Residual Dynamics (RD)
A Finite-State Formal System that Treats Failure as a First-Class Object

Xie Kaifan

2026-02-15

1 Research Background and Problem Stance

In most existing formal systems, dynamical systems, and computational models, the central questions are usually framed as: how a system runs successfully, how it generates valid structures from an initial state, and how it constructs an overall "world" via evolution rules.

Within these research frameworks, failure, deviation, non-closing orbits, and illegal paths are often treated as objects to be excluded or ignored. Even when they are discussed, they typically appear only as "errors", "noise", or "exceptional cases", and are not granted an independent structural status.

This paper adopts a stance fundamentally different from that traditional route. We no longer treat failure as an auxiliary phenomenon. Instead, we elevate it to a first-class formal object, and systematically study the following questions:

Can failure be formalised as an independent state space?

Does failure possess a composable, accumulative structure?

Under what conditions can failure be eliminated completely?

Based on this stance, we propose and study an independent finite-state formal system, called **Residual Dynamics (Residual Dynamics, abbreviated RD)**.

The core idea of residual dynamics is not to describe "how a system step by step generates a world", but rather to characterise "how, during the attempt to generate a world, the degree of deviation from world-consistency requirements is produced, how it changes, and how it is eliminated."

Under this perspective, the world is no longer the starting point of evolution, but the limiting state in which failure disappears completely.

2 The Basic State Space of Residual Dynamics

This section will strictly and step by step define the state space used by the residual dynamics system. All objects will be introduced explicitly in this section, and no new basic types will be introduced in later sections.

2.1 Definitions of Modular Sets

First define two basic modular sets:

$$\mathbb{Z}_2 = \{0, 1\}, \qquad \mathbb{Z}_5 = \{0, 1, 2, 3, 4\}.$$

Here, $\mathbb{Z}_2$ is used to describe binary deviations, such as flips and parity changes; $\mathbb{Z}_5$ is used to describe phase offsets with a 5-period structure.

In this paper these two sets are used only as finite algebraic structures, with no physical or semantic meaning assumed in advance.

2.2 Formal Definition of the Residual State Space

Define the state space of residual dynamics as the following product set:

$$\mathcal{R} := (\mathbb{Z}_2 \times \mathbb{Z}_5)^2.$$

That is, any residual state $r \in \mathcal{R}$ consists of two ordered pairs.

We write it as:

$$r = (r_5, r_{10}),$$

where

$$r_5 = (r_5^b, r_5^p) \in \mathbb{Z}_2 \times \mathbb{Z}_5, \qquad r_{10} = (r_{10}^b, r_{10}^p) \in \mathbb{Z}_2 \times \mathbb{Z}_5.$$

The subscripts 5 and 10 are used only as labels, to distinguish residual components defined at different observation scales. Formally they are completely symmetric, and no ordering relation between them is presupposed.

Therefore, the residual state space $\mathcal{R}$ is a finite set, with theoretical size:

$$|\mathcal{R}| = (2 \times 5)^2 = 100.$$

2.3 Intuitive Explanation (Non-Formal Part)

To aid understanding, one may regard r_5 as the offset produced by the system relative to ideal closing behaviour under a shorter observation window, and regard r_{10} as the offset produced by the system relative to ideal closing behaviour under a longer observation window.

It must be stated explicitly: the above explanation is only for intuition. In the subsequent formal derivations and checks, all conclusions depend only on the sets and algebraic definitions above, and do not depend on any semantic interpretation.

3 The Algebraic Composition Structure of Residuals

This section defines the most central operational structure in residual dynamics: how residuals are "composed (accumulated)". This composition is not an arbitrary rule. It is designed so that the "total amount of failure" resulting from concatenating multiple failure fragments can be described consistently. Therefore, we must provide an operation on the residual space that is closed, computable, and composable.

3.1 Base Composition on $\mathbb{Z}_2 \times \mathbb{Z}_5$

Each half-component of a residual state, r_5 and r_{10}, lies in $\mathbb{Z}_2 \times \mathbb{Z}_5$. So we first define a base composition operation on this set.

Let

$$x = (x^b, x^p) \in \mathbb{Z}_2 \times \mathbb{Z}_5, \qquad y = (y^b, y^p) \in \mathbb{Z}_2 \times \mathbb{Z}_5.$$

Where:

- $x^b, y^b \in \mathbb{Z}_2$ are the binary components;
- $x^p, y^p \in \mathbb{Z}_5$ are the 5-period components.

Define the base composition operation $\oplus$ by:

$$x \oplus y := (x^b \oplus y^b,\ (x^p + y^p) \bmod 5),$$

where the $\oplus$ on the left denotes XOR on $\mathbb{Z}_2$, and the $+$ and mod5 on the right denote addition modulo 5 on $\mathbb{Z}_5$.

The meaning of this definition is:

- The binary component composes via XOR: the parity of the number of flips accumulates;
- The 5-period component composes via addition mod 5: the phase offset accumulates in the cyclic group.

We emphasise: no physical interpretation is required here. This is merely a computable formal structure for "how two deviations add up".

3.2 Total Composition Operation on the Residual Space $\mathcal{R}$

Now we lift the composition structure to the full residual space

$$\mathcal{R} = (\mathbb{Z}_2 \times \mathbb{Z}_5)^2.$$

Take any two residual states

$$r = (r_5, r_{10}) \in \mathcal{R}, \qquad s = (s_5, s_{10}) \in \mathcal{R},$$

where

$$r_5, s_5 \in \mathbb{Z}_2 \times \mathbb{Z}_5, \qquad r_{10}, s_{10} \in \mathbb{Z}_2 \times \mathbb{Z}_5.$$

Define the total composition operation $\boxplus$ on the residual space by:

$$r \boxplus s := (r_5 \oplus s_5,\ r_{10} \oplus s_{10}).$$

That is: the "short-window component" and the "long-window component" compose independently, and then are paired to form a new residual state.

3.3 Zero Residual and Identity Element

In the residual space, we need a special state representing "no deviation at all", which will serve as the identity element of the composition.

Define the zero residual as:

$$\mathbf{0} := ((0,0),(0,0)) \in \mathcal{R}.$$

Where:

- $(0,0) \in \mathbb{Z}_2 \times \mathbb{Z}_5$ means the binary deviation is 0 and the phase offset is 0;
- Both components being $(0,0)$ means both the short-window and long-window residuals are zero.

From the definition one can directly verify: for any $r \in \mathcal{R}$,

$$r \boxplus \mathbf{0} = r, \qquad \mathbf{0} \boxplus r = r.$$

Hence $\mathbf{0}$ is the identity element of $\boxplus$.

3.4 Remarks on Closure and Composability

All subsequent evolution rules in residual dynamics will be built on the operation $\boxplus$. Therefore we make explicit:

- **Closure:** If $r, s \in \mathcal{R}$, then $r \boxplus s \in \mathcal{R}$. This is because $\mathbb{Z}_2$ is closed under XOR, and $\mathbb{Z}_5$ is closed under modular addition, hence $\mathbb{Z}_2 \times \mathbb{Z}_5$ is closed under $\oplus$, and therefore $\mathcal{R}$ is closed under $\boxplus$.
- **Composability:** The composition operation allows us to accumulate the residual contributions of multiple failure fragments, while still representing the result as a residual state. This makes "accumulated failure" a strict formal object, rather than a vague qualitative description.

4 Drive Elements and Residual Evolution Rules

This section introduces the second core object of residual dynamics: the drive element. A drive element represents a "local fragment" injecting a deviation into the residual space. Evolution in residual dynamics is the accumulation of drive elements within the residual space.

4.1 Definition of the Set of Drive Elements

Let

$$\mathcal{E} \subseteq \mathcal{R}$$

be a given finite set, called the **set of drive elements**.

The role of $\mathcal{E}$ is: to specify which residual increments are permitted to be injected into the system.

Equivalently, if we view a residual state as the "current degree of failure", then a drive element $e \in \mathcal{E}$ corresponds to a "failure injection event", which advances the system from the current residual state r to a new residual state.

4.2 Origin and Independence of Drive Elements (Must Be Made Clear)

The residual dynamics system itself does not prescribe how $\mathcal{E}$ is obtained. $\mathcal{E}$ may come from any external system, for example:

- a local orbit fragment of a finite-state automaton;
- a formal system with local step rules;
- or any generative mechanism that maps local fragments to residuals.

Residual dynamics requires only: that the external system can provide some "fragments", and compute a residual for each fragment, thereby producing a residual set $\mathcal{E}$. The external system's states, rules, and semantics are not internal objects of residual dynamics.

This is the sense in which "residual dynamics is independent of the generating system": RD does not study how the external system generates fragments. It studies only how, once those fragments are mapped into residuals, they accumulate and evolve within the residual space.

4.3 Residual Evolution Relation $\Rightarrow$

Define a binary relation

$$\Rightarrow \subseteq \mathcal{R} \times \mathcal{R},$$

called the residual evolution relation.

Intuitively, $r \Rightarrow r'$ means: the system can evolve from residual state r to residual state r'.

We specify how this relation is generated in the form of inference rules. The advantage of inference rules is: they explicitly specify the allowed evolution steps, and they can be composed to obtain multi-step evolution conclusions.

4.4 Basic Inference Rule: Residual Step

(RD-STEP) Residual Step Rule For any $r \in \mathcal{R}$ and any $e \in \mathcal{E}$, define a one-step evolution rule:

$$r \Rightarrow r \boxplus e.$$

The meaning of this rule is fully explicit:

- the current residual is r;
- inject an allowed drive element e (i.e. an allowed failure increment);
- the new residual becomes $r \boxplus e$.

Therefore, a "step" in residual dynamics is not a one-step state transition of the original system, but an event of "concatenating one failure fragment".

4.5 Reflexivity and Compositionality (Ensuring Multi-Step Evolution Is Expressible)

To discuss "multi-step evolution", we add two standard closure rules.

(RD-REFL) Reflexivity Rule For any $r \in \mathcal{R}$, we stipulate:

$$r \Rightarrow r.$$

This rule means: zero injections are allowed, i.e. staying unchanged.

(RD-COMP) Composition Rule For any $r, r', r'' \in \mathcal{R}$, if

$$r \Rightarrow r' \quad \text{and} \quad r' \Rightarrow r'',$$

then we stipulate:

$$r \Rightarrow r''.$$

This rule means: if one can evolve from r to r', and also from r' to r'', then one can evolve from r to r''.

This guarantees that the residual evolution relation is "concatenable", allowing us to describe arbitrarily long failure-accumulation processes by finite-step reasoning.

4.6 What This Section Has Established

Up to this point, the residual dynamics system already has three ingredients:

- a clearly defined finite state space $\mathcal{R}$;
- a clearly defined residual composition operation $\boxplus$;
- a clearly defined one-step evolution rule: $r \Rightarrow r \boxplus e$ (where $e \in \mathcal{E}$).

This means: once a concrete drive-element set $\mathcal{E}$ is given, residual dynamics becomes a fully determined formal system. Its evolution occurs entirely inside the residual space, and no longer depends on the external system's states or rules.

5 World, Success Criterion, and Absorbing State

In the previous two sections we completed the core formal structure of residual dynamics: we gave the finite state space $\mathcal{R}$, defined the residual composition operation $\boxplus$, and specified via inference rules how residuals evolve under injection of drive elements.

This section aims to answer a key question: *in residual dynamics, what is called "success", and what is called the "world"?*

Unlike traditional systems, success here is not defined by "generating some object", but by whether failure has been eliminated completely.

5.1 Definition of the Zero-Residual Slice

Recall the definition of the zero residual:

$$\mathbf{0} := ((0,0),(0,0)) \in \mathcal{R}.$$

This state means that at all considered observation scales, the system has no remaining deviation.

Based on this, we define the **zero-residual slice** (also called the "world slice") as the set:

$$\mathcal{W}_0 := \{\, r \in \mathcal{R} \mid r = \mathbf{0} \,\}.$$

Because $\mathbf{0}$ is a unique element, $\mathcal{W}_0$ is in fact a singleton subset of the residual space. However, it is exactly this single point that carries the entire semantics of "success state" and "world state" in residual dynamics.

5.2 Why Zero Residual Is Used as the Success Criterion

In the context of residual dynamics, any nonzero residual state indicates: the system still deviates, at some scale, from world-consistency requirements.

Therefore,

- if $r \neq \mathbf{0}$, then the system has not yet succeeded;
- if $r = \mathbf{0}$, then the system satisfies consistency requirements at all scales.

This definition has the following important properties:

- the success criterion is **endogenous** to the system, and does not depend on any external semantic interpretation;
- success is not a process, but a decision outcome;
- the definition of success does not involve "the generating path", only whether "any failure residue remains".

5.3 The Concept of an Absorbing State

In residual dynamics, we care about the following question: once the system reaches the zero-residual state, can it later deviate again?

For this, we introduce the notion of an absorbing state.

If, in the absence of injecting a new nonzero drive element, once the system enters some state it cannot leave it, then that state is called an **absorbing state**.

In this system, the zero residual $\mathbf{0}$ has the following property:

$$\mathbf{0} \boxplus \mathbf{0} = \mathbf{0}.$$

Moreover, if the drive-element set $\mathcal{E}$ contains no element other than $\mathbf{0}$, then without external injection, once the system reaches the zero-residual state, it will remain there unchanged.

Thus, the zero residual naturally plays the role of an absorbing state in residual dynamics.

5.4 Summary of the Formal Meaning of "World"

Combining the above definitions, in residual dynamics:

> The world is not an object "generated" by evolution rules, but the state in which failure has been completely eliminated.

This means:

- the world is not the starting point of the dynamics;
- the world is not an orbit of the dynamics;
- the world is the terminal criterion of residual dynamics.

This stance is fundamentally different from the intuition of traditional dynamical systems, but it is exactly this difference that allows failure to be elevated to a first-class research object.

6 Residual Dynamics as a Formal System: Summary

Before entering the exhaustive verification of a concrete example, we first give a complete and precise summary of residual dynamics as a formal system, to clarify which objects have already been fixed, and which aspects will be made concrete in the example.

6.1 Constituent Elements of the Formal System

A residual dynamics system is fully determined by the following data:

1. a finite state space
$$\mathcal{R} = (\mathbb{Z}_2 \times \mathbb{Z}_5)^2;$$
2. a composition operation $\boxplus$ that is closed on $\mathcal{R}$;
3. a given finite set of drive elements
$$\mathcal{E} \subseteq \mathcal{R};$$
4. a set of inference rules generating the residual evolution relation $\Rightarrow$.

Once these elements are given, all reachable states, all evolution paths, and the success criterion are completely determined in a formal way.

6.2 Relation to an External Generating System

It must be emphasised again: residual dynamics does not describe the external system's state space, local rules, or evolution details.

The external system's only role in residual dynamics is: to compute residuals for some "local fragments", thereby providing a drive-element set $\mathcal{E}$.

Once $\mathcal{E}$ is given, the rest of the external system's structure is completely forgotten, and residual dynamics becomes an independently running formal system.

6.3 What We Will Do Next

In the next section, we will choose a concrete external system example, and carry out the following steps:

- explicitly specify the external system's state space and local rules;
- enumerate all local orbits of a given length;
- compute a residual for each orbit;
- count reachability and distribution of residuals;
- verify that the zero-residual slice corresponds to the "valid world".

This example will be developed by complete exhaustive enumeration, with no skipped steps and no omissions, providing a checkable concrete support for residual dynamics.

7 A Concrete Example: Fully Exhaustive Verification

This section provides a **step-by-step, fully checkable** concrete example, to verify and demonstrate how residual dynamics operates in a real finite system. We will explicitly write down the system parameters, compute the total number of candidate orbits, define the concrete residual computation, and obtain all reachable residuals and the zero-residual slice by full-space enumeration.

7.1 Parameter Setting of the External System

Consider the following concrete external finite system:

- **State space** is defined as
  $$S = \{0,1\} \times \mathbb{Z}_5.$$
  Hence the system has
  $$|S| = 2 \times 5 = 10$$
  distinct states. We write an arbitrary state as
  $$s = (b,p),$$
  where $b \in \{0,1\}$ is the binary component, and $p \in \mathbb{Z}_5$ is the phase component.

- **Local step rules** allow only the following two atomic rules:

  1. Forward-step rule R_1:
  $$(b,p) \mapsto (b,\ p+1 \bmod 5);$$
  2. Backward-flip rule R_2:
  $$(b,p) \mapsto (1-b,\ p-1 \bmod 5).$$

- **Orbit length** is fixed to 10 steps. For any initial state s_0, an orbit of length 10 is written as
  $$\gamma = (s_0, s_1, s_2, \dots, s_{10}),$$
  where for all $i = 0, \dots, 9$, the state s_{i+1} is obtained from s_i by either R_1 or R_2.

7.2 Total Number of Candidate Orbits

We first compute the total number of all possible candidate orbits under the parameters above.

- For a fixed initial state s_0, at each step there are 2 rules to choose from;
- therefore the number of distinct rule sequences of length 10 is

$$2^{10} = 1024;$$

- the system has 10 possible initial states.

Thus, the total number of candidate orbits is

$$10 \times 1024 = 10240.$$

These 10240 orbits form the complete enumeration space for the residual computation and statistics that follow.

7.3 Concrete Residual Computation Formula

Now we instantiate the abstract residual definition in residual dynamics for this external system.

Let

$$s_i = (b_i, p_i)$$

denote the orbit γ's state at step i.

7.3.1 Definition of the Five-Step Residual

Define the five-step residual of the orbit γ as

$$\varepsilon_5(\gamma) = (b_5 \oplus (1 - b_0),\ (p_5 - p_0) \bmod 5)\,.$$

This definition contains two components:

- the binary component $b_5 \oplus (1 - b_0)$, used to detect whether the "expected flip" is completed within the first five steps;
- the phase component $(p_5 - p_0) \bmod 5$, used to record the net phase offset within the first five steps.

7.3.2 Definition of the Ten-Step Residual

Define the ten-step residual of the orbit γ as

$$\varepsilon_{10}(\gamma) = (b_{10} \oplus b_0,\ (p_{10} - p_0) \bmod 5)\,.$$

Where:

- the binary component $b_{10} \oplus b_0$ checks whether the whole orbit segment completes an even number of flips;
- the phase component $(p_{10} - p_0) \bmod 5$ checks whether the whole segment closes in phase.

7.3.3 Total Residual

Define the total residual of the orbit γ as

$$\varepsilon(\gamma) = \big(\varepsilon_5(\gamma),\ \varepsilon_{10}(\gamma)\big) \in (\mathbb{Z}_2 \times \mathbb{Z}_5)^2.$$

7.4 Criterion for Zero-Residual Orbits

By the previous definition of "world" and "success", an orbit is called a **valid orbit** if and only if it satisfies

$$\varepsilon_5(\gamma) = (0,0) \quad \text{and} \quad \varepsilon_{10}(\gamma) = (0,0).$$

We call all orbits satisfying these conditions **zero-residual orbits**.

7.5 Complete Enumeration Result for Zero-Residual Orbits

Computing residuals for all 10240 candidate orbits one by one, we obtain the following strict conclusions:

- the number of orbits satisfying the zero-residual condition is **exactly** 10;
- these 10 orbits correspond to 10 distinct initial states s_0;
- all these 10 orbits share **the same rule sequence**, namely using the backward-flip rule R_2 at every step of the 10 steps;
- for each zero-residual orbit γ, we have the strict closure relation

$$s_{10} = s_0.$$

Therefore, in this example, the "valid world" is not composed of many structurally different orbits, but rather consists of a family of closed orbits that are *structurally identical and differ only by the starting point.*

7.6 Statistics of Residual Reachability

We further collect statistics of residuals over all 10240 orbits, and obtain the following reachability results:

- the reachable values of the ten-step residual ε_{10} cover the whole set $\mathbb{Z}_2 \times \mathbb{Z}_5$, i.e. 10 possibilities;
- the reachable values of the five-step residual ε_5 do not cover all of $\mathbb{Z}_2 \times \mathbb{Z}_5$, but only 6 possibilities;
- the theoretical total residual space contains 100 possible states, but in this example, the actually reachable total residual states are only 36.

This means: even though the theoretical state-space size of residual dynamics is 100, the drive-element set induced by a concrete system may compress it into a smaller effective subspace.

7.7 Formal Meaning of This Section's Conclusion

Through the above complete exhaustive enumeration, we verify the following facts:

- residual dynamics is not an abstract idea, but can be fully enumerated and verified in a concrete system;
- "world" is equivalent to the zero-residual slice, and in this example that slice is a finite set that can be fully listed;
- the object studied by residual dynamics is not the states themselves, but the failure structure of state orbits under a consistency criterion.

This provides direct, checkable instance support for residual dynamics as an independent formal system.

8 Overall Summary: What Has Residual Dynamics Actually Achieved

In the previous sections, we constructed residual dynamics (RD) from scratch as a formal system, and on a concrete instance with 10 states, 2 local rules, and a 10-step window, we provided checkable verification results by full-space exhaustive enumeration.

This section introduces no new objects. It only provides a **point-by-point, sentence-by-sentence comparable summary** of what has been completed, making explicit what the reader has obtained, what the reader has not obtained, and why this framework can be regarded as a general template for "failure dynamics".

8.1 Formal Objects the Reader Already Has (Item-by-Item Confirmation)

By the end of Section 7, this paper has provided the following fully explicit formal objects.

(1) An Independent Residual State Space We defined the residual dynamics state space as

$$\mathcal{R} = (\mathbb{Z}_2 \times \mathbb{Z}_5)^2,$$

and wrote any residual state as

$$r = (r_5, r_{10}).$$

This space is finite, with theoretical size 100. This step provides a purely formal object: it does not depend on how the external system defines its states, nor on the external system's local rules.

(2) A Residual Composition Operation We defined the composition operation $\boxplus$ on $\mathcal{R}$, and explicitly gave its component-wise computation. This operation guarantees that accumulation of residuals can be formalised, and guarantees that "concatenating multiple failure fragments" still stays within the same residual space.

(3) A Drive-Element Set and an Evolution Relation We introduced the drive-element set

$$\mathcal{E} \subseteq \mathcal{R},$$

and defined the residual evolution relation $\Rightarrow$ to be generated by inference rules, with the core one-step rule:

$$r \Rightarrow r \boxplus e \qquad (e \in \mathcal{E}).$$

Therefore, once $\mathcal{E}$ is specified, the reachable states, reachable paths, and closure of the evolution relation are all determined.

(4) World / Success Criterion We used the zero residual

$$\mathbf{0} = ((0,0),(0,0))$$

as the success criterion, and defined the world slice as

$$\mathcal{W}_0 = \{r \in \mathcal{R} \mid r = \mathbf{0}\}.$$

In this framework, "world" is no longer defined by a generation process, but by whether failure has been eliminated completely.

8.2 What This Paper Intentionally Does Not Do (Boundary Statement)

To avoid misunderstanding, we explicitly state that this paper **does not** do the following:

- it does not interpret residuals as physical energy, entropy, or any external meaning;
- it does not treat any particular external system (e.g. some automaton or some dynamical system) as the only source;
- it does not promise that residuals must converge to zero, nor claim that systems necessarily converge;
- it does not replace formal enumeration with any probability, statistical fitting, or heuristic method.

In other words, the goal is not to provide a worldview, but to provide a **auditable, checkable, and falsifiable** formal framework.

8.3 Why This Forms a General Template for "Failure Dynamics"

The most crucial structural conclusion of this paper can be stated as:

> As long as an external system can map "local fragments" into a finite residual space, and residuals can be composed and accumulated within that space, then the study of how failure adds up, how it propagates, and how it is eliminated, can be fully rewritten as an internal problem inside residual dynamics.

Therefore, the external system's role is compressed into one thing: **to provide the drive-element set $\mathcal{E}$.**

Once $\mathcal{E}$ is fixed, the external system's states, rules, and semantic details cease to be necessary inputs. Residual dynamics becomes an independent object, enabling purely formal study of failure structures.

8.4 Difference from the Traditional View of "A Single Evolution Function Generates the World" (Emphasised Again)

The traditional view often assumes that "the world" arises from iterating some evolution function, or from solution trajectories of some time-evolution equation.

The stance of this paper is:

- local rules generate only candidate fragments (which may be correct, or may fail);
- global consistency requirements are not realised by generation, but by a residual criterion;
- the world is not a "generated result", but the "slice where failure is zero".

Thus, "generation" and "legality / consistency" are structurally separated, and this is the fundamental reason residual dynamics can stand as its own system.

8.5 One Last Sentence About the Section 7 Example (Formal Level)

The exhaustive verification in Section 7 was not to claim that some number is "nice", but to prove:

- in an extremely simple finite system, residuals and the world slice can be **fully enumerated**;
- the set of zero-residual orbits can be **fully characterised**;
- the reachable residual set can be much smaller than the theoretical space, thereby giving a concrete meaning to the "effective state space" of failure dynamics.

These three points are completion metrics in the sense of a formal system: they ensure the reader can check, reproduce, and also refute.

Acknowledgements and Notes (Optional)

This paper does not rely on any statistical fitting or external semantic interpretation. If the reader wishes to extend further, one may keep the RD formal structure unchanged, and swap the external system to obtain a different drive-element set $\mathcal{E}$, thereby comparing the failure-structure differences across systems. Such an extension belongs to the application layer, not the formal layer.