The history space — the Platonic object behind the session

Closing theoretical companion to `SESSION_FINDINGS_2026_04_11_12.md`.

The one-line claim

> The Platonic object is not the compressor, not the swarm, not the Hamiltonian, and not the state space. **It is the weighted space of admissible histories with boundary conditions.** Everything else is a chart on it.

Or, harder:

> **Het platonische object is de geschiedenisruimte; alles anders is een chart.**

Formal definition

```
h = (Gamma_adm, d_in, d_out, A, C)
```

where

- **Gamma_adm** — the set (or groupoid) of admissible histories: sequences of steps through a guarded bulk, consistent with the open dynamics `U, R` and every applicable Hoare-style corridor
- **d_in : Gamma_adm -> Sigma*** — incoming boundary projection
- **d_out : Gamma_adm -> Sigma*** — outgoing boundary projection
- **A : Gamma_adm -> Q>=0** — action functional (= local costs summed along the history)
- **C** — cochain / topological data carried by the history skeleton (exact / harmonic / residual decomposition, charge, current, potential)

The Lean symbolic side of `Gamma_adm, d_in, d_out, A` is already in `Proof/Scattering.lean`:

- `History X Sym U E`
- `WellFormed D gamma` + `Admissible D g gamma`
- `inboundary gamma`, `outboundary gamma`
- `action L gamma` with `action_nil`, `action_cons`, `action_append`

What the history space adds is the *totality*: not one gamma, but the indexed family of all admissible gamma with fixed boundary conditions.

The generating function

```
Z(lambda; b^in, b^out) = Sigma_{gamma in Gamma_adm,  d_in gamma = b^in,  d_out gamma = b^ou
                         exp(-lambda . A(gamma))
```

At `lambda = 1` this is the unnormalised scattering weight already proven correct for the deterministic codec case in `research/tlc/scattering.py`. As a function of lambda it is a spectral object: its asymptotics encode entropy rate, its poles encode phase transitions, its derivatives encode expected action.

When Gamma_adm collapses to a single admissible path (a deterministic codec with fixed `(b^in, b^out)` by encode/decode correctness), Z is a single `exp(-A)`. When the guards allow branching — nondeterministic routing, stochastic corrections, MCTS lookahead — the sum has real content.

Every session artefact is a chart

Each file in `research/tlc/` and each theorem in `Proof/` is a LOCAL description of h. None of them is h itself.

Artefact	Chart on h	What it exposes
`tlc_compressor.py`, `ArithCoder.Model`	single deterministic gamma	boundary coding of one history
`dual_compressor.py`	same gamma in Q[epsilon]/(epsilon?)	multiplicative action form
`casimir_compressor.py`	families of gamma with different guard structure	DeltaL as comparison of two charts
`atlas_spectrum.py`, `RepresentationalAtlas.lean`	ladder of `FiniteFamily` + advantage curves	the measured projection of h onto rungs
`hyperbolic_compressor.py`	local chart where branching is geometrically cheap	hyperbolic coordinates
`open_kernel.py`	EncoderKernel / DecoderKernel	the same gamma with swapped ports
`scattering.py`, `Scattering.lean`	enumerator + Sigma exp(-A) over a finite slice of Gamma_adm	the sum-over-histories directly
`swarm_model.py`, `SwarmModel`	measure on a local chart of Gamma_adm	Bayesian filter over charts
`charged_swarm.py`	same measure + field coupling	rho -> phi -> drift, adds topological charge
`field_coupled_swarm.py`	same + field feeds into emission	C is feeding back into A
`guarded_mcts.py`	PUCT exploration of a subtree of Gamma_adm	search within the history space
`thermal_swarm.py`	Z(lambda) at various lambda	finite-temperature slices
`quantum_swarm.py`	replace Sigma exp with |Sigma e^{iPhi}|?	Born rule version of Z
`solomonoff_swarm.py`	`2^(-L)` prior over generators of histories	universal-prior sum
`adversarial_swarm.py`	saddle point of Z over boundary data	minimax on h
`Proof.ArithCoder.Model.encode_decode_id`	gamma is uniquely determined by (d_in, d_out)	deterministic-codec corner
`Proof.RepresentationalAtlas`	`advantageCurve`, `coordinateSusceptibility`	ladder-relative projections of Z
`Proof.CasimirBridge.casimirRatio_lt_one_iff_width_lt`	ordering of charts <-> ordering of Z-ratios	the Q<->N handshake

In this reading, the whole session's output is one book of charts. The book is finite and the reader can pick any chart for any local measurement, but the object described is not any of the charts. It is their colimit.

The Ramanujan program

The stance: stop asking "what is the next state?" and ask "how does the space of admissible histories count?" The questions that become sensible are:

1. **Dirichlet series of admissible histories.**

   ```
   D(s; b^in, b^out) = Sigma_gamma  A(gamma)^(-s)
   ```

   Well-defined when A(gamma) has discrete values (integer bit counts or bounded Q grids). Its abscissa of convergence encodes the growth rate of admissible histories by action. For a codec on a bounded alphabet this number is the byte-conditional entropy rate.

2. **Euler product over local scattering vertices.** Each step `(x_t, b^in_t, u_t, epsilon_t) -> (x_{t+1}, b^out_{t+1})` is a local vertex. If the guards factor through local-at-each-step constraints, then Z factorises as a product over steps, with each factor a finite sum. That is the Euler side of the history-zeta:

   ```
   Z = Prod_{step}  Sigma_{u, epsilon valid here}  exp(-L(.))
   ```

3. **Modular / dual symmetry between encode and decode.** The open-kernel port-swap already operational in `open_kernel.py` is a Z_2 symmetry: relabel boundary in <-> out and run the same dynamics. In the generating function this should be an involution Z(lambda; b^in, b^out) -> Z(lambda; b^out, b^in) — a functional equation for the coder/decoder pair. The Lean statement candidate:

   ```
   casimirRatio m_1 m_2 = (casimirRatio m_2 m_1)??
   ```

   (trivially true from the definition, but semantically: encoder and decoder are charts of each other.)

4. **Poles of Z(lambda).** Thermal-swarm style scan of lambda = 1/T on a non-stationary corpus. A pole in (extrapolated) Z(lambda) at some lambda_c means a phase transition of the computation — a regime switch. The session's `thermal_swarm.py` measured no peak on stationary 1 KB because one particle already dominates; the prediction is that on a regime-mixed corpus Z(lambda) will show real critical structure.

5. **Functional equation for the codec.** A dual relation like

   ```
   Z(lambda; b^in, b^out) = f(lambda) . Z(1 - lambda; b^out, b^in)
   ```

   would be the sharp form of encoder-decoder duality. Speculative; no concrete candidate yet.

Why this framing is the correct level

The preceding sessions produced a long list of related artefacts:

- A proof of round-trip correctness (`decode_message_correct`)
- A scattering sum enumerator (`scattering.py`)
- An MDL ladder with advantage curves (`RepresentationalAtlas.lean`)
- Swarm models with Bayesian, quantum, charged, thermal variants
- An open-kernel port-swap
- A Poincaré-ball geometric compressor
- A Casimir boundary-pressure spectrum

None of these feel like the underlying object. Each lives inside the others in some way: the scattering sum contains the codec as its deterministic corner; the swarm measure projects onto the scattering sum; the ladder projects the swarm onto its coarse costs; the hyperbolic chart gives a local parametrisation of the branching tree; the Casimir shifts are comparisons between two choices of bulk geometry.

The smallest object that contains all of them *as projections* is the history space h. It is the only thing in this session that is not a chart on something else.

Operational consequences

Three types of inquiry that become well-defined once h is the target:

- **Counting** instead of sampling. The MCTS tree search, the scattering sum, and the Solomonoff prior are all already sums over sub-regions of Gamma_adm. Any quantitative improvement comes from computing those sums more cleanly.
- **Boundary fibres as the natural unit of analysis.** A Casimir measurement is a comparison of two boundary fibres (two choices of d_in/d_out structure). A compression ratio is the action of one gamma normalised by the log-size of one fibre. An entropy rate is the log-growth of fibres.
- **Asymptotic spectral data.** Instead of asking "does this compressor beat gzip at 10M?", ask "what is the asymptotic leading singularity of Z(lambda) at lambda -> lambda_c?" The former is a measurement on one chart; the latter is an invariant of h itself.

The formal summary

```
Computation          = choice or sum over gamma in Gamma_adm
Execution            = min-action gamma in Gamma_adm
Generation           = sample from e^(-A) / Z
Compression          = boundary coding of a selected gamma
Physics-analogy      = interpretation of (Gamma_adm, A) as open
                       scattering
Training             = parameter tuning so data histories sit low
                       in A
Search (MCTS, ...)     = partial enumeration of Gamma_adm
Field dynamics       = C evolving through the history, feeding
                       back into A
```

And a final compact formalism:

```
Z(lambda; b^in, b^out) = Sigma_{gamma in Gamma_adm}
                    exp(-lambda A(gamma))
                    1[d_in gamma = b^in]
                    1[d_out gamma = b^out]
```

The session's job was to build one concrete coordinate system on this object. The next session's job — if taken this direction — is to study h itself: its Dirichlet series, its Euler factorisation, its dualities, its spectral data. The arithmetic of admissible histories.

The closing slogan

> **State, field, swarm, charge, code, and action are all coordinate systems on one history object.**

And the compact form from which the whole session unfolds:

> **The Platonic object is the space of admissible histories weighted by their action.**