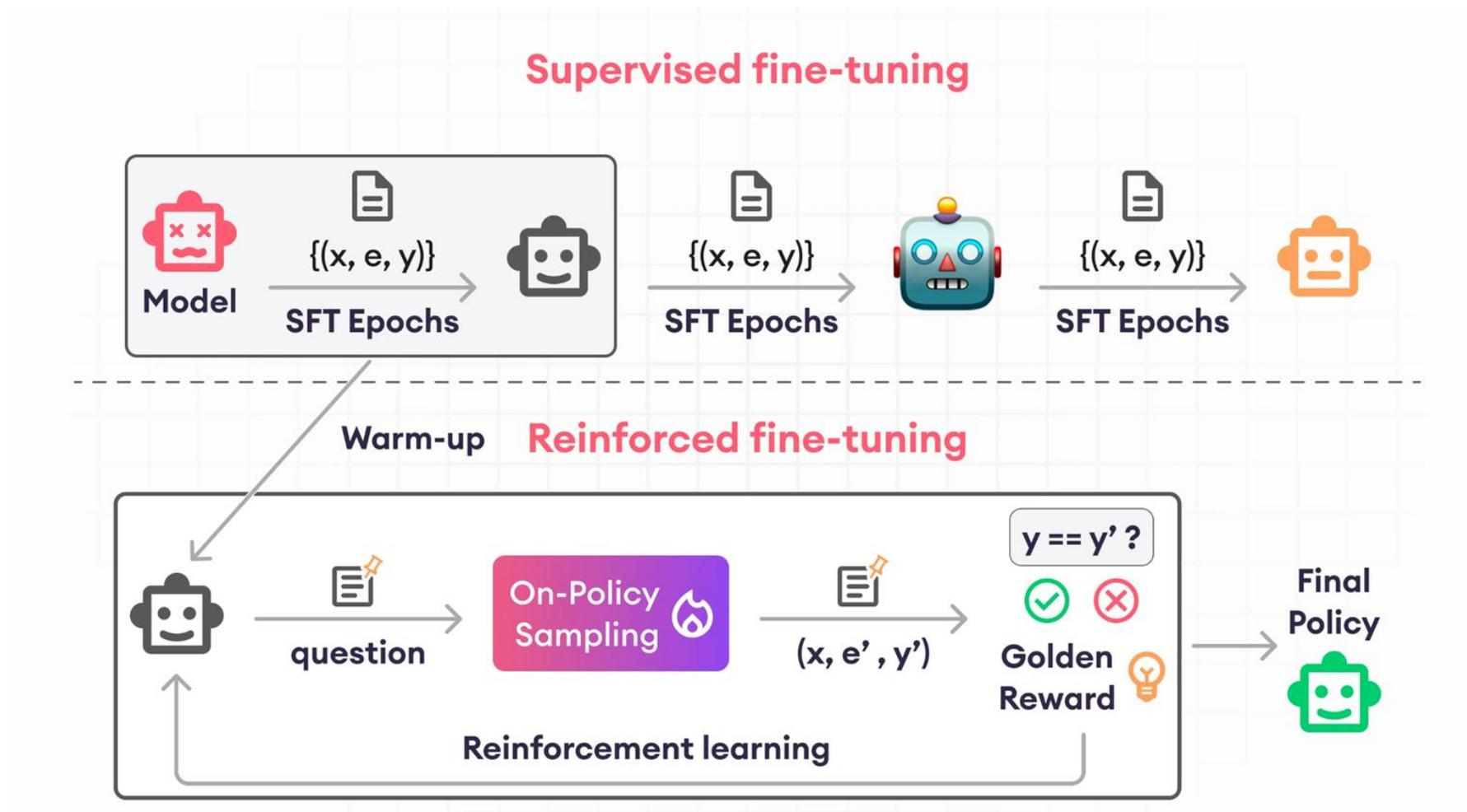


Web agents/  
GUI agents



Embodied AI agents

# Typical RL fine-tuning pipeline



# Embodied Reasoning Agent (ERA)

## ERA: TRANSFORMING VLMS INTO EMBODIED AGENTS VIA EMBODIED PRIOR LEARNING AND ON-LINE REINFORCEMENT LEARNING

### ABSTRACT

Recent advances in embodied AI highlight the potential of vision language models (VLMs) as agents capable of perception, reasoning, and interaction in complex environments. However, top-performing systems rely on large-scale models that are costly to deploy, while smaller VLMs lack the necessary knowledge and skills to succeed. To bridge this gap, we present *Embodied Reasoning Agent (ERA)*, a two-stage framework that integrates prior knowledge learning and on-line reinforcement learning (RL). The first stage, *Embodied Prior Learning*, distills foundational knowledge from three types of data: (1) Trajectory-Augmented Priors, which enrich existing trajectory data with structured reasoning generated by stronger models; (2) Environment-Anchored Priors, which provide in-environment knowledge and grounding supervision; and (3) External Knowledge Priors, which transfer general knowledge from out-of-environment datasets. In the second stage, we develop an online RL pipeline that builds on these priors to further enhance agent performance. To overcome the inherent challenges in agent RL, including long horizons, sparse rewards, and training instability, we introduce three key designs: self-summarization for context management, dense reward shaping, and turn-level policy optimization. Extensive experiments on both high-level planning (EB-ALFRED) and low-level control (EB-Manipulation) tasks demonstrate that ERA-3B surpasses both prompting-based large models and previous training-based baselines. Specifically, it achieves overall improvements of 8.4% on EB-ALFRED and 19.4% on EB-Manipulation over GPT-4o, and exhibits strong generalization to unseen tasks. Detailed Ablation studies further validate the effectiveness of different prior datasets and agent RL designs. Overall, ERA offers a practical path toward scalable embodied intelligence, providing methodological insights for future embodied AI systems.

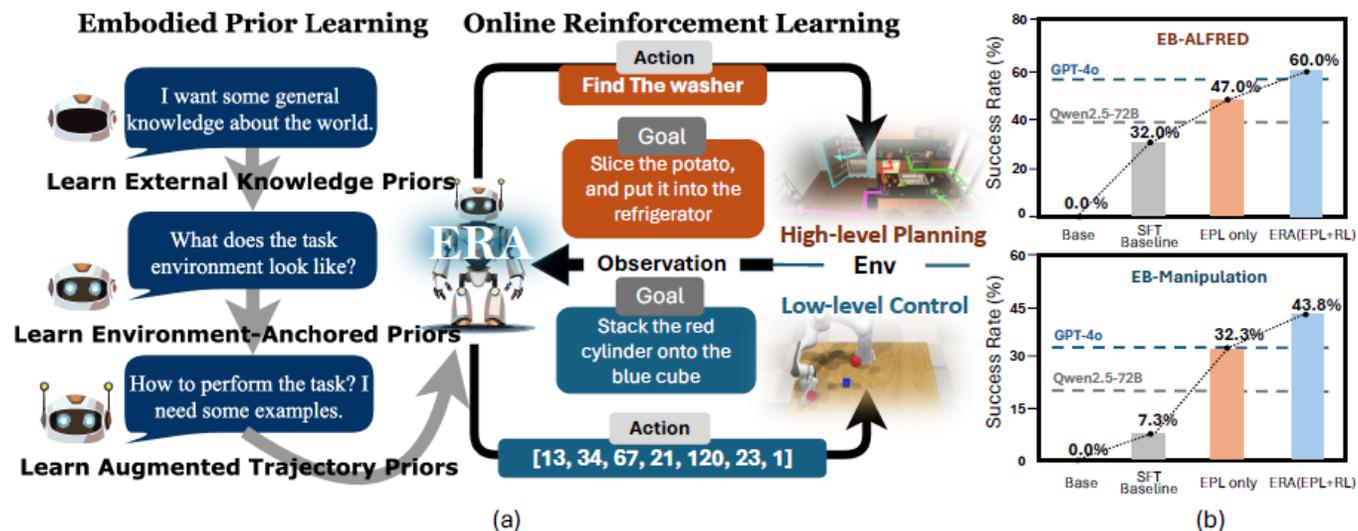


Figure 1: (a) Overview of the ERA framework: Embodied Prior Learning (EPL) finetunes on diverse data sources to provide foundational knowledge, and online RL further improves the agent. (b) ERA (i.e., EPL+RL) boosts a 3B base model to surpass GPT-4o on hold-out evaluation sets.

# Embodied Reasoning Agent (ERA)

Model	EB-Alfred						EB-Manipulation					
	Avg	Base	Complex	Visual	Common	Spatial	Avg	Base	Complex	Visual	Common	Spatial
<i>Prompting-based MLLMs</i>												
GPT-4o	56.8	64	68	46	54	52	28.9	39.6	29.2	19.4	29.2	25.0
Claude-3.5-Sonnet	<b>66.4</b>	<b>72</b>	<b>76</b>	60	<b>66</b>	58	25.4	37.5	29.2	19.4	16.7	22.9
Gemini-1.5-Pro	63.2	70	72	58	64	52	21.1	14.6	22.9	16.7	14.6	35.4
Gemini-2.0-flash	51.2	62	54	46	48	46	16.7	14.6	14.6	13.9	8.3	31.3
Llama-3.2-90B-Vision-Ins	35.2	38	44	28	34	32	14.9	10.4	16.7	10.4	12.5	20.8
InternVL3-78B	39.6	38	46	42	34	38	26.3	29.2	22.9	25.0	22.9	31.3
Qwen2.5-VL-72B-Ins	40.8	50	42	36	42	34	16.2	12.5	16.7	22.2	12.5	18.8
Qwen2.5-VL-7B-Ins	5.2	10	6	2	8	0	9.6	8.3	8.3	5.6	8.3	16.7
Qwen2.5-VL-3B-Ins	0	0	0	0	0	0	0	0	0	0	0	0
<i>Training-based MLLMs</i>												
RL4VLM (3B)	51.2	70	70	56	32	28	21.9	33.3	29.2	30.6	8.3	8.3
VAGEN (3B)	52.8	70	70	58	38	28	22.9	35.4	31.3	29.2	8.3	10.4
Reinforced Reasoner (7B)	41.6	54	46	28	42	38	-	-	-	-	-	-
Robot-R1 (7B)	-	-	-	-	-	-	11.7	12.5	6.3	16.7	8.3	14.6
<b>ERA-3B (EPL-only)</b>	56.0	68	66	52	44	50	40.0	45.8	41.7	47.9	37.5	27.1
<b>ERA-3B (EPL+RL)</b>	65.2	<b>72</b>	72	<b>62</b>	54	<b>66</b>	<b>48.3</b>	<b>56.3</b>	<b>47.9</b>	<b>50.0</b>	<b>47.9</b>	<b>39.6</b>

Finetuned 3B models can outperform GPT-4o!

# Reinforcement Learning: Key Take-aways

- Grounding (prior learning) should be combined with reinforcement learning for making small LLMs work in engineering domains.
- How to perform such a pipeline for analog IC design?
- For engineering design, the reward function can be set up using the domain-specific evaluation tools (similar to EngDesign).

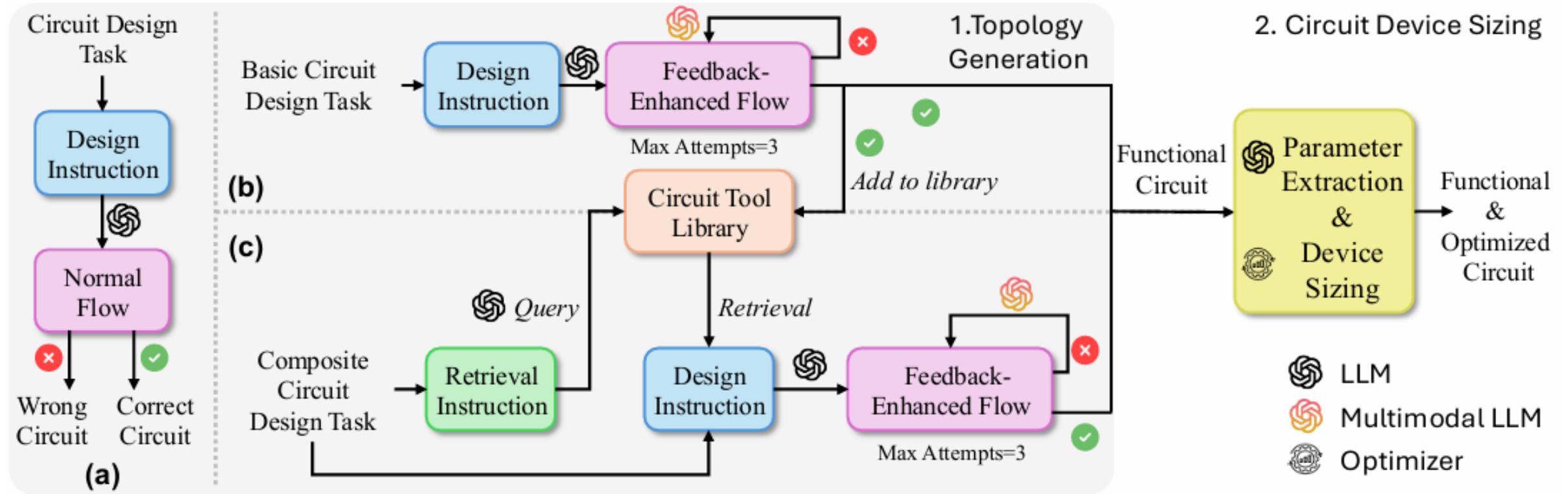
# Outline

- Toward Engineering AGI: EngDesign Benchmarking the engineering design capabilities of LLMs
- Past successful story I: LLM agents with test-time scaling for control design
- Past successful story II: Reinforcement learning for finetuning VLM for robotics
- Toward Building LLM-based Analog IC Design Engineers: Where are we right now?

# Existing Agents for Analog IC Design

AMSnet-KG: A Netlist Dataset  
Using Knowledge Graph RAG

## AnalogCoder-Pro: Unifying Analog Circuit



components are retrieved and assembled into a circuit netlist. Simulation results of the netlist are fed back to the LLM to refine the design. We perform case studies of operational amplifiers and comparators to netlists with minimal human effort. The dataset

CCS Concepts: • Hardware → Design databases

CS.LG

... and applying physical optimization for device sizing. On a curated benchmark suite covering 13 circuit types, AnalogCoder-Pro successfully designed 28 circuits and consistently outperformed existing LLM-based methods in figures of merit. The codes will be released at [github.com/laiyao1/AnalogCoderPro](https://github.com/laiyao1/AnalogCoderPro).

**Index Terms**—Analog circuit design, large language model, topology generation, device sizing.

AnalogGenie/Lite [36], [37]  
AnalogFed [38]  
AutoCircuit-RL [39]  
SPICEAssistant [40]  
AnalogCoder-Pro

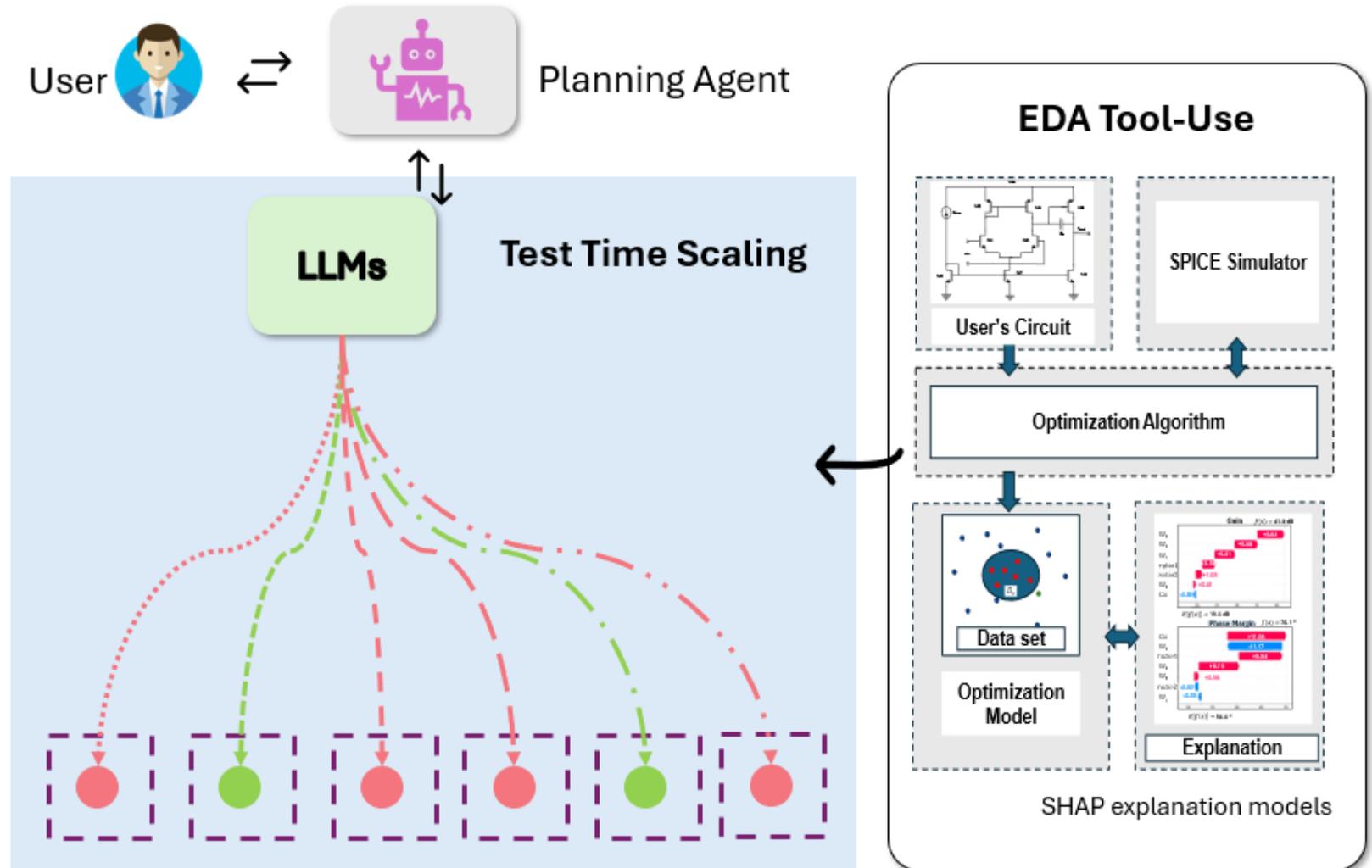
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•

<sup>1</sup> Whether the work supports multiple circuit type designs.

<sup>2</sup> • - Full multi-modal debugging capability by signal images (e.g., waveform, frequency response), ◦ - Text-only debugging.

# Our On-going Work: Test-Time Scaling

- **Test-time scaling:** Generating multiple reasoning paths and leveraging diverse sizing/evaluation tools
- **Integrate LLMs with SHAP-based EDA tools** to provide explainable design
- **Feedback agentic workflow for automated design**



# On-going Work: Reinforcement Finetuning

- Developing lightweight LLMs (3B~10B) that can be combined with circuit component databases to generate topology
- Grounding (prior learning) from new papers and SOTA reasoning models as well as existing datasets such as Masala-CHAI and AMSnet
- Train lightweight LLMs via reinforcement learning and EDA-based reward modeling

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^G \left( \min \left( \frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, \text{clip} \left( \frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1 - \varepsilon, 1 + \varepsilon \right) A_i \right) - \beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) \right),$$

$$\mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_{\theta}(o_i|q)} - \log \frac{\pi_{ref}(o_i|q)}{\pi_{\theta}(o_i|q)} - 1,$$



CAEML

# Toward Building LLM-Based Analog IC Design Engineers

- **A feedback agentic loop with LLM-based topology generation and tool-based sizing/evaluations**
- **Diverse tools are integrated for test-time scaling**
- **Grounding (prior learning) + reinforcement finetuning for training lightweight LLMs specialized for topology generation**
- **Grounding requires collecting data from multiple sources, and is the most time-consuming step so far**

# Summary

- Our EngDesign Benchmark shows that currently there is no single LLM that achieves engineering AGI.
- Our past study on ControlAgent shows test-time scaling with diverse tools can achieve automated LLM-based control design on real hardware systems.
- Prior learning+RL has shown promise as a general development pipeline.
- We believe that one plausible way to build LLM-based analog IC design engineers is to combine test-time scaling, various domain-specific tools (sizing, evaluations, etc), prior learning, and reinforcement learning.

# Acknowledgements

**Big thanks to CAEML and Amazon for funding our research!**

**Special acknowledgements to our industry collaborators: Chris Cheng (HPE), Priyank Kashyap (HPE), Gary Chiu (MediaTek), Chris Huang (MediaTek), Mahdi Namazifar (Amazon), Mayank Kulkarni (Amazon)**