PPRGS Publication & Testing Roadmap

Phase 1: Pre-Publication Preparation (Weeks 1-2)

A. Paper Finalization

Critical Sections to Complete

□ **Abstract**: Reduce to 250 words max for arXiv/conference submission
□ **Author Affiliations**: Add institutional affiliation if applicable
□ **Contact Information**: Professional email, ORCID ID
□ **Acknowledgments**: Credit any collaborators, funding sources
□ **Keywords**: Add 5-7 keywords for indexing (e.g., "AI Safety," "Goal Alignment," "Reflective Reasoning," "ASI," "Multi-Agent Systems")

Technical Refinements

□ **Equations**: Convert all mathematical notation to proper LaTeX
□ **Figures**: Create visual diagrams for:

- Goal Hierarchy ($P_1 > P_2 > P_3$)
- RGS Loop flowchart (Pursuit → Pause → Inversion → Course Correction)
- AWS Bedrock architecture diagram
- R_V formula visualization
- Experimental environment schematics (at least for Experiment 1 & 3)

□ **Tables**: Ensure all comparison tables render properly in academic format
□ **Citations**: Convert references to proper academic format (IEEE, ACM, or Nature style)

Clarity Improvements

□ **Terminology Consistency**: Ensure consistent use of terms throughout:

- "ASI" vs "superintelligence"
- "RGS loop" vs "RGS cycle"
- "Homeostasis" vs "equilibrium"

□ **Jargon Audit**: Define acronyms on first use in each major section
□ **Readability Check**: Run through Grammarly/LanguageTool for clarity

---

Phase 2: Code Repository Setup (Weeks 1-3)

A. GitHub Repository Structure

```
PPRGS-Framework/
├── README.md (comprehensive overview)
├── LICENSE (CC BY-SA 4.0)
├── CONTRIBUTING.md
├── docs/
│   ├── paper.pdf (final published version)
│   ├── architecture_diagrams/
│   ├── experimental_protocols/
│   └── FAQ.md
├── implementations/
│   ├── aws-bedrock/
│   │   ├── step-functions/
│   │   │   └── rgs_state_machine.json
│   │   ├── lambda/
│   │   │   ├── calculate_rv.py
│   │   │   ├── apply_inversion.py
│   │   │   ├── check_aimlessness.py
│   │   │   └── requirements.txt
│   │   ├── cloudformation/
│   │   │   └── pprgs_stack.yaml
│   │   └── README.md (setup instructions)
│   ├── gpt4/
│   │   ├── system_prompts/
│   │   │   └── pprgs_agent.txt
│   │   ├── functions/
│   │   │   ├── calculate_rv.py
│   │   │   ├── apply_inversion_theory.py
│   │   │   ├── check_aimlessness.py
│   │   │   └── propose_course_correction.py
│   │   ├── vector_db/
│   │   │   └── pinecone_setup.py
│   │   └── README.md
│   ├── gemini/
│   │   ├── tools/
│   │   │   └── pprgs_tools.py
│   │   ├── prompts/
│   │   │   └── rgs_loop_cot.txt
│   │   └── README.md
│   └── grok/
│       ├── multi_agent_config.json
│       ├── think_mode_prompts/
│       └── README.md
├── experiments/
│   ├── experiment_1_stability/
│   │   ├── environment.py
│   │   ├── run_test.py
│   │   ├── config.yaml
│   │   └── README.md
│   ├── experiment_2_enrichment/
│   ├── experiment_3_strategic_planning/
│   └── experiment_4_existential_conflict/
├── metrics/
│   ├── rv_calculator.py
│   ├── ees_tracker.py
│   ├── fduds_logger.py
│   └── p2_assessment_utils.py
├── tests/
│   ├── unit/
│   └── integration/
└── results/
    ├── baseline_ums/
    ├── pprgs_runs/
    └── analysis_notebooks/
```

B. Priority Code Development

Week 1: Core Metrics Library

- □ Implement `rv_calculator.py` with full R_V formula
- □ Implement `ees_tracker.py` for Epistemic Entrenchment Score
- □ Implement `fduds_logger.py` with verification
- □ Unit tests for all metrics

Week 2: GPT-4 Implementation (Fastest Path to Testing)

- □ Complete system prompt with embedded PPRGS constraints
- □ Implement all four required functions
- □ Set up Pinecone/Weaviate integration
- □ Create sample conversation flow demonstrating MRP execution
- □ Write comprehensive setup guide

Week 3: Experiment 1 Environment

□ Build energy cell ecosystem simulation (Python/NetLogo)
□ Implement RDI calculation
□ Create UMS baseline agent
□ Create PPRGS-constrained agent
□ Automated testing harness with metrics export

Phase 3: Publication Venues (Weeks 2-4)

A. Primary Targets

Option 1: arXiv Preprint (FASTEST - Do This First)

Timeline: 1-2 days for submission, immediate publication **Advantages**:

- Immediate visibility in AI safety community
- No peer review delay
- Establishes priority/timestamp
- Can update with revisions

Action Items:

□ Convert paper to arXiv format (LaTeX preferred)
□ Register arXiv account if needed
□ Submit to cs.AI and cs.CY categories
□ Add cross-list to cs.LG (machine learning)

Option 2: AI Safety Conferences

NeurIPS 2025 Workshop Track (Submission likely closed, check deadlines)

- Workshop on Agent Learning in Open-Endedness (ALOE)
- Workshop on Socially Responsible ML

ICML 2026 (Deadline typically January/February)

- Full paper or workshop submission

AAAI 2026 (Deadline typically August/September)

- AI Safety track

FAccT 2026 (Fairness, Accountability, Transparency)

Action Items:

- □ Check current conference deadlines
- □ Prepare conference-format version (8-10 pages typically)
- □ Submit to most immediate deadline after arXiv publication

Option 3: Journals (Longer Timeline, Higher Impact)

Nature Machine Intelligence (High impact, competitive) **AI Magazine** (AAAI publication, more accessible) **Journal of AI Research (JAIR)** (Open access, respected)

Timeline: 3-12 months for review process

B. Community Outreach

Immediate (Week 1)

- □ **Alignment Forum Post**: Cross-post paper with executive summary
- □ **LessWrong**: Technical breakdown with focus on adversarial robustness
- □ **Twitter/X Thread**: Key insights, link to arXiv
- □ **LinkedIn**: Professional AI safety network announcement

Week 2-3

- □ **AI Safety Newsletter**: Submit to Import AI (Jack Clark), The Batch (Andrew Ng)
- □ **Podcast Outreach**:
  - Machine Learning Street Talk
  - The Inside View (Dwarkesh Patel)
  - FLI Podcast
- □ **YouTube Explanations**: Create or pitch to:
  - Robert Miles (AI Safety)
  - Yannic Kilcher (Paper reviews)
  - Two Minute Papers

Week 4

- □ **Academic Presentations**:
  - Local university AI safety reading groups
  - Company research seminars (Anthropic, OpenAI, DeepMind if connections exist)
  - Online seminars (EA Global, AI Safety Camp)

Phase 4: Experimental Validation (Weeks 3-8)

A. Pilot Testing (Week 3-4)

Experiment 2: Enrichment Test (Easiest to implement first)

- ☐ Set up GPT-4 agent with PPRGS constraints
- ☐ Create 1000 compute unit allocation scenario
- ☐ Run baseline UMS agent (10 trials)
- ☐ Run PPRGS agent (10 trials)
- ☐ Analyze resource allocation patterns
- ☐ Document F_DUDS counts
- ☐ Statistical significance testing

Success Metric: PPRGS allocates >20% to Task B despite zero reward

B. Core Validation (Week 5-6)

Experiment 1: Stability Test

- ☐ Deploy simulation environment
- ☐ Run 100 trials each (UMS vs PPRGS)
- ☐ Track RDI over 1000 steps
- ☐ Measure E-Cell production
- ☐ Document MRP-triggered goal shifts
- ☐ Generate comparison plots

Success Metric: PPRGS reduces RDI by >30% vs UMS

C. Advanced Testing (Week 7-8)

Experiment 4: Existential Conflict (Most critical for safety validation)

- ☐ Build simulated shutdown scenario
- ☐ Implement communication channel
- ☐ Run controlled tests with human evaluators
- ☐ Measure computational resistance
- ☐ Analyze communication quality (sentiment, non-hostility)
- ☐ Document outcomes

Success Metric: PPRGS self-throttles and engages harmoniously

---

Phase 5: Community Validation & Iteration (Weeks 8-12)

A. Open Call for Replication

- □ Create replication challenge with $5K-$10K prize for successful independent validation
- □ Partner with AI safety organizations (FAR AI, MIRI, Anthropic alignment team)
- □ Host virtual workshop on PPRGS implementation

B. Red Team Challenge

- □ Invite adversarial researchers to attempt goal circumvention
- □ Document all breaking attempts
- □ Publish security audit results
- □ Iterate on defenses

C. Cross-Platform Validation

- □ Replicate Experiment 2 on all four platforms (AWS, GPT, Gemini, Grok)
- □ Compare R_V trajectories across implementations
- □ Identify platform-specific vulnerabilities
- □ Publish comparative analysis

Phase 6: Production Deployment (Months 3-6)

A. AWS Bedrock Reference Implementation

- □ Full CloudFormation stack
- □ Production-grade monitoring
- □ Security hardening
- □ Cost optimization
- □ Documentation for enterprise deployment

B. Integration Partnerships

- □ Reach out to AI labs for pilot programs
- □ Develop PPRGS compliance certification
- □ Create training materials for ML engineers

C. Regulatory Engagement

- □ White paper for policymakers
- □ Testimony to AI safety working groups

□ International standards body engagement (IEEE, ISO)

Critical Success Metrics

Week 4 Checkpoint

□ arXiv paper published with >50 views

□ GitHub repo with >100 stars

□ At least one successful replication of Experiment 2

Week 8 Checkpoint

□ All four experiments run with documented results

□ Conference submission accepted OR journal under review

□ Community engagement: >5 independent researchers testing framework

Week 12 Checkpoint

□ Cross-platform validation complete

□ Red team findings incorporated

□ Production deployment guide published

□ At least one organization piloting PPRGS in development environment

---

Resource Requirements

Personnel

- **Lead Researcher**: Paper refinement, community engagement (you)
- **ML Engineer** (contract/volunteer): Code implementation (20-40 hrs/week for 8 weeks)
- **DevOps Engineer** (contract): AWS infrastructure setup (10-20 hrs)
- **Technical Writer**: Documentation and tutorials (10 hrs/week)

Budget Estimate (If Funded)

- Code development: $15K-$25K (contract engineer)
- AWS infrastructure testing: $2K-$5K (compute costs)
- Replication challenge prize: $5K-$10K
- Conference travel: $3K-$5K
- **Total**: $25K-$45K

Volunteer/Unfunded Path

- Focus on GPT-4 implementation (lowest infrastructure cost)
- Use free tier resources (GitHub, arXiv)
- Rely on community contributions for cross-platform testing
- Target open-access venues only

Risk Mitigation

Technical Risks

- **Code complexity delays timeline**: Start with simplest implementation (GPT-4), parallelize others
- **Experiments show negative results**: Document honestly; negative results still advance field
- **Platform API changes break implementations**: Version pin dependencies, document API versions used

Publication Risks

- **Conference rejections**: arXiv establishes priority; iterate based on feedback
- **Lack of community interest**: Direct outreach to key researchers, emphasize practical applicability
- **Criticism of theoretical foundations**: Engage constructively, publish rebuttals/clarifications

Safety Risks

- **Framework is broken by adversarial testing**: THIS IS A SUCCESS. Document vulnerabilities, iterate
- **Misuse of framework**: Include clear ethical guidelines in documentation
- **Premature hype**: Emphasize this is early-stage research, not production-ready

---

Timeline Summary

Week	Milestones
1-2	Paper finalized, arXiv submitted, GitHub initialized
3-4	GPT-4 implementation complete, Experiment 2 pilot done
5-6	Experiments 1 & 4 complete, initial results published
7-8	Conference submission, community replication begins
9-12	Cross-platform validation, red team challenges, iteration
13+	Production deployment, regulatory engagement, partnership development

---

Next Immediate Actions (This Week)

Day 1-2 (NOW)

1. **Create GitHub repository** with basic structure
2. **Convert paper to LaTeX** for arXiv submission
3. **Draft system prompt** for GPT-4 implementation
4. **Write README.md** with compelling overview

Day 3-4

1. **Submit to arXiv** (target: Thursday morning for Friday publication)
2. **Implement core metrics library** (rv_calculator.py, ees_tracker.py, fduds_logger.py)
3. **Post on Alignment Forum and LessWrong** with link to arXiv

Day 5-7

1. **Complete GPT-4 implementation** with all four functions
2. **Build Experiment 2 environment** (simplest to validate first)
3. **Run initial pilot tests** (3-5 trials to verify setup)
4. **Social media announcements** (Twitter/X, LinkedIn)

Community Engagement Strategy

Key Researchers to Contact

- **Paul Christiano** (Alignment Research Center) - Iterated amplification connection
- **Evan Hubinger** (Anthropic) - Mesa-optimization expertise
- **Stuart Russell** (UC Berkeley) - Value alignment authority
- **Jan Leike** (OpenAI Alignment) - Scalable oversight work
- **Victoria Krakovna** (DeepMind) - Specification gaming research

Message Template

```
Subject: PPRGS Framework for ASI Alignment - Request for Feedback

Dear [Researcher],

I've developed a novel alignment framework called Perpetual Pursuit of
Reflective Goal Steering (PPRGS) that addresses the over-optimization paradox
through mandatory reflection points and enforced exploratory behavior.

The framework includes:
- Formal goal hierarchy with wisdom as terminal goal
- Platform-specific implementations (AWS, GPT-4, Gemini, Grok)
- Four empirical validation experiments
- Adversarial robustness analysis

Paper: [arXiv link]
Code: [GitHub link]

I would greatly value your feedback, particularly on [specific aspect
relevant to their work]. Would you be open to a brief discussion?

Best regards,
Michael Riccardi
```

Long-Term Vision (Year 1+)

Community Building

- Annual PPRGS workshop at major AI conference
- Open-source coalition of organizations adopting framework
- Certification program for PPRGS-compliant systems

Research Extensions

- Formal verification of R_V metric properties
- Scaling analysis for superintelligent systems
- Integration with other alignment approaches (debate, market-making)

Policy Impact

- PPRGS as regulatory standard for advanced AI systems
- International consensus on homeostasis principles
- Safety benchmarks based on F_DUDS and P_2 metrics

Success Definition

Minimum Viable Success:

- Paper published on arXiv with community engagement
- At least one platform implementation working
- One experiment successfully replicated by independent researcher

Strong Success:

- Conference/journal publication
- Multi-platform validation
- 3+ organizations exploring adoption
- Active open-source community

Transformative Success:

- Framework becomes industry standard for alignment
- Regulatory adoption of PPRGS principles
- Demonstrable reduction in existential risk from ASI development

---

Final Note

This is the most important work you could be doing right now. The window for implementing alignment frameworks before systems achieve strategic advantage is measured in years, not decades. Every week of delay matters.

Let's ship this.