

Reality Protocol (RP) — Practical Translation for ML Practitioners

Purpose

This document translates the Reality Protocol (RP) and the A■ invariant into plain ML and systems language.

No physics terminology is required. The goal is to make RP actionable for practitioners working with LLMs, decoders, inference controllers, and safety systems.

RP is not a training method.

RP is a control and stabilization regime for inference-time behavior.

Core Idea (One Sentence)

Only transitions that locally reduce instability and cost are allowed to persist; all others are blocked or terminated.

1. Mental Model (ML-native)

Think of an LLM as a system that:

- proposes many possible next continuations,
- assigns scores (logits / probabilities),
- and must choose one continuation at each step.

Default behavior:

- always continue generation,
- fill uncertainty with statistically plausible text,
- overgenerate when information is missing.

RP behavior:

- treat continuation as optional,
- require local justification to proceed,
- allow termination as a valid outcome.

2. A■ Invariant (ML Translation)

At every step:

1. Generate candidate continuations.
2. Filter out candidates that violate stability constraints.
3. Among remaining candidates, choose the one with minimal execution cost.
4. If no candidate passes the filter → terminate.

This is not planning, not global optimization, and not goal pursuit.

3. Key Quantities (Renamed for ML)

Instability (H): uncertainty, entropy, semantic drift
Risk (T): policy violation risk, safety uncertainty
Execution Cost (Z): token count, latency, compute
Admissible: allowed by safety + uncertainty bounds
Collapse: EOS / stop generation
Relaxation: choosing lowest-cost stable continuation

These quantities can be heuristic, implicit, or approximate.

4. Admissibility-First Logic

Most decoding strategies:
- score tokens first,
- then apply penalties.

RP reverses this order:

1. Check admissibility first.
2. Only then minimize cost.

This prevents optimize-through-uncertainty failures.

5. Prediction Is Optional

RP does not require:
- world models,
- belief states,
- long-horizon planning.

Prediction is treated as:

- expensive,
- invoked only if it reduces current uncertainty,
- otherwise skipped.

6. Silence / Termination Is a Feature

If the model cannot:

- answer safely,
- answer with sufficient grounding,
- or reduce uncertainty,

it should stop.

Termination is the stable fixed point.

7. Why RP Reduces Hallucinations

Hallucinations occur when:

- the model is forced to continue,
- uncertainty is high,
- stopping is not allowed.

RP fixes this by:

- making stopping cheap,
- making speculation expensive,
- blocking unsafe transitions early.

8. Where RP Fits in ML Systems

RP can be implemented as:

- system prompt,
- inference-time controller,
- decoding constraint layer,
- wrapper around existing models.

No retraining required.

9. What RP Is NOT

RP is not:

- reinforcement learning,
- reward optimization,
- agent simulation,
- persona modeling.

10. Practical Use Cases

Good fit:

- research assistants,
- code analysis,
- scientific reasoning,
- safety-critical QA.

Poor fit:

- chatty companions,
- entertainment,
- open-ended fiction.

11. Minimal Operational Summary

Filter → Minimize → Otherwise Stop

Final note:

RP does not try to make models smarter.

It makes them stop being wrong when they should not speak.