The Spectral Information State

A Common State for Inference, Learning, and Decision

A mode-level state object that unifies uncertainty, complexity, and action in spectral problems

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Draft • March 2026

Executive Summary (Non-Technical)

Modern inference often looks more fragmented than it really is. One workflow produces confidence intervals, another produces posterior distributions, another chooses model size, another estimates decision weights. In practice these are often treated as separate tools, with separate debates and separate tuning cultures.

This paper proposes that in spectral problems there is a simpler core object underneath them all: the **spectral information state**. For each resolved mode, it records two things: the estimated strength of the mode and the uncertainty that remains around it. Once that pair is known, many familiar inferential outputs become different readouts of the same local state.

The main contribution of the paper is therefore not another estimator or another penalty. It is a **common state description** that can be reused across inference, model selection, compression, and decision. The point is to identify the smallest durable object that survives after eigendecomposition and that still supports the most important downstream questions.

This paper is a **foundational theory paper**, not the bridge paper about philosophical schools. A companion manuscript, The Duality of Bayesian and Frequentist Statistics, uses this state object to show that Bayesian, frequentist, and MDL model-complexity rules converge to the same practical cutoff in spectral problems. Here the goal is earlier in the chain: define the object, justify why it deserves to be treated as a state, and map what it can support.

The paper does **not** claim that every statistical problem is already solved by one eigendecomposition, or that causal, structural, and domain-modeling questions disappear. It claims something narrower and more useful: once a problem has a meaningful spectral basis, there is a canonical mode-level state that organizes what the data currently knows, how much uncertainty remains, and which downstream decisions are justified.

Short Abstract

This paper introduces the **spectral information state** as the canonical mode-level object carried by a spectral learning problem. For each mode, the state is the pair $\psi_k = (\widehat{A}_k, \sigma_k^2)$ of estimated coefficient and residual uncertainty. In Gaussian linear settings this is a complete sufficient statistic for the mode; more generally it is the minimal reusable local state from which a large inferential menu can be read out in closed form or near-closed form. Confidence intervals, posterior shrinkage,

Bayes factors, MDL penalties, effective sample size, information gain, and mode-level decision weights all become different functions of the same state. The paper's contribution is to define this object, show why it deserves to be treated as a state rather than a notational convenience, relate it to the Universal Spectral Representation Theorem, and position the Bayesian-frequentist duality as one downstream consequence rather than the primary identity of the construction.

Abstract

We introduce the **spectral information state** as a canonical mode-level state object for spectral inference. After eigendecomposition, each mode k carries an estimated coefficient $\widehat{A}_k$ and a residual uncertainty σ_k^2 induced by finite sample size, noise level, and eigenvalue strength. We define

$$\psi_k = (\hat{A}_k, \sigma_k^2)$$

and argue that this pair is the correct reusable local state for inference in the spectral basis. In Gaussian linear models it is a complete sufficient statistic for the mode. More generally, it is the minimal practical state needed to support the main inferential readouts associated with that mode: confidence intervals, posterior means and variances, Bayes factors, minimum-description-length penalties, effective sample size, information gain, and decision weights.

The paper makes four claims. First, the spectral information state is not Bayesian or frequentist in itself; it is the common substrate on which both frameworks act. Second, many quantities that are usually computed by separate workflows are readouts of the same state once the eigendecomposition is fixed. Third, the state evolves with sample size as a progressive spectral collapse: high-information modes resolve first, boundary modes later, and noise modes remain unresolved. Fourth, the state connects naturally to the Universal Spectral Representation Theorem, which supplies an objective decay prior and an explicit complexity frontier.

The goal of the paper is foundational rather than polemical. We do not claim that the spectral information state replaces all of statistics, or that every problem admits a faithful spectral lens. We claim that whenever a spectral lens is meaningful, the pair $(\widehat{A}_k, \sigma_k^2)$ is the right primitive state object for organizing inference, complexity, and decision. The companion bridge paper The Duality of Bayesian and Frequentist Statistics is then read as a corollary of this deeper object: once both traditions act on the same state, much of their practical disagreement collapses to a thin finite-sample boundary layer.

---

1. Introduction

Statistical workflows often look fragmented because they are organized by output rather than by state. One pipeline produces confidence intervals. Another produces posterior distributions. Another selects model size. Another computes compression penalties or decision weights. Each has its own notation, its own software, and often its own school of interpretation.

In spectral problems this fragmentation is misleading. Once a learning problem is written in its eigenspace, the central local question is repeated mode by mode: what is the current estimated strength of this mode, and how uncertain is that estimate? The claim of this paper is that the answer to that question is itself the correct primitive object.

We call that object the **spectral information state**. For mode k, it is the pair

$$\psi_k = (\hat{A}_k, \sigma_k^2),$$

where $\hat{A}_k$ is the estimated coefficient carried by the mode and σ_k^2 is the residual uncertainty associated with that estimate after accounting for sample size, noise, and spectral strength.

The paper's thesis is not that this notation is occasionally convenient. It is that this pair should be treated as a real state object. Once it is known, many downstream inferential quantities become straightforward readouts rather than separate estimation problems. The result is a cleaner ontology for spectral inference:

1. the eigendecomposition determines the coordinates,
2. the spectral information state records what is currently known in each coordinate,
3. Bayesian, frequentist, MDL, and decision-theoretic outputs become functions of that state.

1.1 What this paper contributes

This paper contributes four things.

First, it defines the spectral information state and explains why it deserves to be treated as a state rather than as a bookkeeping pair.

Second, it gives a reusable inferential menu built from that state. The point is not that every quantity is new, but that they can be generated from one common object instead of from disconnected workflows.

Third, it shows that the state evolves with data as a **progressive spectral collapse**: high-information modes resolve first, boundary modes later, and unresolved modes remain explicitly visible rather than being silently mixed into a single error term.

Fourth, it locates the state inside the broader repo program. The spectral information state sits between the representational theorem layer and the bridge-paper layer:

$$\text{USRT} \longrightarrow \text{Spectral Information State} \longrightarrow \text{Bayesian/Frequentist Duality.}$$

1.2 What this paper does not claim

This paper does **not** claim that all statistical questions reduce to one scalar threshold, or that every model should be spectralized regardless of structure. It also does not claim that causal identification, experimental design, or domain semantics disappear once a spectral basis is chosen. The claim is narrower: if a problem admits a meaningful spectral basis, then there is a canonical mode-level state that should organize what the data currently knows and what downstream actions are justified.

1.3 Relation to the Spectral Time Paper

This paper is designed to pair with Spectral Time: Optimal Stopping, First Passage, and Subordination via the Fo

- Spectral Time is the dynamics/computability layer: it asks when path history admits finite-state closure and generator-based propagation.
- The Spectral Information State (this paper) is the inferential layer: it defines what is known, mode by mode, once representation and propagation are fixed.

Programmatically:

$$\text{dynamical closure and propagation} \longrightarrow \text{mode-level information state} \longrightarrow \text{inferential and decision readouts.}$$

If dynamic closure fails in a regime, the same information-state ontology still applies, but propagation must be hybridized rather than treated as pure spectral.

---

2. Definition of the State

2.1 The mode-level object

Let a learning problem admit a spectral decomposition with mode strengths λ_k and mode coefficients A_k. Given data, the observed response induces an estimated coefficient $\widehat{A}_k$ for each mode. Finite sample size and noise induce a residual uncertainty

$$\sigma_k^2 = \frac{\sigma_{\text{noise}}^2}{n\lambda_k}$$

in the simplest Gaussian linear setting, with analogous local uncertainty quantities in more general spectral models.

We define the **spectral information state** of mode k to be

$$\psi_k = (\widehat{A}_k, \sigma_k^2).$$

This pair records the two things that matter locally:

- how large the resolved signal currently appears to be,
- how uncertain that local estimate still is.

2.2 Why call it a state?

We call ψ_k a **state** for three reasons.

First, it is **local**: it belongs to one mode and can be updated as new data arrive.

Second, it is **reusable**: many inferential outputs are functions of it.

Third, it is **decision-relevant**: it tells us whether a mode is clearly signal, clearly noise, or still unresolved.

In Gaussian linear models, $(\hat{A}_k, \sigma_k^2)$ is a complete sufficient statistic for the mode. Outside that exact setting, we use the term in a broader practical sense: it is the canonical local state object that carries enough information for the principal inferential readouts associated with the mode.

2.3 State of a whole model

For the first K resolved modes, the global spectral state may be written as

$$\Psi_K = \{\psi_1, \dots, \psi_K\}.$$

The individual mode-level objects remain primary. The model-level state is just their organized collection.

---

3. The Inferential Menu

Once ψ_k is available, many familiar outputs become readouts of the same local state.

Output	Representative formula	Readout type
t-statistic	$t_k = \hat{A}_k/\sqrt{\sigma_k^2}$	frequentist significance
confidence interval	$\hat{A}_k \pm 1.96\sqrt{\sigma_k^2}$	frequentist uncertainty
posterior mean	$h_k\hat{A}_k$	Bayesian shrinkage
posterior variance	$h_k^2\sigma_k^2$	Bayesian uncertainty
Bayes factor	$BF_k(\hat{A}_k, \sigma_k^2, \tau_k)$	evidence for signal
information gain	$\frac{1}{2}\log(1 + \hat{A}_k^2/\sigma_k^2)$	bits learned
effective sample size	$n_{\text{eff},k} = n\lambda_k/\sigma^2$	local data budget
MDL increment	$\frac{1}{2}\hat{A}_k^2/\sigma_k^2 + \frac{1}{2}\log n$	coding cost
decision weight	$w_k \propto \hat{A}_k/\sigma_k^2$	action sizing

The novelty is not that each row is individually unknown. The novelty is that all of them are anchored in one shared state object.

3.1 A common substrate, not a school

The spectral information state is not itself Bayesian or frequentist. It is prior to that split. A Bayesian reads it through prior-to-likelihood updating. A frequentist reads it through sampling distributions and coverage. MDL reads it through coding cost. Decision theory reads it through utility-weighted action. None of those readouts changes the underlying local state they are acting on.

3.2 Why this matters

When workflows are organized by output, one often reruns multiple procedures that are all responding to the same resolved local evidence. Treating ψ_k as the primitive object compresses the inferential stack. It also makes disagreements more legible: many of them are not disagreements about signal, but disagreements about how aggressively to act near the unresolved boundary.

---

4. Progressive Spectral Collapse

The spectral information state is dynamic. As the sample size n increases, uncertainties $\sigma_k^2(n)$ shrink, but they do not shrink equally across modes. High-information modes resolve first. Low-information modes resolve later. Some remain unresolved at the current data budget.

This produces a natural picture of learning as **progressive spectral collapse**:

- leading modes become stable early,
- middle modes form a boundary region,
- tail modes remain noise-dominated.

The state therefore records not just what the model currently estimates, but also **which parts of the latent structure are already reliable** and which parts are still speculative.

4.1 The boundary layer

For a given n, there is typically a small set of modes near the effective cutoff where inferential decisions are unstable. Those are the modes where shrinkage, significance, coding cost, and decision weights are most sensitive to assumptions. The existence of this boundary layer is one reason the state object is useful: it keeps unresolved modes explicit instead of forcing a premature binary decision.

4.2 Complexity as a state frontier

The effective complexity cutoff K^* should be understood as a **frontier induced by the spectral state**, not as an externally imposed model-size convention. The state determines where signal has become reliable enough to keep and where additional modes are still dominated by uncertainty.

---

5. Relation to USRT

The Universal Spectral Representation Theorem supplies the deeper representational backdrop. It gives a decay law for spectral coefficients of the form

$$|A_k| \le C\rho^{-k},$$

which in turn induces a natural prior scale

$$A_k \sim \mathcal{N}(0, C^2\rho^{-2k}).$$

This matters for the state story in two ways.

First, it explains why the state is not an arbitrary statistical invention. The coefficient scale and its decay are anchored in a broader representation theorem.

Second, it gives an objective route from representation quality to inferential complexity. The decay rate ρ and the uncertainty budget jointly determine where the reliable state frontier sits.

The state paper therefore occupies a middle layer. USRT explains why a compressed spectral representation exists. The spectral information state explains how data populate that representation under uncertainty. The duality paper then shows that Bayesian and frequentist complexity rules are two readouts of that populated state.

---

6. Companion Consequences

The most immediate companion consequence is the bridge result developed in The Duality of Bayesian and Frequen once both frameworks act on the same mode-level state, their practical complexity rules converge to the same cutoff up to a thin finite-sample boundary layer.

But the object is broader than that one bridge result.

The same state also supports:

- mode-level sample-size targeting,
- overfitting audits,
- reusable compression penalties,
- spectral decision weights,
- and dynamic risk engines such as Bayesian Live Risk.

The purpose of the present paper is to make clear that these are not disconnected tricks. They are downstream consequences of one common local state object.

---

7. Point-Event Information and Resonator Memory

So far the state has been introduced in coefficient-estimation form. A natural question is whether the same state logic survives when information arrives as sparse point events rather than as smooth batched samples.

This section gives a minimal theoretical answer: yes, if the ingestion map into spectral coordinates is made explicit.

7.1 Point-event arrival model

Let arrivals be represented as an impulse stream

$$x(t) = \sum_i a_i \, \delta(t - t_i),$$

with event amplitude a_i and arrival time t_i.

The key claim is that the stored object should not be the raw impulse process itself. The stored object should be a mode-level state induced by a stable operator chain:

$$x(t) \xrightarrow{\mathcal{E}_{\text{kernel}}} u(t) \xrightarrow{\mathcal{E}_{\text{basis}}} c(t) \xrightarrow{\mathcal{E}_{\text{res}}} z(t).$$

7.2 The event-to-state operator chain

We define three stages.

1. **Impulse to field spread**
$$u(t) = (k * x)(t),$$
with a causal kernel k.

2. **Field to spectral coefficients**
$$c_k(t) = \langle u(t, \cdot), \phi_k \rangle.$$

3. **Resonator memory dynamics**
$$\dot{c}_k(t) = -\lambda_k c_k(t) + \beta_k \xi_k(t),$$
where $\lambda_k > 0$ is mode damping and ξ_k is the projected driving term.

The stored information state can then be taken as

$$\psi_k^{\text{dyn}}(t) = (c_k(t), \nu_k(t)),$$

with $\nu_k(t)$ a mode-level uncertainty or residual-energy tracker.

7.3 Why this is a real theoretical extension

This is not just an implementation trick. It extends the static state story in three ways.

First, it gives a principled map from sparse arrivals to the same mode-level ontology used by the paper.

Second, it turns memory horizon into a spectral property: slow modes (small λ_k) retain event mass longer; fast modes decay quickly.

Third, it explains why point-like information appears distributed after ingestion: the event leaves a mode footprint, not a point record.

7.4 A stability condition

If the resonator update is written in discrete time as

$$z_{t+1} = Az_t + B\eta_t,$$

then a minimal sufficient stability condition is $\rho(A) < 1$, where ρ is the spectral radius. Under bounded input energy, this gives bounded state energy and prevents unphysical memory blow-up.

This condition is the dynamic analogue of state hygiene in the static sections: without it, inferential readouts from the state become unreliable regardless of estimator quality.

7.5 Relation to the paper's core claim

The core paper claim remains unchanged: spectral inference is organized by a reusable mode-level state. The point-event extension shows that the same claim survives a harder regime where information is sparse, asynchronous, and impact-like.

In other words, the spectral information state is not limited to "batch coefficient estimation." It also admits a resonator interpretation in which events are transformed into stable mode-memory footprints that support the same downstream readouts (inference, complexity control, and decision).

7.6 Abstract definitions: future, information, and storage

To separate foundation from implementation, we formalize three objects first.

Let

$$(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{Q})$$

be a filtered probability space, and let the future path random element at time t be

$$Y_t := X_{[t,T]} \in \mathcal{Y}.$$

Future

The "future" is not one realized path. It is the random path object Y_t viewed from time t.

Information about the future

The information object is the regular conditional law

$$\kappa_t(\omega, \cdot) := \mathcal{L}_{\mathbb{Q}}(Y_t \mid \mathcal{F}_t)(\omega) \in \mathcal{P}(\mathcal{Y}).$$

So the fundamental object is a conditional measure-valued kernel, not a single forecast trajectory.

Storage

Storage is a compression map

$$R : \mathcal{P}(\mathcal{Y}) \to \mathcal{K}, \qquad K_t := R(\kappa_t),$$

where $\mathcal{K}$ is the state-object space used operationally.

This defines "storing future information" precisely: we store a compressed representation of the conditional future law.

7.7 Axioms for a future-information object

We require five axioms.

- **A1 (Non-anticipativity):** K_t is $\mathcal{F}_t$-measurable.
- **A2 (Dynamic closure):** there exists an update operator U_Δ such that $K_{t+\Delta} = U_\Delta(K_t, \text{new data on } [t, t+\Delta])$.
- **A3 (Stability):** bounded perturbations in input law imply bounded perturbations in stored state under a chosen metric pair on $\mathcal{P}(\mathcal{Y})$ and $\mathcal{K}$.
- **A4 (Readout compatibility):** each target observable family is produced by a readout on $\mathcal{K}$.
- **A5 (Residual honesty):** every readout includes an explicit residual/error term; no hidden exactness claims.

These axioms are model-agnostic and application-agnostic.

7.8 Exact and approximate direct usability

Let $\mathcal{H}$ be a payoff/functional class on future paths.

Exact direct usability for $\mathcal{H}$

The storage is exact for $\mathcal{H}$ if for every $h \in \mathcal{H}$ there exists $\phi_h : \mathcal{K} \to \mathbb{R}$ such that

$$\mathbb{E}^{\mathbb{Q}}[h(Y_t) \mid \mathcal{F}_t] = \phi_h(K_t) \quad \text{a.s.}$$

Approximate direct usability for $\mathcal{H}$

The storage is $\varepsilon_{\mathcal{H}}$-usable if there exist readouts ϕ_h with

$$\sup_{h \in \mathcal{H}} \left\| \mathbb{E}^{\mathbb{Q}}[h(Y_t) \mid \mathcal{F}_t] - \phi_h(K_t) \right\|_{L^p} \le \varepsilon_{\mathcal{H}}(t).$$

This is the formal answer to "can we use the stored object directly for pricing?" Yes, exactly for classes where exact sufficiency holds, and approximately where the residual is controlled.

---

8. Application Program (Event-Native Pilots)

The immediate value of the resonator extension is not only conceptual. It creates a practical route from point-like arrivals to decision-ready state updates.

This section defines a bounded application program with concrete benchmark surfaces.

8.1 Pilot A: Event-native market risk updates

Use case:

- asynchronous market-impact arrivals (news shocks, liquidity spikes, micro-burst volatility events),

- with memory-aware risk updates instead of rolling-window-only updates.

Baseline:

- EWMA and GARCH updates on fixed windows.

Spectral-state variant:

- ingest sparse event stream into dynamic mode state $\psi_k^{\mathrm{dyn}}(t)$,
- compute risk readouts from the updated mode state.

Primary KPI set:

- breach-rate reduction at fixed confidence level,
- earlier regime-shift detection (lead time),
- lower false-positive rate under bursty noise.

8.2 Pilot B: Event-native alert triage

Use case:

- sparse security/ops alert streams where incidents arrive as bursts.

Baseline:

- threshold + rule-engine triage.

Spectral-state variant:

- convert burst arrivals into mode footprints,
- rank alerts by state-level confidence and persistence, not by point intensity alone.

Primary KPI set:

- precision at top-k triage budget,
- incident recall under fixed analyst capacity,
- alert fatigue reduction (false-alert load per day).

8.3 Pilot C: Biomedical sparse-signal progression

Use case:

- sparse biomarker events and intervention-time observations.

Baseline:

- static or short-window feature aggregation.

Spectral-state variant:

- event-to-state ingestion with mode-dependent retention,
- uncertainty-aware progression scoring.

Primary KPI set:

- calibration (ECE / Brier),
- early-warning lead time,
- ranking quality ($\mathrm{hit}-\mathrm{rate}@\mathrm{k}$) under bounded intervention budget.

8.4 Why these pilots are the right first step

All three pilots share the same structural stressor: information arrives as sparse, asynchronous events.

That is exactly where the resonator extension should outperform static coefficient-only pipelines:

- it preserves event footprint in mode space,
- separates short-lived and persistent components,
- and exposes uncertainty jointly with signal strength.

This keeps the paper's theory-to-practice bridge honest: the claim is not universal superiority, but improved decision quality in event-native regimes.

8.5 Pilot D: Clean mathematical benchmark (controlled SDE/PDE)

To isolate mechanism quality before domain complexity, we define a publication-oriented clean benchmark.

System class:

- controlled toy dynamics with known ground truth and impulse forcing,
- one SDE track and one linear PDE track, both with synthetic reproducible generators.

Suggested canonical pair:

1. **SDE track (Ornstein-Uhlenbeck with impulse kicks)**

$$dX_t = -\theta X_t \, dt + \sigma \, dW_t + \sum_i a_i \delta(t - t_i) \, dt$$

2. **PDE track (forced diffusion / advection-diffusion)**

$$\partial_t u = \kappa \Delta u + f(t, x), \quad f(t, x) = \sum_i a_i \delta(t - t_i) \, g_i(x)$$

Benchmark protocol:

- generate impulse-forced trajectories with fixed random seeds,
- run baseline estimators and resonator-state estimators on exactly the same data,
- evaluate against known latent state and known forcing schedule.

Baselines:

- windowed moving-average / EWMA state update,
- fixed-basis projection without resonator memory dynamics.

Spectral-resonator variant:

- event spread -> spectral projection -> damped resonator update,
- read out both signal estimate and uncertainty from $\psi_k^{\text{dyn}}(t)$.

Primary metrics (pre-registered):

- state reconstruction error (RMSE to latent true state),
- event detection lead time,
- false-positive event rate,

- calibration score (ECE and Brier) for event/no-event confidence.

8.6 Proof-target package for Pilot D

Pilot D is designed to support both theorem-level and empirical claims.

Minimal theorem targets:

1. **BIBO stability of resonator update** under $\rho(A) < 1$.

2. **Mode-energy decay bound** for unforced dynamics.

3. **Impulse-to-state gain bound** (bounded event mass implies bounded state-energy increment).

Minimal empirical targets:

- statistically significant RMSE improvement over baseline on both SDE and PDE tracks,
- lower false-positive rate at matched recall,
- reproducible win across at least three seed families.

This is the first benchmark in the program that can support a strong "provable + measurable" claim without relying on domain-specific data idiosyncrasies.

8.7 Path-dependency abstraction (general extension)

A key extension is to treat path-dependent information as a first-class state object rather than as ad hoc feature engineering.

Let the raw trajectory be X_[0,t]. Define a path-information abstraction operator

$$\Pi_t = \mathcal{A}(X_{[0,t]}),$$

where Π_t is a compressed path state (finite-dimensional in implementation, functional in theory).

In the resonator setting, this can be implemented as a memory-kernel lift:

$$\Pi_t(\tau) = \int_0^t K(t-s,\tau)\, dX_s,$$

followed by spectral projection:

$$p_k(t) = \langle \Pi_t, \varphi_k \rangle.$$

The dynamic local state then becomes

$$\tilde{\psi}_k(t) = (c_k(t), p_k(t), \nu_k(t)),$$

which jointly carries:

- current event-driven mode signal (c_k),

- path-memory contribution (p_k),
- mode-level uncertainty (nu_k).

Interpretation:

- non-path models are recovered when the kernel is degenerate (K collapses to instantaneous readout),
- path-dependent systems are represented without leaving the same spectral-state ontology.

Minimal theorem targets for the abstraction layer:

1. **Well-posedness of path abstraction** under bounded-variation or Itô-integrable trajectories.
2. **Stability of lifted path state** under bounded kernel norm and resonator damping.
3. **Approximation bound**: finite-mode truncation error for path state readouts.

This gives a direct bridge to generic path-dependency definitions while keeping the benchmark provable and empirically testable.

This abstraction is aligned with the Spectral Time taxonomy (direct / augmentable / impractical / hybrid): stable finite-dimensional closure corresponds to the augmentable class, while non-closure pushes the same state ontology into hybrid propagation.

8.8 Concrete choice of the abstraction operator

For the first executable benchmark release, we fix one concrete abstraction family.

8.8.1 Exponential-memory kernel lift

Choose a nonnegative exponential kernel bank

$$K_j(t-s) = \alpha_j e^{-\beta_j (t-s)} \mathbf{1}_{s \le t}, \quad \alpha_j > 0,\ \beta_j > 0,\ j = 1, \dots, m.$$

Define lifted path coordinates

$$q_j(t) = \int_0^t K_j(t-s)\, dX_s.$$

These coordinates satisfy a closed ODE/SDE recursion in many settings:

$$dq_j(t) = -\beta_j q_j(t)\, dt + \alpha_j\, dX_t,$$

which gives an implementation-friendly path memory state without storing full trajectory history.

8.8.2 Spectral projection family

Project the lifted path state into a finite orthonormal basis $\{\varphi_k\}_{k=1}^r$:

$$p_k(t) = \langle q(t, \cdot), \varphi_k \rangle.$$

Recommended basis order for the clean benchmark:

1. Laguerre basis (default for exponential-memory compatibility),
2. cosine basis (cross-check baseline),
3. wavelet packet basis (optional stress test for burst localization).

8.8.3 Discrete-time implementable update

With step size Δt, a first-order update is:

$$q_j^{n+1} = e^{-\beta_j \Delta t} q_j^n + \alpha_j \Delta X_n,$$

then project to p_k^{n+1} and combine with resonator state update.

The full benchmark state becomes

$$\tilde{\psi}_k^n = (c_k^n, p_k^n, \nu_k^n).$$

8.8.4 Parameter defaults for reproducible v1

Use fixed defaults for the initial benchmark:

- memory bank size: m = 8,
- decay grid: beta_j log-spaced on [0.1, 10],
- weights: alpha_j = 1/m,
- spectral projection rank: r = 6,
- step size: Delta t = 0.01 on the normalized horizon.

These defaults are not presented as optimal. They are a reproducibility contract for v1.

8.8.5 Immediate theorem-facing consequences

Under the fixed exponential-memory family:

1. lifted path coordinates have explicit stability factor e^{−beta_j Delta t},
2. path state recursion is finite-dimensional and Markov in (X_t, q_t),
3. truncation to rank r has an explicit residual-energy reportable term.

This is the smallest concrete operator choice that is both mathematically legible and directly executable.

8.9 Universal future information object and manifestation map

Building on the abstract definitions in Sections 7.6-7.8, the path-abstraction layer suggests a stronger organizing object: not "the path itself," but a compressed future-facing information object.

Define

$$\mathcal{K}_t = (\Xi_t, \mathcal{R}_t, \mathcal{E}_t),$$

where:

- Ξ_t is the compressed state representing the conditional future law,

- $\mathcal{R}_t$ is a family of readout operators,
- $\mathcal{E}_t$ is an explicit residual/error ledger.

The key operational rule is:

$$\text{manifestation}_j(t) = \Phi_j(\mathcal{K}_t),$$

so every downstream quantity is a readout from the same object.

8.9.1 Why PDF is only one manifestation

In this language, the PDF is one valid readout:

$$f_{t,T}(x) = \Phi_{\text{pdf}}(\mathcal{K}_t).$$

But many other quantities are equally first-class manifestations:

- CDF and quantiles,
- VaR / ES and tail probabilities,
- barrier hit probabilities,
- occupation-time statistics,
- conditional transition moments,
- option prices and Greeks for multiple payoff families.

So the object is primary; the PDF is one projection of it.

8.9.2 Direct pricing condition

For a payoff class $\mathcal{F}$, direct pricing from the object is valid when there exists a readout map Φ_F and a residual bound $\epsilon_F(t)$ such that

$$\left|\mathbb{E}^{\mathbb{Q}}\left[F(S_{[t,T]}) \mid \mathcal{F}_t\right] - \Phi_F(\mathcal{K}_t)\right| \le \epsilon_F(t).$$

Exact sufficiency corresponds to $\epsilon_F(t) = 0$. Approximate sufficiency is the practical case and is acceptable when the residual is tracked and controlled.

This gives a formal criterion for "can we price directly from the information object?"

8.9.3 Universality tiers

The object should be judged by tier, not by a binary yes/no universality claim.

- **Tier 1 (exact class-specific):** exact finite-dimensional state for restricted model/payoff classes.
- **Tier 2 (epsilon-sufficient broad class):** one object supports many readouts with explicit residual bounds.
- **Tier 3 (decision-grade universal interface):** one object drives pricing/risk/routing consistently with calibrated residuals.

The paper's realistic target is Tier 2 with a path to Tier 3.

8.9.4 Residual ledger as part of the object

To avoid fake universality, residuals are not optional diagnostics; they are part of $\mathcal{K}_t$.

For each readout family we maintain:

- approximation residual,
- calibration residual,
- stress residual under distribution shift.

This is why the object is usable in production-style modeling: it carries both what we know and how wrong we may be.

8.10 Spectral Routing v1 (surface-first combinatorial optimization)

The same state-object philosophy extends to route optimization problems such as Euclidean TSP-style tasks.

Given points $\{x_i\}_{i=1}^n$ and edge costs c_{ij}, the exact routing problem is combinatorial:

$$\min_{\pi \in \mathfrak{S}_n} \sum_{k=1}^{n-1} c_{\pi_k, \pi_{k+1}} + c_{\pi_n, \pi_1}.$$

The spectral route is to first construct a geometry-aware graph object, solve a smooth surrogate in an embedded space, then decode to a discrete route.

8.10.1 Graph object and spectral embedding

Construct

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right), \quad D_{ii} = \sum_j W_{ij}, \quad L = D - W.$$

Let $U_r \in \mathbb{R}^{n \times r}$ contain the first nontrivial eigenvectors of L. Each node obtains embedded coordinates

$$y_i = (U_{i1}, \dots, U_{ir}).$$

This gives a low-dimensional surface where neighborhood structure is smoother than in the raw combinatorial space.

8.10.2 Surface-first route construction

Define a continuous ordering surrogate on embedded points (for example, smooth closed-curve fitting or Hamiltonian-cycle relaxations in the embedded geometry), produce an initial permutation $\pi^{(0)}$, then apply a bounded local search (2-opt or Lin-Kernighan step budget) to obtain $\pi^{(*)}$.

The key claim is not exact NP-hard collapse; it is high-quality initialization plus stable refinement.

8.10.3 Shared-object interpretation

For routing, define a route-information object

$$\mathcal{K}^{\text{route}} = (\Xi^{\text{route}}, \mathcal{R}^{\text{route}}, \mathcal{E}^{\text{route}}),$$

where:

- Ξ^{route} stores graph-spectral state (Laplacian spectrum, embedding coordinates, local density),
- $\mathcal{R}^{\text{route}}$ stores readouts (initial tour, refined tour, lower-bound proxy),
- $\mathcal{E}^{\text{route}}$ stores residuals (optimality gap estimate, perturbation sensitivity).

This is the routing analogue of the future-information object: one object, multiple manifestations.

8.10.4 Theorem targets for Spectral Routing v1

1. **Embedding stability bound** under bounded edge-weight perturbation (Davis-Kahan type angle control).
2. **Decode distortion bound** linking embedded-neighbor consistency to route-length inflation.
3. **Refinement monotonicity** for bounded 2-opt loop (non-increasing route cost with finite termination).

These targets are intentionally modest and verifiable; they support a publishable "provable + measurable" position without overclaiming global optimality.

8.10.5 Benchmark matrix (pre-registered)

Case class	Sizes	Baselines	Metrics
Euclidean random TSP	$n = \{20, 50, 100, 200\}$	nearest-neighbor, random+2-opt, Christofides (metric)	optimality gap vs exact solver on feasible sizes, runtime, variance across seeds
Clustered geometric TSP	$n = \{50, 100, 200\}$	same	gap under cluster imbalance, runtime scaling
Noisy-coordinate TSP	$n = \{50, 100\}$	same	robustness of tour quality under perturbation

Acceptance targets for v1:

- mean gap improvement over nearest-neighbor and random+2-opt,
- competitive runtime at $n \geq 100$,
- lower seed-to-seed variance than non-spectral initializers.

8.11 Nested expectation depth collapse (XVA-style structure)

Many practical pricing/risk systems have nested conditional expectations. A canonical abstraction is:

$$V = b + T(V),$$

where one application of T corresponds to one additional nesting level.

If T is a contraction on the chosen normed space,

$$\|T\| < 1,$$

then the fixed point exists uniquely and admits a Neumann expansion:

$$V = (I - T)^{-1} b = \sum_{n=0}^{\infty} T^n b.$$

The level-L truncation

$$V^{(L)} := \sum_{n=0}^{L} T^n b$$

has explicit residual bound:

$$\|V - V^{(L)}\| \le \frac{\|T\|^{L+1}}{1 - \|T\|} \|b\|.$$

So "infinite nesting" is mathematically valid, but the effective computational depth is finite once the target error budget is fixed.

8.11.1 Spectral mode view

If $T\phi_k = \lambda_k \phi_k$, then mode-wise:

$$V_k = \frac{b_k}{1 - \lambda_k}, \qquad |R_{k,L}| \le \frac{|b_k|\,|\lambda_k|^{L+1}}{1 - |\lambda_k|} \quad (|\lambda_k| < 1).$$

This makes depth planning explicit:

- modes with small $|\lambda_k|$ need few levels,
- modes with $|\lambda_k| \approx 1$ dominate depth cost,
- modes with $|\lambda_k| \ge 1$ violate the contraction regime and require model/operator repair.

8.11.2 Practical stop rule

For a target tolerance ε, stop when either:

$$\frac{\|T\|^{L+1}}{1 - \|T\|} \|b\| \le \varepsilon$$

or an empirical residual proxy stays below tolerance over a stability window.

This subsection gives a formal bridge between nested Monte Carlo intuition and the spectral-state program: depth is not a heuristic constant; it is a spectrum-controlled quantity with explicit residual accounting.

8.12 Notation alignment across manuscripts

To keep meta_theory_spectral_information_state and fin_spectral_time consistent, we fix the following shared notation layer.

Symbol	Canonical meaning	fin_spectral_time instantiation
$\mathcal{K}_t$	primary stored information object	$\mathcal{K}_t^{\mathrm{ST}} = (M_t^{\mathrm{aug}}, A_t, \mathcal{R}_t, \mathcal{E}_t)$
Ξ_t	compressed conditional-future state	represented by generator + coefficients
$\mathcal{R}_t$	readout family	pricing, first-passage, stopping, subordination readouts
$\mathcal{E}_t$	residual/error ledger	approximation, calibration, stress residuals
T	nesting/readout propagation operator	continuation or nested expectation operator
λ_k^M	eigenvalue of generator-side operator	intrinsic temporal mode scale in fin_spectral_time
λ_k^T	eigenvalue of nesting operator T	depth-decay control in nested stacks

Two practical conventions:

1. Use λ_k^M for generator-time spectra and λ_k^T for nesting-depth spectra to avoid overload.
2. Treat PDF/CDF, VaR/ES, first passage, and route/proxy outputs uniformly as manifestations $\Phi_j(\mathcal{K}_t)$ from one object.

8.13 Spectral Memory Optimizer (global-local bridge)

A direct extension of the state-object idea is an optimizer that carries an explicit spectral memory of landscape structure while iterating.

8.13.1 Core optimizer state

Define a spectral optimizer memory object

$$\mathcal{M}_t = (U_t, \Lambda_t, a_t, \Sigma_t, \mathcal{E}_t),$$

where:

- U_t are learned principal mode directions of local landscape variation,
- Λ_t are mode-scale curvature proxies,
- a_t are mode amplitudes (state projection strengths),

- Σ_t is mode-level uncertainty,
- $\mathcal{E}_t$ is a residual ledger (model mismatch, stale-spectrum risk, update error).

This is a structural prior over optimization geometry, not just a prior over parameters.

8.13.2 Online update and step rule

Given observation stream $o_t = (f_t, g_t, \text{optional } Hv_t)$, update

$$p(\mathcal{M}_{t+1} \mid o_{1:t}) \propto p(o_t \mid \mathcal{M}_t)\, p(\mathcal{M}_t \mid o_{1:t-1}).$$

Use a hybrid descent-explore step:

$$\Delta\theta_t = -\eta\, U_t(\Lambda_t + \tau I)^{-1} U_t^\top g_t + \beta\, U_t \Sigma_t^{1/2} \epsilon_t.$$

Interpretation:

- first term = geometry-aware local descent in learned spectral coordinates,
- second term = uncertainty-directed exploration to reduce local-trap lock-in.

8.13.3 Why this is beyond classical priors

Classical priors typically constrain parameter distributions. The spectral memory prior constrains and updates a latent model of landscape structure:

- where stable descent directions are,
- where curvature is stiff or flat,
- where uncertainty is high enough to justify exploratory motion.

So the optimizer updates its world-model of the objective while optimizing.

8.13.4 Theorem targets for v1

1. **Contraction-region convergence** under bounded spectral-estimation error on locally strongly convex regions.
2. **Approximation-robust convergence bound** with explicit penalty in terms of spectral model error.
3. **Strict-saddle escape probability improvement** under uncertainty-guided spectral exploration.

These statements are intentionally scoped to avoid overclaiming global nonconvex optimality.

8.13.5 Benchmark surface

Compare against Adam / L-BFGS / CMA-ES on controlled nonconvex suites:

- multimodal synthetic objectives,
- saddle-heavy quartic constructions,
- noisy objective variants.

Primary metrics:

- best objective value vs evaluation budget,
- seed-to-seed variance,
- time-to-escape from saddle neighborhoods,
- local-minimum lock-in rate.

8.14 Session-derived thought ledger (non-loss consolidation)

To ensure no high-value ideation is lost from live discussion, we keep this compact ledger inside the paper-level theory object.

Source session export used for this consolidation: .cursor/state/session−exports/cursor_infor_state.md.

Session idea	Current status	Canonical section	Next bounded action
Point events should be stored as spectral memory footprints, not raw impulses	formalized	7.1-7.3	prove impulse-to-state gain bound in clean benchmark lane
The same state object should be reusable across risk, detection, routing, and optimization tasks	formalized (cross-surface)	3, 8.9, 8.10, 8.13	add one shared-manifestation benchmark table proving one-object multi-readout reuse
Future information should be represented as an object; PDF is only one manifestation	formalized	8.9	implement multi-manifestation report (PDF/CDF/VaR−ES/barrier proxy) from one object in A-04
Core definitions should be application-free before benchmarks (future, information, storage)	formalized	7.6-7.8	keep theorem program and benchmark claims explicitly separated
Direct pricing should be conditioned on object sufficiency with explicit residual	formalized	8.9.2, 7.8	add per-payoff residual ledger outputs in benchmark JSON
Path-dependency should be abstracted by a kernel-lifted state, not ad hoc features	formalized	8.7-8.8	test basis sensitivity (Laguerre/cosine/wavelet) under fixed synthetic data
The "stored future" should be conditional-law information, not a claimed realized future path	formalized	7.6-7.8, 8.9	add one explicit experiment that reports sufficiency residual by payoff family

Session idea	Current status	Canonical section	Next bounded action
Infinite nesting is theoretical; effective depth is spectrum-controlled	formalized	8.11	add auto depth-stop rule from contraction estimate in prototype runners
Surface-first spectral routing may reduce combinatorial search burden	formalized + runnable v1	8.10	run full matrix (n=20,50,100,200, multi-seed) and log gap/runtime/variance
Optimizer can carry a spectral world-model prior and update online	formalized (program)	8.13	implement A-06 minimal optimizer prototype and compare vs Adam/L-BFGS/CMA-ES
Candidate physical domains (earthquake, grid transients, turbulence) as event-native tests	partially captured	8.1-8.5	promote one physical-domain benchmark after clean SDE/PDE pass stabilizes

This ledger is intentionally compact: it preserves idea identity, current maturity, and the smallest next executable step.

Non-loss policy for this paper line:

- if a harvested session idea is valuable but not central to the current manuscript arc, it is moved to an explicit preserved surface (companion section, task lane, or extracted-material bank), never silently dropped.

8.15 Physical benchmark frontier harvested from session ideation

The session also proposed three concrete physical benchmark families that should be preserved as bounded next-lane candidates after the clean SDE/PDE benchmark stabilizes.

1. **Seismic micro-event precursor detection**
   - Baseline: trigger-centric STA/LTA or fixed-threshold wavelet detectors.
   - Spectral-memory hypothesis: micro-impulses create stable mode-footprints that improve early warning signal quality.
   - KPIs: lead time, false alarm rate, event-window precision/recall.
2. **Power-grid transient fault handling**
   - Baseline: relay thresholds and snapshot FFT classification.
   - Spectral-memory hypothesis: impulse transients separate into persistent structural vs short disturbance modes.
   - KPIs: fault classification latency, false trip rate, recovery decision time.
3. **Turbulence/vortex burst tracking**
   - Baseline: windowed PSD/energy thresholding.
   - Spectral-memory hypothesis: burst events admit cleaner mode-trajectory tracking with explicit decay controls.

- KPIs: event localization error, detection delay, false positive burden.

All three lanes inherit the same residual-honesty policy of this manuscript: report explicit object-to-readout residuals, not only task-level scores.

8.16 Lane freeze note: quant-finance readout mapping and current optimizer status

This note records the current lane state at freeze time, to avoid loss of practical conclusions.

8.16.1 Quant-finance use mapping from one stored object

Under the universal object interface

$$\mathcal{K}_t = (\Xi_t, \mathcal{R}_t, \mathcal{E}_t),$$

the same stored state supports multiple quant readouts:

- **Density layer**: PDF/CDF as manifestations $\Phi_j(\mathcal{K}_t)$.
- **Risk layer**: VaR/ES, tail probabilities, barrier-hit likelihoods.
- **Pricing layer**: payoff-family pricing readouts from the same object, with residual reporting.
- **Sensitivity layer**: local sensitivity/Greek-like approximations as readouts, with explicit error ledger.
- **Nested layer**: continuation and nested-expectation proxies with spectrum-guided depth accounting.

So PDF is one valid manifestation, but not the only target surface.

8.16.2 Spectral memory optimizer lane status (A-06 snapshot)

The optimizer extension has been implemented and benchmarked as a bounded pilot:

- baseline lanes: Adam, L-BFGS, basic CMA-ES;
- spectral lane: online spectral-memory update with residual-honesty tracking;
- added stabilization: trust-region clipping + backtracking line-search.

Empirical snapshot at freeze:

- stabilization materially improved spectral-lane robustness (lower blow-up behavior and lower seed variance than prior unstabilized version),
- but promotion gate versus strong baselines is not yet passed on the expanded matrix.

Current interpretation:

1. The methodology is operational and reproducible as an evaluation framework.
2. The spectral lane is promising as a hybrid warm-start/preconditioner component.
3. A “baseline-replacing optimizer” claim is not yet justified and remains out of scope.

8.16.3 Re-open conditions for this lane

This lane should be re-opened for headline claims only when all hold:

- stable wins on at least two objective families under fixed budget and multi-seed CI reporting,

- no variance explosion relative to reference baselines,
- at least one task-grounded quant lane (calibration/pricing-risk loop) showing practical gain.

---

9. Scope and Limitations

The spectral information state has a real scope boundary.

It requires a meaningful spectral lens. If the chosen basis is poor, unstable, or unrelated to the real latent structure, the state will inherit that weakness.

It is also not a full substitute for causal structure. The state says how much local information is present in a mode, not whether the mode is causally interpretable or policy-invariant.

Finally, the strongest exact sufficiency claims belong to Gaussian linear settings. In broader nonlinear or heavy-tailed settings, the state should be treated as the canonical local inferential object rather than as a universally sufficient statistic in the strict classical sense.

These are limitations of scope, not reasons to avoid the object. They simply mark where the state should be used precisely and where it should be paired with additional structure.

---

10. Conclusion

The main claim of this paper is simple: in spectral problems, the right primitive inferential object is not a posterior alone, not a confidence interval alone, and not a model-size penalty alone. It is the mode-level pair $(\widehat{A}_k, \sigma_k^2)$.

We call this pair the **spectral information state** because it records what the data currently knows about a mode, how uncertain that knowledge remains, and what inferential and decision-theoretic readouts are justified from that point.

Once this object is made explicit, the surrounding landscape becomes cleaner. Representation theorems explain why the spectral basis exists. State theory explains how knowledge is stored locally in that basis. Bridge papers then show why apparently competing inferential traditions often agree once they act on the same state.

That is the intended role of this paper: to make the state itself visible.