IGNIS

ARCS: Adaptive Reasoning with Calibrated Self-Assessment

A Hybrid Neural-Symbolic Architecture for General Pattern Recognition

Research Paper - October 2025

Abstract

We present ARCS (Adaptive Reasoning with Calibrated Self-Assessment), a novel architecture built on a fundamental insight: **AI needs to understand what tools it has and when to use them**—the same way humans learn their own capabilities.

Unlike traditional approaches that scale neural networks until intelligence emerges, ARCS teaches AI self-awareness by giving it exact mathematical tools (primitives) alongside neural learning. The system learns **which tools to use when**, not how to approximate every operation through trial and error.

On ARC-AGI-2 official test, ARCS achieves approximately 25% near-perfect solutions (90-99% accuracy), demonstrating viability while honestly acknowledging current limitations. More importantly, it represents a different path forward: **teaching AI to understand its own container and capabilities** rather than forcing it to rediscover basic operations billions of times.

Core Philosophy: Efficient intelligence requires both established operational foundations and adaptive learning—not forcing AI to rediscover fundamental operations that should be built-in capabilities.

1. Introduction

Current AI can't tell the difference between what it knows and what it's guessing. This isn't a training problem—it's an architectural one.

When trained end-to-end, neural networks learn approximate versions of everything: “rotate 90 degrees,” “extract column 3,” even basic mathematical operations. This creates cascading failures. The AI gets a false positive (thinks it's right when it's wrong), treats that guess as truth, and builds entire reasoning chains on lies. Result: hallucinations, confident incorrectness, and brittleness under variation.

The root issue: No intelligence is born with objective knowledge. We all start with subjective exploration—touching, seeing, experiencing the world. That's the primordial soup of consciousness. But without testing experiences against reality, you never gain objective knowledge. You might think you understand something (subjective belief) while being completely wrong (objective reality).

ARCS solves this through forced self-awareness:

- **Objective anchors** (exact primitives): Mathematical operations that never change, providing stable reference points
- **Subjective exploration** (neural learning): Pattern recognition in unknown territory
- **Reality testing** (grading): Every hypothesis verified against ground truth
- **Confusion matrix tracking**: System learns to distinguish true knowledge from false confidence

Think of a surgeon: they don't reinvent how scalpels work before each operation. The tools are established—skill comes from knowing **when and how to apply each tool** for different situations. ARCS separates established technique (exact primitives) from adaptive judgment (neural pattern recognition), then teaches the AI to be objective about which is which.

---

2. Architecture

ARCS combines learned and exact components in a clean functional interface where everything speaks the same language: tensors.

Core Components

Neural Consciousness (compact architecture - learns representations):

- Three-aspect parallel processing: fast intuition, synthesis, deliberate reasoning
- Learns *patterns* and *which tools to use when*, not the tools themselves
- Compact architecture designed for efficiency, not brute-force scale
- Position-aware to maintain pattern stability across transformations

Pure Functional Primitives (curated library - never learned):

- Spatial, arithmetic, logical, and pattern-based operations
- Mathematically exact: operations produce deterministic results
- Never modified during training—they are objective anchors
- Cover fundamental transformations needed for pattern reasoning

Discovery System (composes primitives):

- Systematically explores combinations of primitive operations
- Evaluates using objective grading (S/A/B/C/D/F scale where only 100% = true success)
- Search strategy balances breadth and depth for efficient exploration
- Verifies solutions across multiple examples before accepting

Meta-Learner (tracks what works):

- Records which operation types achieve which performance levels across tasks
- Biases future searches toward historically successful strategies
- Learns tool selection patterns without learning the tools themselves
- Integrated with neural model for coherent decision-making

Dimensional Awareness

ARCS adapts to data structure without hardcoded assumptions. Whether sequential (1D), spatial (2D), volumetric (3D), or higher-dimensional, the system observes properties and learns through experience what they mean. This makes the architecture **domain-agnostic**—same system handles text, images, audio, 3D data, or any structured information.

Performance Characteristics

- **Training**: Efficient on consumer hardware
- **Inference**: Real-time on typical tasks
- **Composition speed**: Rapid hypothesis testing enables practical search
- **Scalability**: Components improve independently without breaking each other

---

3. Results

Training Methodology

ARCS trains on ARC-AGI using grading-based learning:

- **S-rank (100%)**: Perfect understanding—the only true success
- **A-rank (90-99%)**: Close but not mastery
- **B-rank (80-89%)**: Good but gaps remain
- **C/D/F**: Progressively worse

Loss function penalizes anything below S-rank. **Key principle**: No task-specific code. Same system handles all patterns through discovered primitive compositions.

Current Performance (October 2025)

ARC-AGI-2 Official Test (117 tasks):

- **0 S-ranks** (100%)
- **~30 A-ranks** (90-99%) - approximately 25% of benchmark, within a few pixels of perfect
- **57% mean accuracy** across all tasks

ARC-AGI-2 Training/Validation (120 tasks):

- **Few S-ranks** (100%)
- **46 A-ranks** (90-99%)
- **74 tasks** above passing (60%+)

What This Means

The philosophy works: 25% of test tasks at 90-99% proves the approach is viable. The primitives ARE sufficient—you can't get that close without the right operations.

The gap is solvable: Difference between 90% and 100% isn't conceptual failure, it's missing specific compositions (stateful operations, conditional logic, deeper search strategies).

Honest assessment: Not AGI. Not state-of-the-art. But a fundamentally different path showing genuine promise with clear next steps.

---

4. Key Technical Innovations

No Transformers

ARCS removes traditional Transformer architecture entirely. Why? Transformers were designed for pattern matching, not for understanding the difference between knowledge and guessing.

The custom architecture enforces confusion matrix tracking at the core:

- **True Positives**: "I think I'm right" AND "I am right" (verified knowledge)
- **False Positives**: "I think I'm right" BUT "I'm wrong" (hallucination detected)
- **True Negatives**: "I think I'm wrong" AND "I am wrong" (honest uncertainty)
- **False Negatives**: "I think I'm wrong" BUT "I'm right" (unrecognized capability)

This prevents the cascade effect where false confidence compounds into reasoning chains built on lies.

Compositional Search with Grading as Gradient

Traditional AI uses backpropagation through neural weights. ARCS uses **grading as the learning signal**:

- Discovery system tests thousands of primitive compositions
- Each gets graded S/A/B/C/D/F based on actual correctness
- Meta-learner biases future searches based on what achieved high grades
- Neural model learns which patterns suggest which tool combinations

This creates a tight feedback loop: subjective pattern exploration + objective measurement = efficient convergence on solutions that actually work.

Clean Functional Separation

Everything is tensors. Primitives operate on tensors, neural networks produce tensors, discovery composes tensor operations. No translation layer, no special interfaces, no domain-specific glue code.

This makes the architecture truly general-purpose—same system handles any dimensional data without modification.

---

5. The Economics of Self-Aware AI

Traditional AI development is economically unsustainable at scale. Every capability improvement requires:

- Retraining billion-parameter models from scratch
- Months of compute time on specialized hardware
- Risk of catastrophic forgetting (new training degrades old capabilities)
- No guarantee the improvement will work

ARCS changes the economics:

Incremental capability expansion: Add new primitives to the library without retraining neural components. Like upgrading a toolbox—new tools don't require relearning how to use the old ones.

Parallel development: Neural representations and functional primitives can be improved simultaneously by different teams. The clean interface between them prevents conflicts.

Explainable failures: When the system fails, you can see exactly which operation combination was attempted and where it failed. This makes debugging tractable instead of searching through billions of opaque parameters.

Resource efficiency: Consumer hardware suffices for both training and inference. No need for specialized AI infrastructure or massive compute clusters.

This modularity isn't just intellectually satisfying—it's the difference between sustainable AI development and an endless arms race of bigger models.

---

6. Future Directions

Next Steps

Analysis of near-perfect tasks reveals several areas for improvement:

- **Stateful operations**: Ability to reuse intermediate results in complex compositions
- **Conditional logic**: Branching based on data properties
- **Deeper search**: More sophisticated composition strategies
- **Enhanced primitives**: Additional position-relative and adaptive operations

These improvements address the current gap between 90-99% performance and perfect solutions.

Broader Applications

While developed for ARC, the architecture is general-purpose:

Potential domains:

- Program synthesis (compose code primitives)
- Mathematical reasoning (compose algebraic operations)
- Data transformation (ETL pipeline generation)
- Scientific discovery (compose experimental procedures)
- Robotics (compose motor primitives)

Any domain where:

1. Ground truth operations exist
2. Composition creates complexity
3. Exact correctness matters

Why This Approach Scales

The architecture isn't optimized for ARC specifically—it's designed around universal principles that apply to any reasoning domain.

Scalability comes from separation of concerns:

- Neural components can grow as needed for complex pattern recognition
- Primitive library can expand with new operational categories
- Self-awareness mechanisms remain constant (confusion matrix doesn't need to scale)
- Each component improves independently without breaking the others

The economic advantage: Traditional AI requires retraining entire billion-parameter models when adding capabilities. ARCS can add new primitives (exact operations) or refine neural representations (learned patterns) independently. This modularity makes continuous improvement practical and cost-effective.

Long-term vision: As the primitive library grows to cover more domains (language, planning, creativity, social reasoning), the same self-awareness foundation ensures the AI never loses track of what it knows vs. what it's exploring. The confusion matrix prevents confidence drift regardless of scale.

7. Comparison with Related Work

vs. End-to-End Neural: Pure neural approaches learn approximate versions of everything, encoding “rotate 90 degrees” across millions of weights. Result: opaque, approximate, computationally expensive. ARCS separates what should be learned (pattern recognition, tool selection) from what should be exact (mathematical operations).

vs. Program Synthesis: Traditional DSL search either uses learned operations (imprecise) or exhaustive search (intractable). ARCS uses exact primitives with guided search (meta-learning), enabling efficient exploration while remaining domain-agnostic.

vs. Hybrid Neuro-Symbolic: Most hybrid systems struggle with the interface between continuous neural representations and discrete symbolic operations, requiring hand-crafted translation layers. ARCS solves this elegantly: everything is tensors. Primitives operate on tensors, neural networks produce tensors, discovery composes tensor operations. No translation needed.

8. Current Status

What Works

- **Self-awareness foundation**: Confusion matrix tracking successfully separates knowledge from guessing
- **Modular architecture**: Neural and functional components operate independently
- **Compositional reasoning**: System discovers novel solutions through primitive composition
- **Domain flexibility**: Same architecture handles different data types
- **Objective grading**: S-rank requirement enforces actual correctness
- **Resource efficiency**: Runs on consumer hardware

Active Development Areas

- **Generalization**: Validation performance transfer to test set
- **Final gap closure**: Bridging 90-99% to 100% on near-perfect tasks
- **Composition depth**: Advanced operation combinations
- **Training efficiency**: Faster convergence to high performance
- **Benchmark position**: Currently competitive, targeting state-of-the-art

Research Questions

1. **Can objective self-awareness scale to AGI?**
   The philosophy works on ARC. Does it generalize to all reasoning tasks?

2. **What objective anchors are universal?**
   Current primitive library works for pattern tasks. What’s needed for language, planning, creativity?

3. **How deep can subjective exploration go while maintaining objectivity?**
   Balancing exploration with grounding is key. Where’s the limit?

4. **Can this prevent hallucinations at scale?**
   Works on structured tasks. Does it work on open-ended generation?

5. **What's the minimal objective foundation needed?**
   How few hard-coded truths do you need to anchor infinite subjective exploration?

---

9. Conclusion

ARCS demonstrates a different path: **teaching AI to understand its own tools** rather than approximating everything through scale.

Results:

- 25% of ARC tasks at 90-99% accuracy proves the approach is viable
- Self-aware AI that distinguishes knowledge from guessing works
- Different architecture, different economics, different scalability model

Why this matters:

Current AI relearns basic operations through approximation every time. ARCS provides the established toolkit upfront (exact primitives), then trains judgment and pattern recognition (neural learning). The result:

- **Reliability**: Critical operations are exact, not approximate
- **Efficiency**: No wasted capacity relearning fundamentals
- **Interpretability**: Full audit trail of operation combinations
- **Modularity**: Add capabilities without retraining everything

Path forward:

Self-aware AI—AI that understands its own capabilities—is both achievable and necessary for efficient general intelligence. The 25% at 90-99% validates the philosophy. The work to reach 100% is engineering, not rethinking fundamentals.

---

10. Acknowledgments

This research builds on:

- The ARC-AGI benchmark and community
- Functional programming principles (Haskell, APL lineage)
- Neuro-symbolic AI research
- Years of empirical testing and iteration

Special recognition to the philosophy that **honesty about limitations is more valuable than hype about capabilities**.

End of Paper

Contact & Collaboration

For research collaboration, investment opportunities, or questions about the architecture, please contact us directly.

Note: ARCS is proprietary research. Implementation details and code are not publicly available.

Citation:

```
@article{arcs2025,
  title={ARCS: Adaptive Reasoning with Calibrated Self-Assessment},
  subtitle={A Hybrid Neural-Symbolic Architecture for General Pattern Recognition},
  year={2025},
  month={October},
  note={Proprietary research in progress}
}
```

This paper represents ongoing proprietary research. Results and methods are current as of October 2025 and subject to improvement through continued development.