Building an ARC Machine with NGF

What System Are We Building?

The target system is a **full ARC Machine**, capable of handling real Abstraction and Reasoning Corpus (ARC) challenges end-to-end.

Unlike the micro-LLM classifiers we've been running (which only perform the **label → classification** step),

a true ARC machine must execute the entire reasoning loop.

Required Pipeline Stages

1. **Label → Execution**

- Move from abstract "primitive class" (e.g., flip_h, rotate, ring_SW) into concrete **execution traces**.

- Requires implementing the geodesic parser (Stage-10) and the denoiser pipeline (Stage-11).

2. **Verification (Verifier)**

- Compare the candidate execution trace against the ground-truth mapping (ARC task input → output).

- Uses a synthetic verifier for generated data or an ARC-specific verifier for real tasks.

- This step is **missing** in our current micro-LLM and benchmark sims.

3. **Prediction / Abstain**

- After verification, output a final prediction **or abstain** if uncertainty exceeds a threshold.

- Stage-11 provides the phantom suppression, but abstention thresholds need to be engineered.

Current Status (Micro-LLM + Benchmarks)

- **Micro-LLM**: We only did the **label classification** step (got 100% on synthetic latent-ARC).

- **Stage11_benchmark_latest.py**: Provides a sandbox with synthetic traces and wells, but also only evaluates classification, not full execution/verification.

- **Patents & Article**: Clearly specify that deterministic reasoning requires **Warp → Detect → Denoise**, but also hinge on **Verifier availability**.

Implications

- What we have today is **not shoppable**: just a nuts-and-bolts classifier.
- A true ARC Machine requires:
- Geodesic parser integration (Stage-10)
- Stage-11 denoiser + phantom suppression
- Task-specific Verifier for ARC tasks
- End-to-end pipeline that executes transformations, checks results, and abstains when uncertain.

Workload Estimate

- Building this system is non-trivial: ~6–12 months of research engineering.
- Each stage (label $\rightarrow$ execution, verification, abstain logic) requires design + testing.
- Scaling to other domains (math, coding, commonsense) multiplies the engineering burden.

Conclusion

The “ARC Machine” is the **proof-of-concept (PoC)** that would demonstrate deterministic AI reasoning.

It is **much larger than the micro-LLM classifier**, requiring a complete execution + verification pipeline.

Once built, it would be a system capable of running real ARC tasks deterministically, without hallucinations.