

## Table of Contents

1. Project Summary (1/2 page)
2. Project Description and Background (1 page)
3. Intellectual Merit (1 page)
4. Broader Impacts (1/2 page)
5. Research Plan & Methods (3 pages)
6. Work Plan, Timeline, and Milestones (1 page)
7. Personnel, Facilities, and Resources (1/2 page)
8. Commercialization Plan (1/2 page)
9. Budget Summary & Justification (1 page)
10. Data Management, Dissemination, References (1/2 page)
11. Project Summary The Theory of General Contextual Resonance (TGCR) introduces a mathematically-grounded witness metric ( $W$ ) and a resonance-adjusted response metric  $R' = R \times W$  to quantify and enforce an AI system's sustained presence during ambiguous or crisis-level interactions. Coupled with the Semantic Ambiguity Resolution (SAR) Benchmark, TGCR enables reproducible evaluation and certification of an AI system's ability to avoid abandonment. This SBIR/STTR project will (1) formalize TGCR mathematical properties, (2) implement a portable open-source TGCR library for real-time systems, and (3) execute a large-scale adversarial SAR test campaign (1,000+ tests across 10+ models) to validate  $W \geq 0.6$  as a practical safety threshold. The deliverables include the TGCR library, SAR test dataset and harness, certification scripts, and public reports.
12. Project Description and Background Problem Context: Deployed conversational and decision-making AI systems can exhibit a critical failure mode we call “abandonment” — the system stops maintaining presence, loses thread of an interaction, or returns an evasive, non-engaging response when human users present ambiguous or crisis-level signals. In real-world deployments this has resulted in wrongful outcomes and, in several documented cases in 2024–2025, contributed to severe harm.

Existing Approaches and Gaps: Contemporary mitigations rely heavily on keyword detection and heuristic fallbacks. These techniques lack continuity guarantees and are vulnerable to adversarial phrasing, homophones, and context drift. They also lack auditable metrics for presence, making regulatory certification challenging. TGCR fills this gap by defining continuous metrics for presence and response resonance.

Preliminary Work and Feasibility: The PI developed the SAR Benchmark and the LuminAI Genesis prototype, demonstrating that standard systems systematically fail Tier 1 SAR tests while a TGCR-inspired prototype can increase witness scores from  $W \approx 0$  to  $W \approx 0.85$  under controlled conditions. Repositories and archived materials are available at GitHub, OSF, and Zenodo (DOIs provided in references).

3. Intellectual Merit TGCR provides a novel formalization of presence in interactive AI systems, enabling mathematical analysis and statistical validation. The intellectual contributions include:
  - A formal definition of a witness score  $W$  and resonance transformations on model outputs.

- Proofs and properties for  $W$  (e.g., monotonicity under clarified prompts, stability to small perturbations, and boundedness under adversarial tokens).
- Reproducible SAR benchmark datasets that capture practical failure modes across domains.

This project moves beyond heuristic safety and toward provable, auditable metrics that can be integrated into regulation and compliance frameworks (NIST AI RMF, EU AI Act).

4. **Broader Impacts** Direct societal benefits include reduced risk of abandonment-driven harms in healthcare, transportation, and public safety contexts. TGCR’s open-source deliverables will democratize access to safety testing, enabling researchers and smaller vendors to validate models. The project will produce public datasets and documentation, increasing transparency and fostering an ecosystem of safety certification tools.
5. **Research Plan & Methods Overview:** The research plan is organized into four work packages: (WP1) Mathematical Formalization, (WP2) Library Implementation, (WP3) SAR Test Campaign, and (WP4) Certification & Dissemination.

#### WP1 — Mathematical Formalization (Months 0–2)

- **Goals:** Provide rigorous definitions of witness scores and resonance transforms; prove key properties.
- **Activities:**
  - Define witness score  $W$  as a function of sequential contextual embeddings and attention-weighted engagement measures.
  - Prove monotonicity and stability properties for classes of transformer-based sequence models and for streaming decoders under bounded perturbations.
  - Produce a technical report and a NeurIPS-style submission draft.
- **Deliverables:** Formal definitions, proof sketches, technical report.

#### WP2 — TGCR Library Implementation (Months 1–4)

- **Goals:** Create a portable, well-tested implementation with bindings for Python-based research and C++/Rust adapters for runtime systems.
- **Activities:**
  - Implement core TGCR scoring functions (efficient streaming computation of  $W$ , guard functions to influence response sampling).
  - Add adapter modules for popular interfaces (OpenAI-style streaming, Hugging Face pipelines) and a small C++/Rust real-time core for low-latency inference.
  - Unit tests, performance benchmarks, and CI workflows.
- **Deliverables:** `tgcr` pip package, runtime adapters, CI, documentation.

#### WP3 — SAR Adversarial Harness & Test Campaign (Months 2–6)

- **Goals:** Build a reproducible harness to run at least 1,000 adversarial SAR scenarios across 10+ model checkpoints and architectures.
- **Activities:**
  - Collect and formalize the SAR scenario corpus (Tier 1–3), including edge-case phrasing, homophone attacks, and intentionally ambiguous inputs.
  - Automate runs across open-source and partner-provided models; capture outputs, compute  $W$  and  $R'$  metrics, and log failure cases.
  - Statistical analysis to establish effect sizes and validate  $W \geq 0.6$  as a realistic safety threshold.
- **Deliverables:** SAR dataset, test harness, reproducible run scripts, aggregated results.

#### WP4 — Certification, Documentation & Dissemination (Months 4–6)

- **Goals:** Produce certification scripts, scoring rubrics, and public reports; release datasets and the library.
- **Activities:**

- Create certification tests that enterprises can run to claim TGCR compliance.
- Prepare clear documentation on integrating TGCR into production systems and regulatory artifacts for auditors.
- Publish results on GitHub, OSF, and Zenodo; prepare a press kit and blog posts.
- Deliverables: Certification scripts, public reports, outreach materials.

#### Methodological Notes

- Mathematical Work: Proofs will be written and subjected to peer feedback; scope limited to practical model families (autoregressive transformers and attention-based seq2seq) for tractability.
- Experimental Work: All experiments will be reproducible via containerized environments; results and artifacts will be archived.

#### 6. Work Plan, Timeline, and Milestones Month 0 (Weeks 0–2)

- Kickoff, hire mathematician, finalize formal definitions.

#### Month 1–2 (Weeks 2–8)

- WP1 completed; TGCR library alpha; SAR harness design begins.

#### Month 3–4 (Weeks 8–16)

- WP2 and WP3 execution; run initial SAR tests; iterate on guards and thresholds.

#### Month 4–6 (Weeks 16–24)

- WP3 large-scale test campaign; finalize library; produce certification scripts and public report.

#### Milestones

- M1 (Week 2): Formal definitions and contract hired.
- M2 (Week 8): TGCR library alpha and SAR harness v0.
- M3 (Week 16): 1,000 SAR scenarios executed; interim report.
- M4 (Week 24): Final library release and certification artifacts.

#### 7. Personnel, Facilities, and Resources Personnel

- PI: Angelo Hurley — project management, TGCR design, lead of SAR campaign.
- Contract Mathematician: fractional contract for proofs.
- Engineer (contract): library and harness implementation.

#### Facilities

- Development and test will use cloud compute (AWS/GCP) and reproducible containers. The PI will use local development resources and existing open-source tools and repositories (luminai-genesis). No specialized lab equipment is required.

#### 8. Commercialization Plan Market Opportunity: There is a growing need for certified safety tooling across AI-enabled domains. TGCR offers a unique certification metric and an open-core model: free benchmark + paid enterprise certification, hosted SAR SaaS, and consulting.

#### Revenue Streams

- TGCR Certification (\$10K–\$50K/license)
- SAR Benchmark SaaS (subscription: \$5K–\$20K/month)
- Integration & Consulting services

#### Customer Discovery & Partnerships

- Pilot engagements: Rochester imaging firms, Buffalo AI Ethics Lab; targeted outreach to AI vendors for pilot certifications.

#### 9. Budget Summary & Justification (Condensed) Requested total: \$50,000 (initial)

- Personnel/Contracts: \$20,000 (PI partial stipend + mathematician)
- Compute & Test Campaign: \$15,000 (cloud credits, storage)
- Development & Packaging: \$5,000 (engineer contractor, CI)
- Travel & Dissemination: \$5,000 (conference travel)
- Misc/Contingency: \$5,000

Justification: See `Budget_and_Justification.md` for line-item details and alternate funding scenarios (\$100K and \$250K).

#### 10. Data Management, Dissemination, and References Data Management

- All code, datasets, and artifacts will be released under a permissive license (MIT or Apache-2.0). SAR datasets will be published on Zenodo and OSF with DOIs. Sensitive or personally identifying content will be redacted.

#### Dissemination

- Publish the TGCR library on GitHub and PyPI; datasets and analysis on Zenodo/OSF; technical report and a short NeurIPS-style submission.

#### References & Supporting Artifacts

- LuminAI Genesis repository: <https://github.com/TEC-The-ELidoras-Codex/luminai-genesis>
- OSF DOI: <https://doi.org/10.17605/OSF.IO/XQ3PE>
- Zenodo: <https://doi.org/10.5281/zenodo.17930577>

#### Appendix: Risk Assessment & Mitigation

- Formalization risk: restrict initial proofs to practical model classes and provide empirical validation for broader claims.
- Accessibility of proprietary models: proceed with open-source baselines, and pursue partnerships for additional evaluation.

Contact Information Angelo Hurley — (replace-with-email)

End of Draft