FLOWRRA — Version 2 Plan (Literature Outline)

Author: Sensei / Rohit (collab) **Version:** FLOWRRA v2 (roadmap) **Date:** 2025-09-05 (Asia/Kolkata)

---

0. Purpose

This document provides a literature-style outline for FLOWRRA Version 2. It summarizes what was achieved in Version 1, identifies limitations, and defines the steps and theoretical constructs that will guide the transition into Version 2. The focus is on integrating the new density estimator (speed-aware repulsion, comet-tails), loop-level collapse, and memory-based adaptation of repulsion scars.

---

1. Summary of Version 1

- **Density Function Estimator (v1):** Implemented a basic grid-based density using positive attraction (broad prior) and repulsion from sensed obstacles. Repulsion was node-local, point-based, and decayed over time.
- **Collapse Mechanism (v1):** Collapse occurred at the **node level**. If an individual node fell out of coherence, it was reset or snapped back to a safe state. This led to frequent collapses, instability, and incoherence of the overall loop.
- **Learning Dynamics (v1):** Repulsion was updated as local scars, but without speed-awareness or forward projection. As a result, moving obstacles were not properly anticipated.
- **Achievements:**
- Established core FLOWRRA cycle: sensing → density update → movement → collapse → learning.
- Built groundwork for retrocausal dynamics: collapse events retroactively shaped density.
- Validated that FLOWRRA could *exist* and maintain basic form across repeated perturbations.
- **Limitations:**
- Over-fragmentation from node-level collapse.
- Lack of velocity-awareness in repulsion.
- Collapse triggered almost every step, preventing meaningful long-term coherence.

---

2. Transition to Version 2

FLOWRRA v2 will address these limitations by introducing systemic improvements:

2.1 Density Function Estimator (speed-aware)

- Upgrade repulsion field to include **forward-projected kernels** (comet-tail effect). Each obstacle with velocity v_t generates projected repulsion along its future trajectory.
- Introduce **repulsion scars**: when collapse occurs, the path of failure leaves a persistent repulsion trace, preventing repetition of the same error.
- Maintain smoothness with diffusion + decay, ensuring the field stays differentiable.

2.2 Loop-Level Collapse

- Collapse becomes an **entire loop event**: triggered only when coherence of the *whole loop* drops below threshold.
- Coherence defined via density-consistency, geometric energy, or spectral stability.
- Collapse rule: rollback to last coherent loop state (or projection to coherent manifold).
- Retroactive update: collapsed loops produce repulsion scars along their failing trajectory.

2.3 Integration & Data Ingestion

- Shift from **node-by-node ingestion** to **loop-level state ingestion**: the system stores and evaluates full loops as its atomic unit.
- Historical buffer of coherent loop states introduced, enabling rollback and long-term stability.

2.4 Learning Feedback

- Collapse is reframed as a **learning signal**: every collapse generates structured repulsion updates, guiding future avoidance.
- Event severity s_t proportional to incoherence magnitude.

2.5 Additional Improvements

- Dynamic balance of attraction vs. repulsion using adaptive parameter $\beta(t)$.
- Emergency hard-mask fallback in case of repeated collapses in a small region.
- Potential integration of memory-graph representation of past collapses, enabling meta-learning across episodes.

---

3. Version 2 Goals

1. **Stability:** Reduce collapse frequency drastically by preserving loop integrity.
2. **Anticipation:** Enable FLOWRRA to avoid moving obstacles before direct collision.
3. **Memory:** Ensure collapses leave scars, shaping density toward sustainable behavior.
4. **Coherence:** Make loop structure the primary object of preservation, rather than node survival.
5. **Scalability:** Keep computational complexity manageable by using grid approximations and forward-limited projections.

---

4. Roadmap for Implementation

1. **Integrate RepulsionField v2:** speed-aware comet-tail kernels, scars on collapse, smoothing + decay.
2. **Integrate LoopManager v2:** loop-level coherence metric, rollback, and projection.
3. **Update training pipeline:** shift from node-wise updates to loop-wise updates.
4. **Introduce metrics:** collapse frequency, coherence average, repulsion field entropy.
5. **Benchmark:** compare Version 1 vs. Version 2 performance on stability and learning speed.

---

5. Outlook

With Version 2, FLOWRRA will progress from a system that *merely survives perturbations* to one that *learns sustainable, anticipatory behavior*. The shift from local to systemic coherence (node → loop) and from static to dynamic density (point repulsion → comet tails + scars) marks a major step toward FLOWRRA's ultimate vision of **retrocausal adaptive intelligence**.