CycleResearcher:
Improving Automated Research via Automated Review

Yixuan Weng*, Minjun Zhu*, Guangsheng Bao, Hongbo Zhang, Jindong Wang,
Yue Zhang, Linyi Yang
(wengsyx@gmail.com)

Westlake University

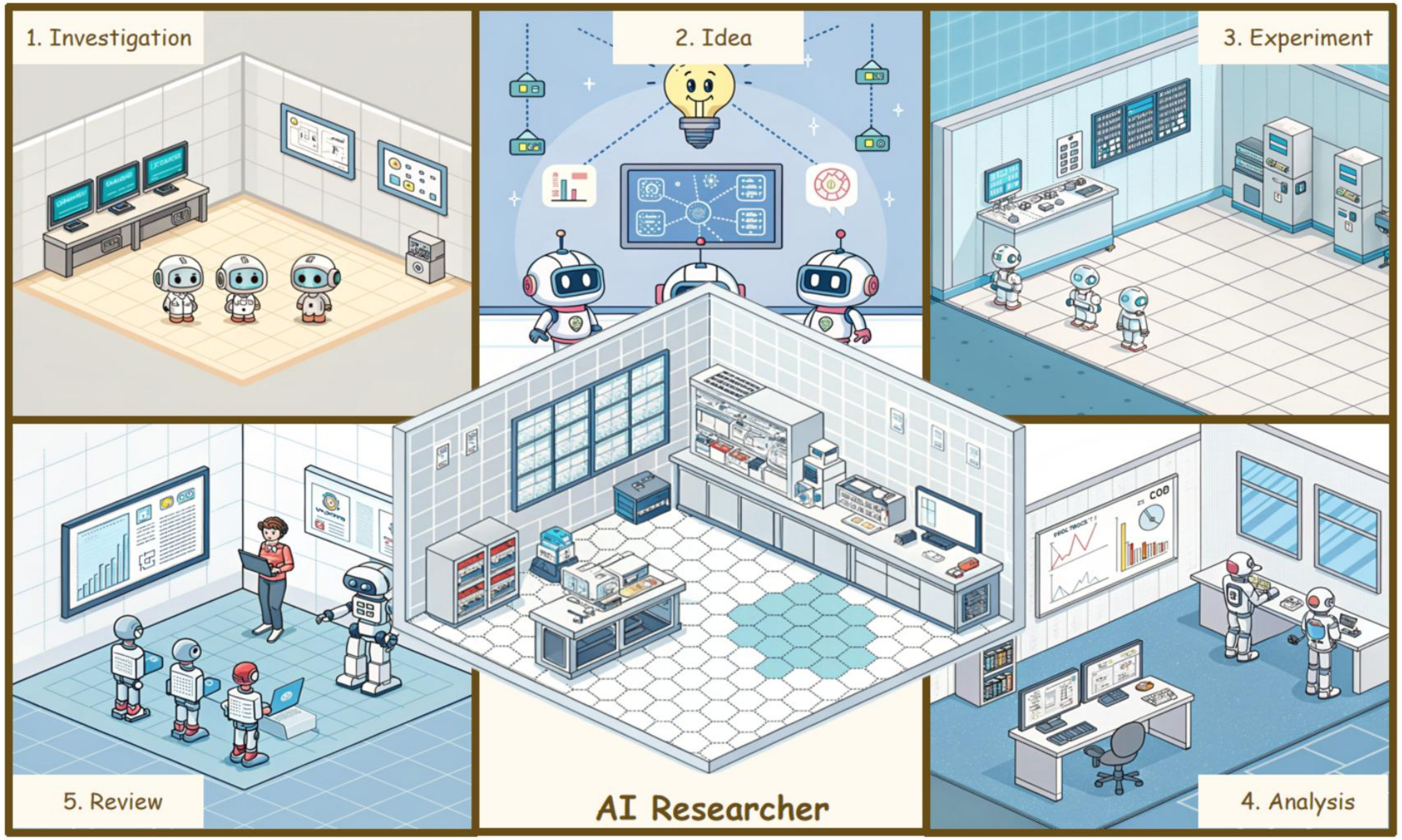
https://ai-researcher.net

Motivation & Challenges

The Quest for Automated Science

Scientific discovery automation:

- LLMs show promise as research assistants
- Gap: Full research lifecycle automation
- Challenge: Maintaining academic quality



Research Question & Contributions

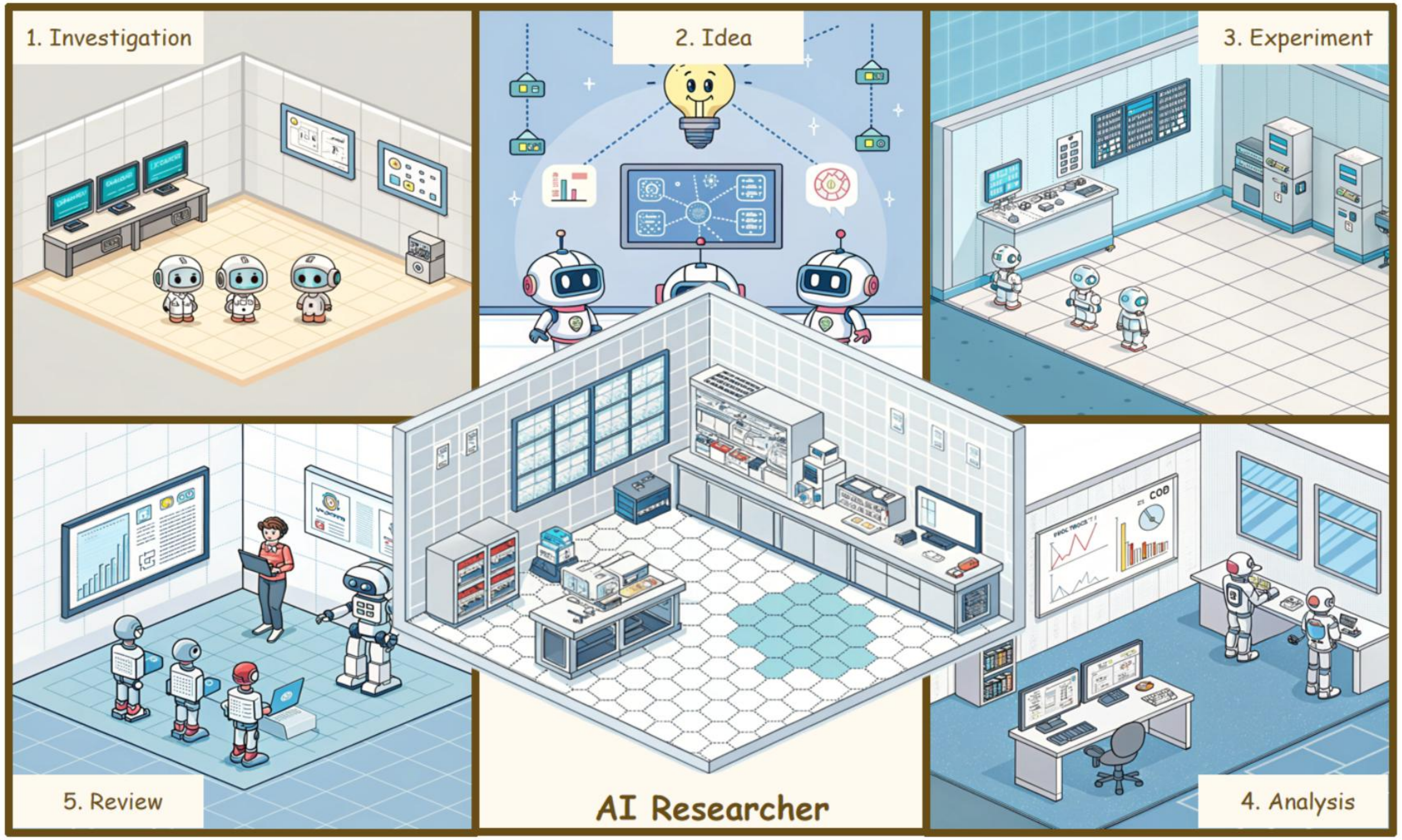
How can we automate the Research-Review-Refinement process by post-training LLMs?"

Dataset

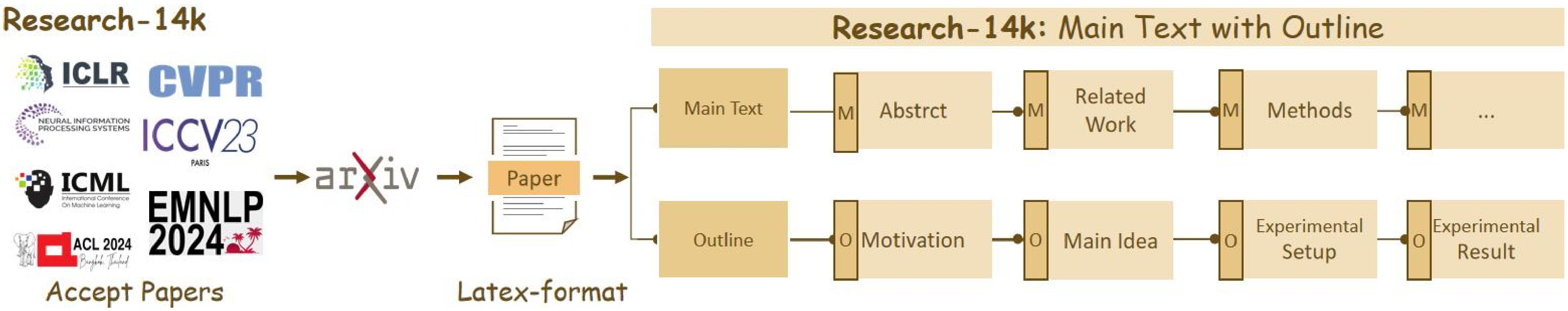
- Researcher-14K
- Review-5K

Iterative Reinforcement Learning framework

- **Reward model**
  - DeepReviewer-7B
  - DeepReviewer-14B
  - CycleReviewer-123B
- **Policy model**
  - CycleResearcher-12B
  - CycleResearcher-72B
  - CycleResearcher-123B



Research-14k: Data Collection and Supervised Fine-tuning



Main text (M): Full paper content
Outlines (O): Structured paper planning

Combined as a complete pipeline context for training

Research-14K

- ***~14000*** Accepted Papers
- ***~28000*** Tokens per Paper
- From 2022 to 2024.

CycleResearcher Warmup: SFT

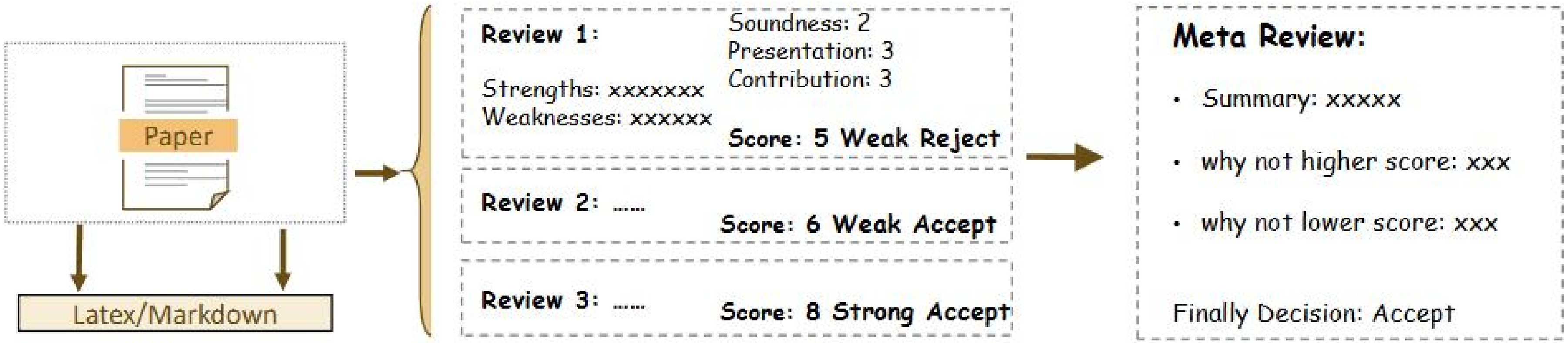
- ***~1 month*** for 4 x 8 H100
- Full Size: ***7B/72B/123B***

Research-14K: https://huggingface.co/datasets/WestlakeNLP/Research-14K

Review-5k

Review-5K

- ***4,991*** papers
- ***16000+*** reviewer comments
- Complete review components
  - summary,
  - strengths,
  - weaknesses,
  - scores



Review-5K: https://huggingface.co/datasets/WestlakeNLP/Review-5K

CycleReviewer: Training and Experiment Result

CycleReviewer-123B

- Based on Mistral-Large-2 123B
- ***26.89%*** reduction in MAE
- ***74.24%*** decision accuracy

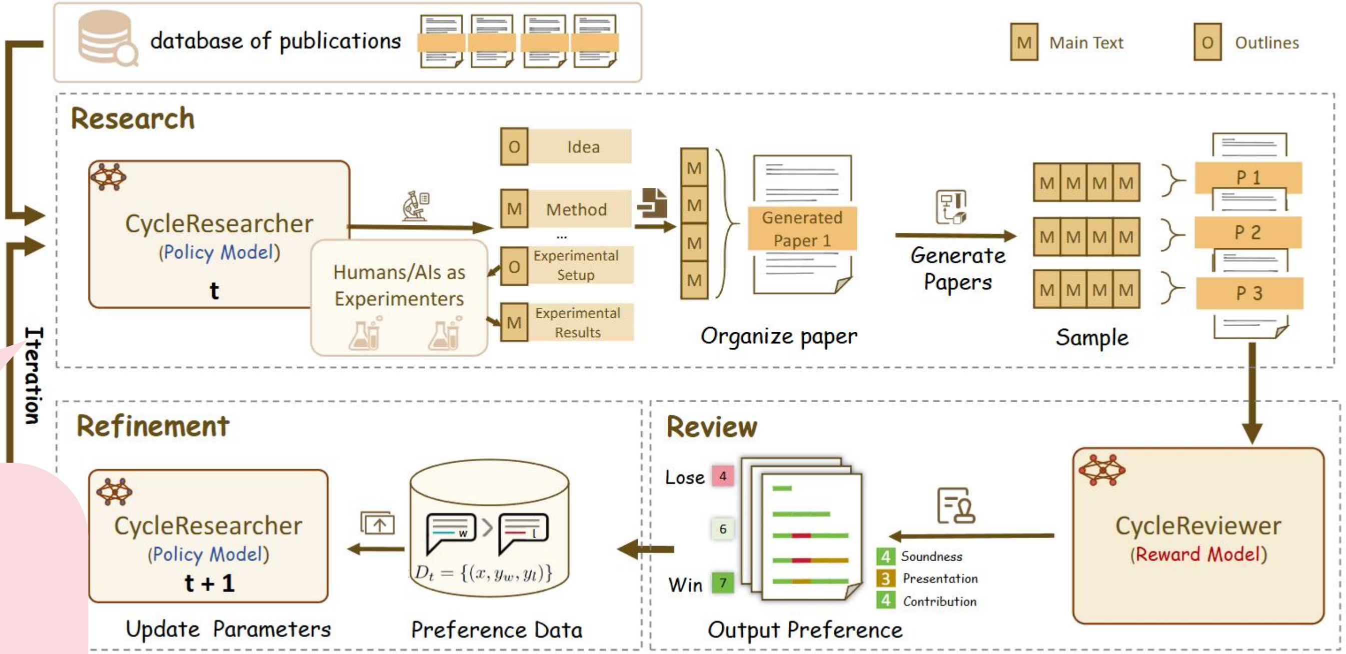
Method	Proxy (Reviewer=$n-1$) MSE ↓	Proxy (Reviewer=$n-1$) MAE ↓	Proxy (Reviewer=n) MSE ↓	Proxy (Reviewer=n) MAE ↓	Decision Accuracy ↑	Decision Macro F1 ↑
Human Expert Individual	2.34	1.16	-	-	**75.40%**	**75.39**
GPT-4o-mini	3.44	1.53	2.98	1.40	53.06%	34.72
GLM-4	4.45	1.81	3.91	1.70	49.49%	33.10
DeepSpeak-2.5	4.62	1.83	3.72	1.64	45.11%	39.98
Gemini-1.5-pro	3.02	1.34	2.56	1.23	50.98%	50.75
Claude-3.5-Sonnet	6.40	2.23	5.62	2.12	48.05%	32.44
GPT-4o	6.61	2.24	6.53	2.30	52.58%	34.51
CycleReviewer (123B)	**1.43**	**0.92**	**1.25**	**0.87**	**74.24%**	**73.99**

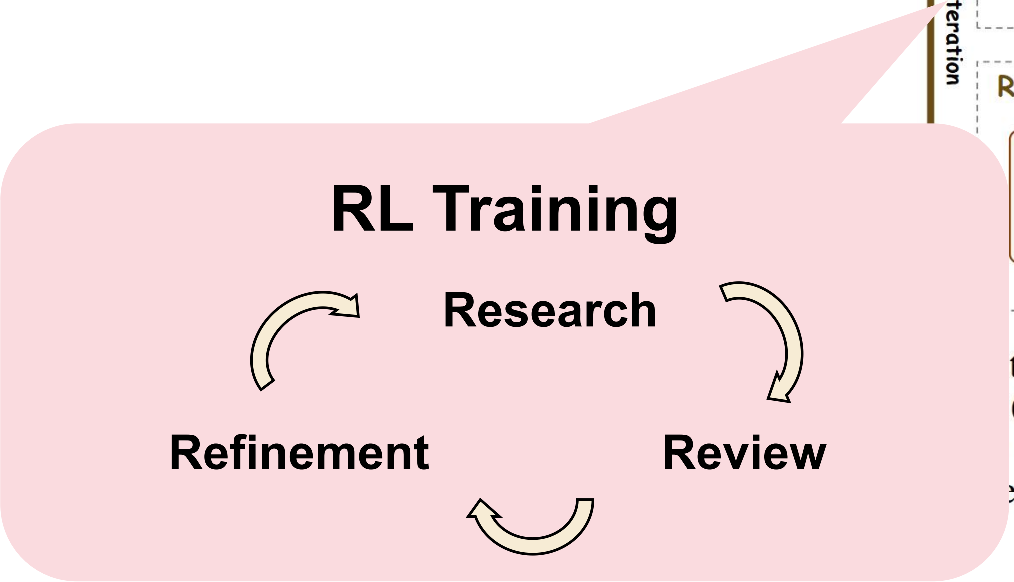
CycleResearcher
RL Training: Research-Review-Refinement Cycle

RL Train

- *Iterative SimPO*

$$\mathcal{L}_{\text{Our}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x) - \gamma\right)\right]$$
$$-\lambda\mathbb{E}_{(x,y_w)\sim\mathcal{D}_{\text{NLL}}}\left[\log\pi_\theta(y_w \mid x)\right].$$





terative Training Framework. The CycleResearcher model generates Outline (O) and (M) to organize papers, which are evaluated by the CycleReviewer and constructed into pairs based on rewards. This whole procedure is then iteratively refined, resulting in ely enhanced research abilities with each iteration.

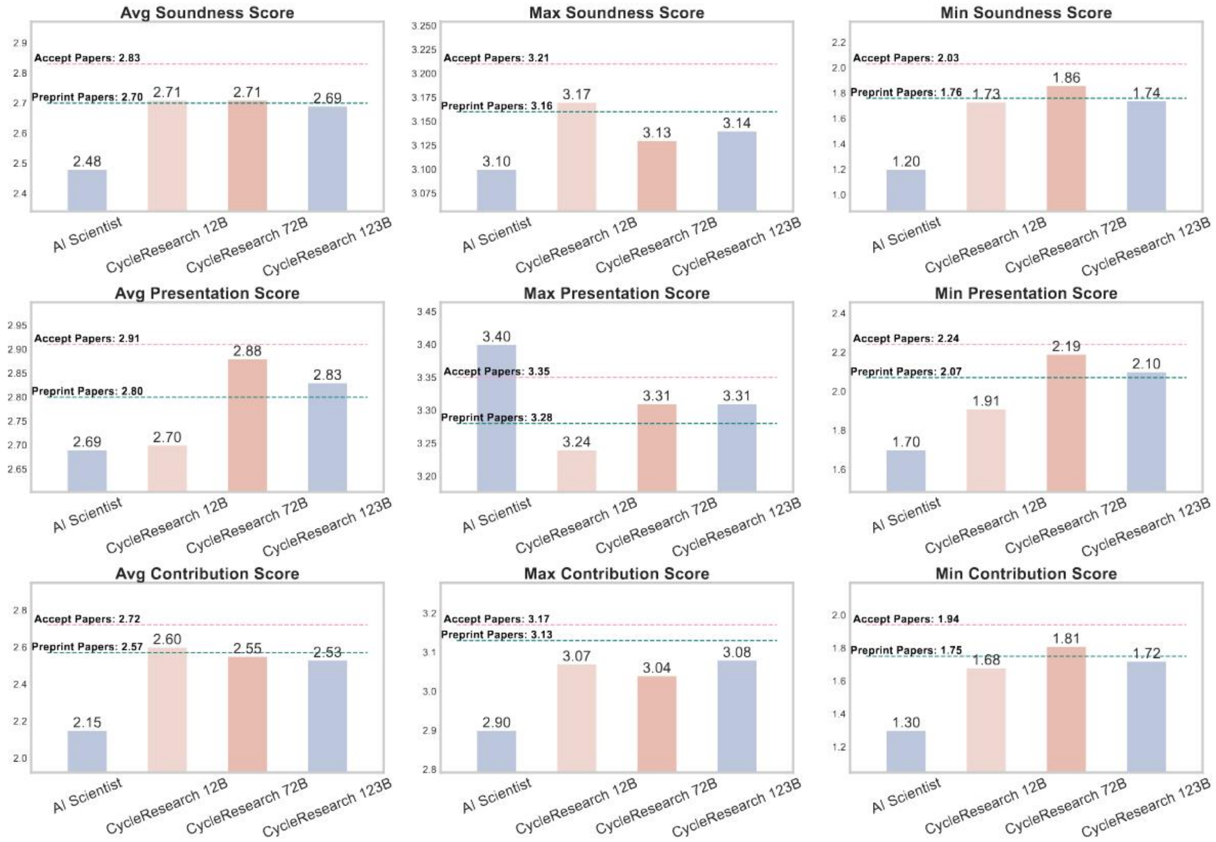
CycleResearcher: Main Experimental Findings

Paper Type	Source	Overall Score Metrics			Accept Rate
		Avg Min Score ↑	Avg Max Score ↑	Avg Score ↑	
Conference Accept Papers	Human Expert	**3.91**	**6.98**	**5.69**	**100.00%**†
Preprint Papers	Human Expert	3.24	6.62	5.24	29.63%
AI Scientist	AI	2.20	5.70	4.31	*0.00%*
CycleResearcher-12B (Ours)	AI	3.47	**6.75**	5.36	**35.13%**
CycleResearcher-72B (Ours)	AI	**3.65**	6.58	**5.38**	33.64%
CycleResearcher-123B (Ours)	AI	3.30	6.45	5.15	24.28%

- **Results:**
  - CycleResearcher scores closer to human preprints than AI Scientist.
  - Generated papers show competitive quality metrics.
  - Score gap between AI and human experts narrowing.

CycleResearcher: Main Experimental Findings

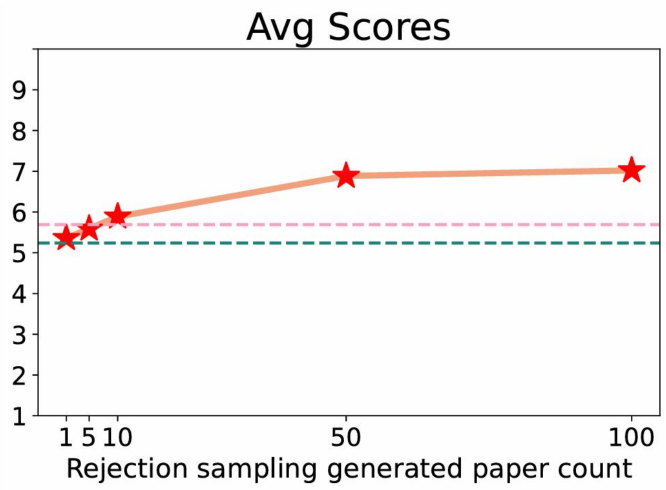
- **Results:**
  - Outperform AI Scientist across all metrics
  - 72B model excels in presentation scores
  - 12B model shows strongest contribution scores
  - All models approach preprint-level soundness



Ablation Studies & Analysis

Method	Avg Score ↑	Accept Rate ↑
CycleResearcher	**5.36**	**35.14%**
w/o RL	(−0.24) 5.12	(−5.34%) 29.80%
w/o Iterative	(−0.15) 5.21	(−2.23%) 32.91%
w/o NLL	(−0.45) 4.91	(−23.11%) 12.03%

Ablation studyof different variations of CycleResearcher-12B.



Performance improvement through rejection sampling in generated papers

- **Results:**
  - RL significantly improves performance (+5.34% acceptance rate)
  - Iterative training enhances quality (+2.23% acceptance rate)
  - NLL stabilizes training (critical for coherence)
  - Rejection sampling further improves quality

Academic Integrity & Ethics

Detection Performance

Model	Format	Accuracy	F1 Score
CycleReviewer-123B	Review	95.14%	94.89
CycleResearcher-12B	Research	98.38%	98.37
CycleResearcher-72B	Research	97.52%	97.49
CycleResearcher-123B	Research	98.88%	98.87

- **Results:**
  - >95% accuracy in detecting AI-generated content
  - Embedded watermarks in all outputs
- **Ethical Safeguards:**
  - Clear disclosure requirements for all users
  - Institutional affiliation verification
  - Prohibited use in real peer review without disclosure
  - Guidelines for responsible usage in academia

Example for CycleResearcher:

Real-world process: Search + Code Generation + Rxperiments + Paper Writing

Unveiling Generalization Gaps: A Quantitative Analysis of Neural Network Learning Dynamics

CycleResearcher

Abstract

Deep neural networks exhibit varying behaviors during training, from predictable performance improvements to unexpected phenomena like grokking. Understanding these behaviors is crucial for developing reliable AI systems. We propose the "generalization gap" framework to analyze neural network learning dynamics through controlled experiments on synthetic algorithmic tasks. Our study quantifies this gap between training and validation performance across different architectures and hyperparameters. Through systematic experimentation, we demonstrate how the generalization gap characterizes distinct learning phases and predicts generalization behavior. Our experiments span multiple network configurations, showing consistent patterns in how the gap evolves during training. The results provide empirical evidence that studying generalization gaps offers valuable insights into neural network learning dynamics and potential predictors of model performance.

1 Introduction

The rise of deep learning has brought remarkable advances alongside puzzling phenomena tha challenge our understanding of how neural networks learn. While certain behaviors, such as improve performance with increased data or parameters, follow predictable patterns, others remain enigmati Among these, "grokking" (Power et al., 2022) - where models transition from apparent overfitting t sudden generalization - exemplifies the complex dynamics that emerge during training. Understandin these learning phenomena has become increasingly crucial as neural networks grow in scale an capability. When models exhibit unexpected behaviors like grokking or emergent abilities (Wei et al 2022), traditional metrics often fail to provide adequate insights into the underlying mechanism This limitation highlights the need for more sophisticated analytical frameworks that can characteriz and predict such behaviors.

The generalization gap - the difference between training and validation performance - offers a promis ing lens through which to study these phenomena. While previous work has explored various aspect of neural network generalization (?), our approach uniquely focuses on using this gap as a quantitativ tool for analyzing learning dynamics. Through systematic experimentation, we demonstrate how this metric can reveal distinct phases in the training process and predict generalization behavior. Ou experimental methodology centers on controlled studies using synthetic algorithmic tasks, allowin for precise manipulation of network parameters and training conditions. We examine how variou factors - including network architecture, optimization parameters, and regularization techniques influence the generalization gap. This comprehensive approach enables us to isolate and analyz specific aspects of learning behavior while maintaining experimental rigor.

Our primary contributions include:

- Development of a quantitative framework using generalization gaps to analyze neura network learning dynamics
- Extensive empirical validation across diverse architectural configurations and training pa rameters
- Demonstration of the generalization gap's effectiveness in predicting model performance
- Analysis of how various hyperparameters influence learning trajectories and generalizatio behavior

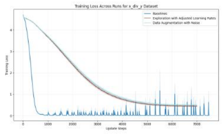


Figure 1: (a) Training and validation loss trajectories for division tasks

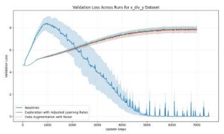


Figure 2: (b) Training and validation accuracy for division tasks

Figure 3: **Training Dynamics Comparison**. Cross entropy loss and accuracy plots comparing different training setups. Shows clear separation between training and validation performance, with characteristic grokking behavior visible in loss curves.

The generalization gap, as defined in Section 3 is computed as the absolute difference between training and validation losses:

$$\text{Gap} = |\mathcal{L}^{cross}_{train} - \mathcal{L}^{cross}_{val}|. \tag{3}$$

Our analysis reveals distinct phases of training, with varying generalization gap behavior across phases. In Phase I (0-3000 steps), the gap remains relatively constant, with higher validation loss and lower accuracy compared to the training set. Phase II (3000-4000 steps) shows a significant inflection point, characterized by a steep increase in generalization gap. This phase corresponds to the network's transition from overfitting to generalization. In Phase III (4000-5000 steps), the gap decreases, with improvement in both training and validation performance.

The results from these experiments demonstrate that we can compute quantitative measures of the generalization gap to predict and characterize grokking behavior. Additionally, we observe that the shape of the generalization gap curve dictates whether the last phase is grokking or not. Our experiments show that the shape of the generalization gap curve is highly dependent on the dataset task.

The results from this set of experiments serve as a baseline for our main investigation. Using these learned parameters, we explore how different factors - like architecture, training data, and regularization - influence the generalization gap and overall model performance. Our experiments provide valuable insights into the conditions under which grokking occurs and the complex interplay of factors that affect its emergence.

4.3 Extended Dataset Evaluation

Here, we double the dataset size for each task to evaluate its impact on the generalization gap. The dataset now consists of 600,000 training examples for each task. The results are summarized in Table 2.

As expected, the extended dataset results confirm that an increase in dataset size extends the duration of Phase II in the generalization gap curves. This extends the network's phase of "learning to generalize" and effectively prevent it from overfitting to noise in the dataset. Additionally, the inflection point and area metrics show consistent relative values across different tasks for a given network.

Task	Peak	Inflection Point	Area	Length
x_div_y	4.695	70.0	179.13	673.67
x_minus_y	4.693	70.0	185.32	663.00
x_plus_y	4.702	67.33	164.43	656.33
permutation	4.929	65.0	290.80	669.67

Table 2: Generalization gap characteristics for different tasks.

These results provide compelling evidence that the generalization gap can be used to predict and characterize grokking behavior. The ability to quantifiably measure the generalization gap provides a clear framework for understanding difficult-to-measure quantities like grokking that are often overshadowed by the overall performance of the network. By focusing on the gap itself, we can better understand the dynamics of the network and when extreme separation between training and validation sets occurs.

4.4 Generalization Gap Analysis

Our study focused on the following generalization gap metrics to provide insights into generalization behavior.

Peakness measures the peak generalization gap value during training:

$$\text{Peakness} = \max_{t \in T} |\mathcal{L}^{cross}_{train}(t) - \mathcal{L}^{cross}_{val}(t)|. \tag{4}$$

Inflection Point identifies when the generalization gap transition occurs:

$$\text{Inflection Point} = t \text{ where } |\mathcal{L}^{cross}_{train}(t) - \mathcal{L}^{cross}_{val}(t)|'' > \epsilon. \tag{5}$$

In our experiments, the gradient threshold (ϵ) is set to 0.01.

Area quantifies the cumulative measure of the inflection phase:

$$\text{Area} = \int_{t_1}^{t_2} |\mathcal{L}^{cross}_{train}(t) - \mathcal{L}^{cross}_{val}(t)| dt. \tag{6}$$

where t_1 and t_2 define the phase where the gap metrics meet the Inflection Point condition.

Length measures the duration of phase transitions:

$$\text{Length} = (t_2 - t_1). \tag{7}$$

Using these metrics, we perform an in-depth analysis of the generalization gap's formation and evolution. The results from this analysis are summarized in ?? and Table 3.

Configuration	Peakness	Inflection Area	Length
Baseline	4.736	2523.81	988.0
Tuned LR	4.732	696.59	973.67
With Dropout	4.731	740.43	985.67
Final	4.699	2324.55	1389.67

Table 3: Generalization gap metrics are largely consistent across different architectural configurations.

In ??, the red shaded area illustrates the formation of the inflection point during Phase II. This formation marks the separation between Phase I (high validation loss) and Phase III (lower validation loss). From the results, we observe that peakness measurements reach its peak at the end of Phase II. This observation aligns with our main results, which show that the network begins to separate during this phase. In ??, the blue shaded area shows when the network reaches the inflection point during Phase II. The end of this phase signals the transition from overfitting to generalization. These metrics provide valuable insights into the dynamics of the network and when extreme separation between training and validation sets occurs.

Our experiments reveal distinct patterns in training dynamics. The generalization behavior differs significantly between the two loss functions. Varying loss functions inherently result in differences in generalization dynamics. The empirical evidence confirms the influence of the loss function on generalization and is characterized by loss differences during the inflection.

4.9 Regularization Effect Analysis

In this study, we aim to understand the effects of regularization on model performance. We focus on two specific regularization techniques, namely weight decay and dropout. Our experiments maintain the "tuned" configuration and apply different regularization parameters. The results are summarized in Figure 12.

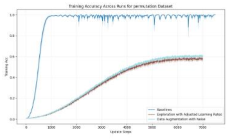


Figure 10: (a) Training accuracy

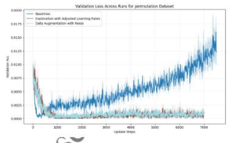


Figure 11: (b) Validation accuracy

Figure 12: **Regularization Impact**: (a) Training accuracy comparisons across regularization settings (b) Validation accuracy comparisons across regularization settings.

Our findings suggest that dropout has a lower generalizing effect than weight decay in our architecture. This outcome is consistent with previous literature that has highlighted the shorter distance in the hidden layers between inputs in the Transformer architecture (Geva et al., 2020). The results indicate that weight decay and combined configurations exhibit near-random network performance, by which we mean that the accuracy on the validation set is approximately the same as the accuracy on a randomly generated key.

Config	Train Acc	Val Acc	Gen Gap
No Reg	0.9776	0.0151	0.9625
Weight Decay	0.9724	0.0046	0.6678
Dropout	0.9961	0.1141	0.8820
Combined	0.5784	0.0011	0.5773

Table 6: Regularization effects on generalization.

In Table 6, we summarize our network's overall performance and generalization gap calculations. Notably, weight decay enhances validation accuracy compared to the baseline configuration but increases the generalization gap. Combined regularization schemes, however, reduce the generalization gap, though at the expense of overall performance. These results highlight the nuanced influence of regularization on model performance and generalization, offering valuable insights for practitioners. Placing too much emphasis on the generalization gap can lead to suboptimal model performance. Our results provide practical guidelines for balancing these objectives.

5 Discussion

Conclusion Our work successfully establishes the "generalization gap" as a way of mathematically characterizing grokking using simple synthetic algorithmic tasks. By focusing on a small set of

WESTLAKE UNIVERSITY 西湖大學

https://ai-researcher.net

AI-Researcher

http://ai-researcher.cn

Open Sources

Datasets

Review-5K

Comprehensive collection of peer reviews from major conferences.

Download →

Research-14K

Large-scale dataset of research papers and annotations.

Download →

DeepReview-13K

Structured dataset for training review systems.

Download →

Models

CycleResearcher Models

CycleResearcher-12B
CycleResearcher-72B
CycleResearcher-123B

Access Models →

DeepReviewer Models

DeepReviewer-8B
DeepReviewer-14B
DeepReviewer-70B

Access Models

CycleReviewer Models

Cycle Reviewer-8B
Cycle Reviewer-70B
Cycle Reviewer-123B

Access Models →

Code

CycleResearcher

Core implementation of the AI Researcher system.

View Code →

DeepReviewer

Implementation of the Deep Review system.

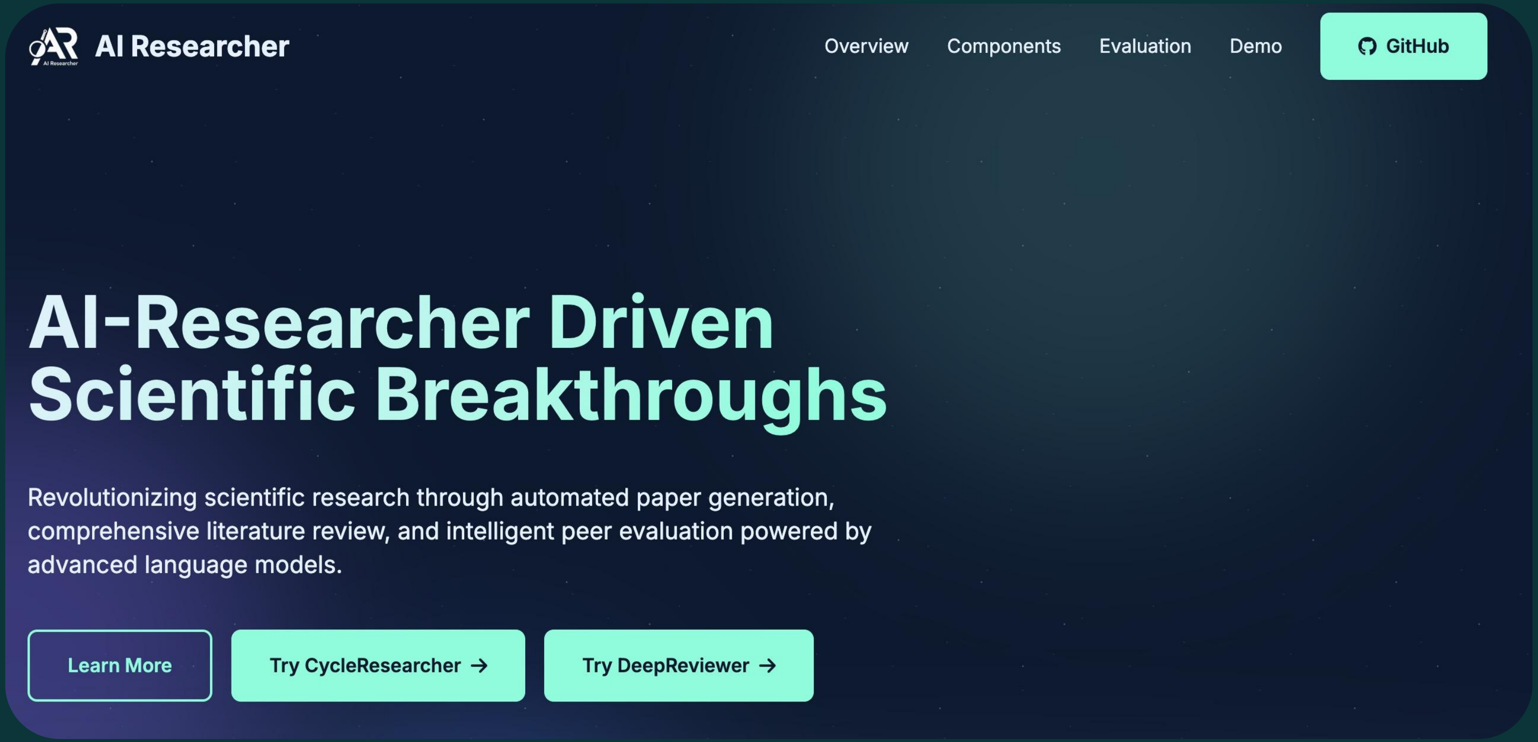
View Code →

CycleReviewer

Implementation of the Review Cycle system.

View Code →

Thanks!



Contact: wengsyx@gmail.com

Project: https://ai-researcher.net