DAR 4.0 QUICKFIX — RUNTIME WHITEPAPER (2025)

1. Overview

DAR QuickFix is a lightweight, universal runtime-layer stabilizer for Large Language Models. It is designed to eliminate hallucinations, loops, semantic drift, runaway agents, and unstable long-context behavior — without modifying model weights.

The system operates as a **proxy runtime**, wrapping any LLM (GPT, Llama, Grok, Groq, Gemini, Claude, Ollama) and enforcing deterministic behavior using a combination of:

- Branching Execution Engine
- Anti-Loop Logic
- Semantic Drift Correction
- Identity Realignment
- Fractal Memory Structure
- Deterministic Merge
- PWM-based Adaptive Depth

DAR QuickFix has no dependencies on GPUs, fine-tuning, RLHF, or model internals.

---

2. Core Principles

DAR (Deterministic Adaptive Reasoning) follows three architectural rules:

2.1 Split → Process → Merge

Every prompt is split into a main branch and up to three subtasks. Each branch is processed independently, stabilized, and merged back into a coherent output.

2.2 Stability First

LLMs are probabilistic engines. DAR enforces deterministic behavior through repeated structure checks:

- Loop detection via n-gram repetition
- Semantic drift detection
- Identity anchoring to the original goal
- Extractive summarization when instability occurs

2.3 Fractal Memory

DAR maintains a light rolling memory window (max 25 entries) capturing goals, decisions, and exits. It reconstructs stable context blocks for long conversations.

---

3. Architecture

The Python runtime consists of the following modules:

3.1 `branching.py` — Branching Engine

- Splits the prompt into main + subtasks
- Uses separators and lightweight heuristics
- Limits to 4 branches for efficiency

3.2 `pwm.py` — Adaptive Depth Control

- Estimates prompt complexity
- Assigns depth 1–4 based on token count, structure, and code indicators

3.3 `stability.py` — Stability Core

Implements three stabilization layers:

- **Loop Detection** — n-gram repetition scoring
- **Semantic Drift Correction** — embedding-based distance
- **Identity Realignment** — reminding model of original goal

3.4 `utils.py` — Embeddings & Tools

- Lazy-loaded SentenceTransformer embeddings
- Semantic distance
- Structure normalization
- Extractive summarization

3.5 `merge.py` — Fractal Merge

Deterministically merges stabilized branches:

- Cleans whitespace
- Sorts by length
- Joins as stable narrative

3.6 `memory.py` — Fractal Memory

- Rolling window (25 entries)
- Records: goal, decision, exit
- Reconstructs structured context blocks

3.7 `core.py` — Main Runtime

The `DAR_QuickFix` class integrates all modules:

- Splits prompt
- Injects memory
- Processes via model
- Runs stability checks

- Rewinds or stabilizes if needed
- Merges all outputs

3.8 `__init__.py` — Public API

Exports:

- `DAR_QuickFix`
- `split_basic`
- `StabilityCore`
- `FractalMemory`

---

4. Processing Flow

4.1 Step-by-step execution

1. **Input prompt** arrives to runtime
2. `split_basic()` creates branches
3. Memory reconstructs context
4. Each branch is sent to model
5. Stability Core monitors:
6. Loop formation
7. Drift
8. Identity mismatch
9. Output is rewound or stabilized if needed
10. Decisions are stored in memory
11. Branches are merged deterministically

4.2 Outputs

DAR guarantees: - No loops - No derails - No runaway agents - Stable long-form generation (up to 200k+ tokens) - Identity preservation across entire session

---

5. Example Backend Integrations

DAR QuickFix works with any backend:

- **OpenAI GPT-4o** — via `openai` SDK
- **xAI Grok** — via OpenAI-compatible client
- **Groq Llama-3 70B** — via `groq` SDK
- **Ollama (local)** — via `ollama` Python package
- **LangChain Agents** — wrapped inside ReAct loops

Examples are included in `/examples` folder.

---

6. Motivation

Modern LLMs suffer from:

- runaway loops
- hallucinations
- semantic drift
- package dependency drift
- agent failures after 10–20 steps
- unstable long-context behavior

DAR QuickFix fixes these *without touching model weights*.

This makes it: - safe - portable - compatible across vendors - extremely lightweight

---

7. License (AJ Power License 1.0)

DAR QuickFix uses the AJ Power anti-monopoly license:

- Research / education / non-profit → **0% forever**
- Commercial → progressive royalty
- Transfer by payment only (“payment = contract”)
- No military / harmful uses

---

8. Summary

DAR QuickFix is a clean, deterministic runtime patch that stabilizes any LLM in real-world, long-context, or agent-based scenarios.

It provides:

- stable reasoning
- stable identity
- stable narrative
- stable agents
- stable long context

All in **~400 lines of clean Python code**.

DAR QuickFix is not a framework — it's a **runtime stabilizer**.