Research Report: Recursive Self-Aggregation (RSA)

Deep Thinking and Test-Time Scaling for LLM Reasoning

Executive Summary

Recursive Self-Aggregation (RSA) is a groundbreaking test-time scaling method that enables smaller language models to match the performance of much larger reasoning models. By combining parallel and sequential scaling through iterative aggregation of reasoning chains, RSA allows a 4B parameter model (Qwen3-4B-Instruct) to achieve competitive performance with models like DeepSeek-R1 and o3-mini—without using external verifiers or complex tools.

Key Achievement: RSA demonstrates that intelligent use of test-time compute can dramatically enhance model performance, enabling smaller models to "think deeper" during inference rather than requiring larger model sizes.

The Problem

Challenge 1: Limitations of Current RL Training

During post-training, Large Language Models are trained using Reinforcement Learning to improve reasoning ability. However, **standard RL training has a critical flaw**: it doesn't account for test-time scaling—the practice of aggregating multiple reasoning chains during inference.

Surprising Discovery: The research found that standard RL fine-tuning can actually **degrade performance** relative to the base model when combined with test-time aggregation. This means that models trained with traditional RL may perform worse when we try to leverage multiple solution attempts.

Challenge 2: Parallel vs. Sequential Scaling Trade-offs

Existing test-time scaling approaches face a dilemma:

- **Parallel Scaling**: Generate multiple independent solutions and choose the best (e.g., majority voting)
  - Limited by the quality of individual attempts
  - Doesn't learn from intermediate steps
- **Sequential Scaling**: Iteratively refine a single solution through self-correction
  - Can get stuck in local optima
  - Doesn't benefit from diversity of multiple reasoning paths

---

The RSA Solution

Core Innovation: Hybrid Evolutionary Approach

RSA elegantly combines the benefits of both parallel and sequential scaling through an evolutionary-inspired process that:

1. Maintains diverse populations of reasoning chains
2. Iteratively refines solutions by aggregating subsets
3. Bootstraps from partially correct intermediate steps across different chains

Think of it as **"natural selection for reasoning chains"**—the method preserves good reasoning patterns while gradually pruning away errors and inconsistencies.

How RSA Works: The Algorithm

RSA operates through four key steps repeated across T iterations:

1. Population Initialization

- Generate N independent candidate solutions for a given query
- Each solution represents a complete chain-of-thought reasoning process
- Initial population ($\mathcal{P}_1$) created by sampling from the LLM

2. Subsampling

- Form N aggregation sets, each containing K candidates
- Sample uniformly without replacement from the current population
- This ensures diversity while allowing cross-pollination of ideas

3. Aggregation (The Core Innovation)

- Each set of K candidates is formatted with an aggregation prompt
- The LLM generates a refined response by combining insights from multiple chains
- Creates new population $\mathcal{P}$_{t+1} of improved candidate solutions
- **Key Insight**: The model doesn't just look at final answers—it learns from intermediate reasoning steps across different chains

4. Termination

- After T iterations, obtain final population $\mathcal{P}$_T
- Extract solution via random sampling or majority voting
- Research uses uniform random sampling to evaluate method without special selection mechanisms

Expected Behavior

As iterations progress (t increases):

- **Diversity decreases**: Populations converge toward better solutions
- **Success rate increases monotonically**: Each iteration improves overall quality
- **Errors are pruned**: Inconsistencies gradually eliminated through aggregation
- **Good patterns preserved**: Favorable reasoning approaches reinforced

---

Aggregation-Aware Reinforcement Learning

The Training Innovation

To fully unlock RSA's potential, the researchers developed a novel training approach: **aggregation-aware RL**.

The Problem with Standard RL

Standard RL doesn't prepare models to effectively combine multiple reasoning chains. This creates a mismatch between training and deployment.

The Solution

Data augmentation strategy that trains LLMs specifically to:

- Recognize patterns across multiple solution attempts
- Synthesize insights from partially correct reasoning chains
- Generate improved solutions through aggregation

This training approach leads to **significant performance gains** beyond what RSA alone achieves, creating a compounding effect where better training enables better test-time aggregation.

Performance Results

Impressive Benchmark Performance

RSA demonstrates substantial improvements across diverse tasks:

Mathematics & Competition Problems:

- **AIME-25** (American Invitational Mathematics Examination)
- **HMMT-25** (Harvard-MIT Mathematics Tournament)
- **Reasoning Gym** (diverse reasoning challenges)

Coding:

- **LiveCodeBench-v6** (real-world coding tasks)

Knowledge & Reasoning:

- **SuperGPQA** (graduate-level knowledge questions)

The Breakthrough Result

Qwen3-4B-Instruct-2507 (a 4 billion parameter model) achieves competitive performance with:

- **DeepSeek-R1** (much larger reasoning-specialized model)
- **o3-mini (high)** (OpenAI's compact reasoning model)

This demonstrates that **test-time compute can compensate for model size**, potentially democratizing access to advanced reasoning capabilities.

Comparison with Other Scaling Methods

RSA outperforms:

- **Purely parallel scaling** (e.g., simple majority voting)
- **Purely sequential scaling** (e.g., iterative self-refinement)
- Consistently across all tested benchmarks and model families

---

Key Insights & Implications

1. Test-Time Compute Matters More Than We Thought

The ability of a 4B model to match much larger models suggests we may have been overemphasizing model size at the expense of inference-time computation strategies.

2. The Aggregation Gap

The discovery that standard RL degrades test-time aggregation performance reveals a fundamental misalignment in how we train models versus how we deploy them. This suggests many existing models may be underperforming their potential.

3. Hybrid Approaches Win

By combining parallel diversity with sequential refinement, RSA achieves benefits neither approach delivers alone. This hybrid philosophy may extend to other AI domains.

4. Evolutionary Algorithms for AI

RSA's success demonstrates that evolutionary principles (selection, variation, inheritance) can be effectively applied to reasoning processes, not just genetic algorithms.

5. Practical Accessibility

RSA is described as "simple to implement" without requiring:

- Complex scaffolding
- External verifiers

- Specialized tools This makes it accessible for widespread adoption.

6. Democratization Potential

If smaller models can match larger ones through better inference strategies, this could:

- Reduce computational costs
- Lower barriers to entry for AI applications
- Enable deployment on resource-constrained devices

Technical Advantages

1. Information-Rich Reasoning

Unlike methods that only use final answers, RSA exploits the **complete reasoning chains**, capturing valuable intermediate insights.

2. Bootstrapping from Partial Correctness

Even if individual chains have errors, RSA can extract and combine correct intermediate steps from multiple chains, building better solutions incrementally.

3. No External Dependencies

The method works without:

- Verifier models
- External tools or APIs
- Human-in-the-loop feedback

4. Scalable Compute-Performance Trade-off

Performance improves systematically with increased compute budget (more iterations, larger populations), giving users control over the accuracy-efficiency trade-off.

5. Model-Agnostic

RSA works across:

- Different model families (Qwen, potentially others)
- Various model sizes (demonstrated on 4B)
- Diverse task types (math, coding, knowledge)

Practical Implementation Considerations

Compute Requirements

- **Population size (N)**: Number of parallel solutions to maintain

- **Aggregation size (K)**: Number of candidates combined per iteration
- **Iteration count (T)**: Depth of recursive refinement
- Trade-off between performance gains and computational cost

Best Practices

1. **Start with baseline parallel scaling** to establish performance floor
2. **Gradually increase iterations** to find optimal compute budget
3. **Use aggregation-aware training** for best results
4. **Monitor diversity metrics** to ensure population doesn't converge prematurely

---

Limitations & Future Directions

Potential Limitations (Inferred)

1. **Computational Cost**: Multiple iterations with large populations require significant inference compute
2. **Task Specificity**: Performance may vary across different domain types
3. **Training Requirements**: Aggregation-aware RL requires additional training infrastructure
4. **Convergence Concerns**: Very complex problems might need many iterations

Future Research Opportunities

1. **Optimal hyperparameters**: Systematic study of N, K, T across tasks
2. **Alternative aggregation strategies**: Beyond simple prompting
3. **Cross-model aggregation**: Combining outputs from different model families
4. **Application to other domains**: Vision, multimodal reasoning, etc.
5. **Real-time deployment**: Optimizing for latency-sensitive applications
6. **Theoretical analysis**: Formal convergence guarantees and complexity bounds

---

Comparison with Related Work

vs. Self-Consistency

- **Self-Consistency**: Sample multiple solutions, take majority vote
- **RSA**: Iteratively refines population through aggregation, learns from intermediate steps

vs. Self-Refinement

- **Self-Refinement**: Sequential improvement of single solution
- **RSA**: Maintains population diversity while refining through cross-aggregation

vs. LADDER

- **LADDER**: Recursive problem decomposition for self-improvement during training
- **RSA**: Test-time scaling method for inference, complementary approach

---

Conclusion

Recursive Self-Aggregation represents a significant advance in making LLM reasoning more effective and accessible. By demonstrating that a 4B parameter model can match much larger specialized reasoning models through intelligent test-time computation, RSA challenges assumptions about the necessary scale for advanced AI capabilities.

The dual innovation of:

1. The RSA inference algorithm (evolutionary population refinement)
2. Aggregation-aware RL training (preparing models for aggregation)

creates a powerful framework that outperforms existing scaling strategies across mathematics, coding, and knowledge domains.

Why This Matters

- **Efficiency**: Smaller models can achieve competitive performance
- **Accessibility**: Simpler implementation without complex infrastructure
- **Flexibility**: Adjustable compute budget for different use cases
- **Foundation**: Opens new research directions in test-time scaling

The Bigger Picture

RSA suggests we may be at an inflection point where **how we use models during inference** becomes as important as model architecture and training. This paradigm shift could reshape how we think about deploying AI systems, potentially making advanced reasoning capabilities more widely available.

Key Takeaways

1. ✅ **Test-time scaling can rival model scaling** for reasoning tasks
2. ✅ **Hybrid approaches outperform pure parallel or sequential methods**
3. ✅ **Training must account for deployment strategies** (aggregation-aware RL)
4. ✅ **Smaller models + smart inference > naive use of larger models**
5. ✅ **Evolutionary principles apply effectively to reasoning chains**
6. ✅ **Simple implementations can yield powerful results**

Resources

- **Project Website**: https://rsa-llm.github.io
- **Code**: Available on GitHub (check project website)
- **ArXiv Paper**: Coming soon (as of search date)
- **Models**: Tested with Qwen3-4B-Instruct-2507

This report synthesizes information from the RSA project website and related research. For complete technical details and mathematical formulations, please refer to the forthcoming academic paper.