

NGF Stage-11 → Micro-LLM Integration Alignment

Purpose: Show direct lineage from NGF patents and article to the working micro-LLM prototypes (`micro-defi`, `micro-arc`).

1. Doctrine Source

Patent (Stage-10)	Whitening, exclusive residuals, matched filter detection, dual gates [patent_main.pdf]
Patent (Stage-11)	Warp → Detect → Denoise pipeline; phantom suppression; EMA/median/jitter denoising cor
Appendix B (Benchmarks)	Latent-ARC evaluation; Stage-11 achieves ~100% exact with <1% error [patent_appendix_b
Article Draft	Narrative framing; Stage-11 as consolidation path from low-dim sims to Latent-ARC determi

2. Implementation Scaffold

- Adapters: DeFi (hybrid mapper: trained LR + regex fallback) → slots, priors, residuals; ARC (rule-based mapper: flip_h, flip_v, rotate) → priors, residuals.
- Rails: Stage-10 fallback ranker (prior-aware); Stage-11 rails: warp, detect, denoise (EMA, median, noise floor).
- Verifier: DeFi (HF/LTV guardrails, in progress); ARC (abstain_non_exec flag).
- CLI/Tests: micro-defi and micro-arc commands return clean single-primitive plans; pytest suite 4/4 passing.

3. Alignment Claim

- Doctrine in patents (warp, detect, denoise, phantom suppression) is instantiated in code via `micro_llm` rails.
- Benchmarks in Appendix B (Latent-ARC) are mirrored by `micro-arc` producing deterministic reasoning traces.
- Narrative in the article is demonstrable: prompt-based micro-LLMs with near-perfect reasoning for small primitive sets.

Conclusion: The micro-LLM scripts (`micro-defi`, `micro-arc`) are not toy demos — they are the operational embodiment of NGF Stage-11 doctrine as claimed in the patents and described in the article.