Discretizing Continuous OHLC into Recursive Swing “Game Pieces” (OS1)

Main proposal

Problem statement (North Star aligned)

We need a discretization of continuous OHLC into **swing-level game pieces** that a recursive generator can manipulate to produce **1-minute OHLC that looks real** while obeying the project’s structural laws (Fibonacci attractors, 2× completion, scale-aware invalidation, decision zones/liquidity voids, and top-down causality). This discretization is the bridge between:

- **Analysis** (what the repo already does: multi-scale swings + Fib levels + structural events), and
- **Generation** (the North Star: synthesize OHLC with the same laws, not a black-box imitation).

Per `.claude/why.md`, **trust is existential**: the representation must be interpretable and replayable so every generated bar can be traced to explicit, auditable decisions before real capital ever touches it.

Success criteria (what “good discretization” means)

1. **Interpretability is first-class**
   - Every discrete transition is explainable in one sentence in market-structure language (“L scale tested 1.5 four times and triggered frustration → symmetric retrace to 0.5”).
   - No “latent state” is required to understand why an outcome happened.
2. **Structural correctness is enforced, not hoped for**
   - Completion, invalidation thresholds, allowed transitions, and top-down causality are hard invariants, with a single explicit escape hatch (“tail override”) that is always logged.
3. **Data efficiency / anti-overfitting**
   - The representation is learnable/tunable from **small data** (expert swing annotations + counts from OHLC), using strong priors and maximum-entropy defaults.
   - Parameters are few, human-meaningful, and stable across regimes; avoid large conditional tables that memorize one dataset.
4. **Stochastic richness without structural drift**
   - Diversity comes from sampling *among structurally valid options* (targets, durations, path shapes) and from a controlled tail channel—not from breaking rules.
5. **Replayable canonical truth**
   - There is a single canonical “game record” log that deterministically regenerates OHLC given seeds (OHLC is a rendering of decisions, not

the decisions themselves).

The recommended discretization: HSMLM + explicit decision dynamics

One canonical coordinate system (bull and bear unify)

Define an **oriented swing frame** where Fibonacci “ratio” always increases in the *expected move direction*.

ReferenceFrame

- For a bull reference: `anchor0 = low`, `anchor1 = high`
- For a bear reference: `anchor0 = high`, `anchor1 = low`

Then:

```
range = anchor1 - anchor0              # sign encodes direction
ratio(p) = (p - anchor0) / range       # works for bull and bear
price(r) = anchor0 + r * range
```

Interpretation: - `ratio == 0` is the defended pivot (low for bull, high for bear). - `ratio == 1` is the origin extremum (high for bull, low for bear). - `ratio == 2` is the 2× completion target. - Negative ratios are “beyond the defended pivot” (stop-run territory).

This gives us a universal notion of “where price is” relative to structure, without separate bull/bear logic in the discrete layer.

Discrete “board”: named levels and bands

Use a small, fixed set of ratios as **named boundaries**; the discrete state is the **band** (interval) between adjacent boundaries, not the raw ratio.

Phase 1 level set (match the North Star rules, keep it small):

- `STOP = -0.10` (explicit stop-run depth; invalidation threshold is scale-specific)
- `0.00`, `0.10`, `0.382`, `0.50`, `0.618`
- `0.90`, `1.00`, `1.10`
- `1.382`, `1.50`, `1.618`
- `2.00` (completion)

This level set is intentionally *structural*: it encodes the decision-zone / void geometry in `Docs/Reference/product_north_star.md` without needing pattern libraries on day one.

Bands are the adjacent intervals (e.g., `1.382-1.50`, `1.50-1.618`, `1.618-2.00`, `0.10-0.382`, etc.).

Scale-dependent termination thresholds (explicit)

Completion/invalidation semantics are hard constraints, but the thresholds vary by scale (mirrors the repo's documented validation rules).

Scale	Completion (bull/bear)		Invalidation (trade)	Invalidation (close)
S / M	`ratio(close)`	`2.0`	`ratio(wick) < 0.0`	`ratio(close) < 0.0`
L / XL	`ratio(close)`	`2.0`	`ratio(wick) -0.15`	`ratio(close) -0.10`

Notes: - `STOP = -0.10` is a named behavioral boundary ("stop run" depth). On S/M it sits *past* invalidation; recoveries still exist but are represented as a new `Reanchor` after invalidation (not as "quiet violations"). - "Wick" vs "close" is evaluated at the swing's aggregation level, not raw 1-minute bars.

Atomic "game piece" (what we discretize into)

The atomic game piece is a **BandTransition** at a given scale:

> "At scale M, price moved from band `1.00-1.10` to band `1.10-1.382` over 47 minutes, under parent context X, with rationale Y."

This is the smallest unit that: - maps cleanly to OHLC reconstruction (entry/exit prices are defined), - supports recursion (lower scales fill the interior), - remains interpretable (it is literally "which structural boundary did we cross?").

State (what exists on the board)

Model each scale as a **Hierarchical Semi-Markov Level Machine (HSMLM)**: a semi-Markov process over bands, with explicit counters for second-order rules.

Per-scale state (`ScaleState`)

- `scale  {S, M, L, XL}`
- `frame: ReferenceFrame` (anchor0/anchor1 + derived prices)
- `band_id` (current band)
- `dwell_bars` (semi-Markov time spent in current band)
- `impulse_ema` (distance / time proxy; drives volatility clustering)
- `attempts[level_name]` (failed tests near key levels; fuels frustration)
- `frustrated[level_name]` (boolean flags)
- `exhausted` (set after completion; forces pullback expectations at top scale)
- `target_stack_pressure` (scalar; tracks "too many targets" rule)
- `tail_mode` (rare override active? if yes, must be logged)
- `parent_context` (derived inputs from the parent scale; see recursion)

Global state (`GameState`)

- `t` (time in 1-minute bars)
- `scales: {XL, L, M, S} → ScaleState`
- `news_stream` (optional, exogenous events; see assumptions)

This is intentionally *small* and *auditable*: it's the minimum that lets us implement decision zones, liquidity voids, frustration/measured-move/exhaustion, and volatility clustering without hidden variables.

Actions / moves (what changes state)

At the discrete layer, there are only three primitives:

1. `BandTransition`
    - `(scale, from_band, to_band, duration_bars, rationale, rng_seed)`
    - Default constraint: transitions are to **adjacent bands** only.
2. `StructuralEvent`
    - `(scale, event_type, level_name, metadata)`
    - Types: `LEVEL_TEST`, `FRUSTRATION`, `MEASURED_MOVE_TRIGGER`, `COMPLETION`, `INVALIDATION`, `TARGET_STACK_RELEASE`, `TAIL_OVERRIDE`.
3. `Reanchor`
    - `(scale, new_frame, reason)` after completion/invalidation or explicit model rules.

What ends an episode

Define episodes in two ways (both useful; choose per experiment):

- **Structural episode (recommended):** one full lifecycle of the **top available scale frame** in the window (ideally XL): begins at `Reanchor`, ends at `COMPLETION` or `INVALIDATION` on that scale.
- **Fixed-horizon episode (practical for simulation runs):** generate for `N` minutes; any active frames persist across horizons in the log.

In both cases, lower scales terminate and reanchor multiple times within a higher-scale episode.

Recursion across scales (how "big moves drive small moves" becomes code)

Top-down causality is enforced by **parent-conditioned sampling**, not by letting children "vote" a parent into existence.

For each child scale `k`, compute a parent context `C_{k+1→k}` including:

- parent band and zone type (void vs decision zone),
- parent pending direction (toward higher ratio vs lower),
- parent distance to next key boundary / completion,

- parent exhaustion/frustration constraints,
- parent "allowed play" envelope (soft bounds for child exploration).

Then sample child transitions as:

```
P(child_next | child_state, C_parent)  base(child_band, zone_type) × gate(parent_direction,
```

Hard rule: child behavior may only change parent state via explicit logged structural events (completion/invalidation) or an explicit logged tail override. This makes causal direction auditable.

Canonical truth: the "game record" (what is real)

The canonical truth is an append-only log of:

- initial frames per scale,
- exogenous news events (optional),
- band transitions + structural events + reanchors,
- RNG seeds for deterministic replay.

OHLC is a deterministic rendering of this log (given seeds + renderer version).

Minimal schema sketch

```json
{
  "meta": {"instrument":"ES", "tick":0.25, "seed":123, "level_set":"os1-v1"},
  "initial": {
    "XL": {"anchor0": 5000.0, "anchor1": 5100.0, "as_of_t": 0},
    "L":  {"anchor0": 5030.0, "anchor1": 5090.0, "as_of_t": 0}
  },
  "news": [{"t": 1280, "polarity": -1, "intensity": 0.7, "ttl": 60}],
  "log": [
    {"id":"m1","t": 0, "type":"move", "scale":"L", "from":"1.00-1.10", "to":"1.10-1.382", "b
    {"id":"e1","t":47, "type":"event","scale":"L", "event":"LEVEL_TEST", "level":"1.382", "a
    {"id":"m2","t":47, "type":"move", "scale":"L", "from":"1.10-1.382", "to":"1.00-1.10", "b
    {"id":"e2","t":80, "type":"event","scale":"L", "event":"FRUSTRATION", "level":"1.382", "
    {"id":"m3","t":80, "type":"move", "scale":"L", "from":"1.00-1.10", "to":"0.50-0.618", "b
    {"id":"m4","t":80, "type":"move", "scale":"M", "parent":"m3", "from":"1.00-1.10", "to":"
  ]
}
```

This log is the "game record" we will calibrate against and debug against. If the generator produces nonsense, the log tells us exactly which rule/path did it.

Avoiding overfitting while enabling stochastic richness

The anti-overfitting posture

1. **Fixed representation; learn only parameters**

- The level set, invariants, event types, and recursion wiring are fixed by the North Star.
- What we estimate from data: a small number of transition weights, duration distributions, and modifier strengths.

2. **Maximum entropy defaults**
   - Where data is sparse, prefer the maximum-entropy distribution consistent with the constraints (no bespoke conditionals until falsified by evidence).
3. **Strong priors + partial pooling**
   - Use Dirichlet priors for categorical transitions and Gamma/LogNormal priors for durations.
   - Pool parameters across scales where the rule is asserted to be self-similar; allow scale-specific multipliers for "smaller can be more extreme."
4. **Keep the degrees of freedom orthogonal**
   - Separate: (a) target choice, (b) duration choice, (c) rendering noise/wicks. This prevents "fixing structure" by hiding it in the renderer.

Where stochastic richness comes from (without breaking rules)

- **Choice randomness:** which adjacent band is targeted next (with interpretable weights).
- **Timing randomness:** semi-Markov durations per transition (captures chop vs snap).
- **Volatility clustering:** impulse/vol regime evolves slowly (EMA), modulating durations and wick budgets.
- **Rendering randomness:** intra-band path shape and wick placement (bounded, reversible at the structural level).
- **Tail channel:** rare, explicitly logged non-adjacent transitions / overshoots (scale-dependent heaviness).
- **News (optional):** exogenous tilts to probabilities and/or duration (accelerants), never a hidden driver.

Forward path (implementation-oriented)

Implement first (fast feedback, low regret)

1. **Lock OS1 coordinate + level set**
   - Implement `ReferenceFrame.ratio()` / `price()` and the OS1 boundary list (including `0.9/1.1/0.1/-0.1`).
   - Add invariant checks as reusable validators.
2. **Define the canonical log schema + deterministic replay contract**
   - Version the schema and renderer; enforce "same log + same seeds same OHLC."

3. **Build a forward discretizer (OHLC → game record)**
    - Input: OHLC + chosen reference frames (start with human annotations; later system-detected).
    - Output: OS1 log of band transitions + events; extract empirical transition counts and dwell distributions by zone.
    - This is the quickest falsification tool: if discretization produces messy, non-stable logs, generation will be worse.
4. **Build a minimal renderer (game record → OHLC)**
    - Start with piecewise-linear + bounded noise / wicks; ensure tick quantization.
    - Validate round-trip at the structural level: `discretize(render(log))` `log` (up to tolerance).

Defer (until the core is stable)

- Full motif library (beyond a handful of interpretable templates).
- Multiple concurrent reference frames per scale (stacking/alternates), except as a simple `target_stack_pressure` scalar.
- Learned news process; semantic news; cross-market coupling.
- Sophisticated "multi-swing rule" interactions beyond explicit events.

Fast falsification experiments (kill the approach quickly if it's wrong)

1. **Band stability test:** On annotated windows, does the forward discretizer produce a compact log where most movement is captured by band transitions and not by constant reanchoring?
2. **Zone separation test:** In real OHLC logs, do decision zones (`1.382-1.618`) show materially higher dwell/attempt counts than liquidity voids (`1.618-2.0`, `1.10-1.382`)? If not, either the level set is wrong or the event definitions are.
3. **Replay test:** Can we render → rediscretize and recover the same structural events (completions/invalidations/frustrations) with high consistency?
4. **Causality leakage test:** In generated data, measure the rate at which child net progress contradicts parent direction absent tail overrides. If it's high, recursion wiring is wrong.

"Done" for this phase (discretization, not full generation)

- OS1 logs can be extracted from real OHLC windows (with annotated frames) without lookahead and with stable event semantics.
- The log is auditable: any event or move has a human-readable explanation and an invariant justification.
- Rendering a log produces OHLC whose rediscretization recovers the same structural event sequence within tolerance.
- We can compute and track a small set of drift metrics (below) and use them as regression gates.

Risk controls (detect drift and fractal-assumption breakage early)

Hard invariant checks (must never fail) - Adjacent-band transitions only (unless `TAIL_OVERRIDE` logged). - Scale-dependent invalidation semantics enforced. - Parent can only change via explicit events.

Drift dashboards (should stay within bands) - Per-scale: transition frequencies, completion/invalidation rates, dwell distributions by zone. - Attempt/frustration rates at key levels (1.5, 1.618, 2.0, 1.382). - Volatility clustering proxies (impulse autocorrelation; asymmetry of up vs down impulses). - "Target stacking pressure" distribution vs liquidation events frequency.

Fractal consistency checks (soft, but monitored) - Similarity of normalized band-transition patterns across scales (after allowing heavier tails at smaller scales). - Scaling of "time to completion" distributions vs scale (should increase; shape should be similar).

Key uncertainties and failure modes (candid)

1. **Reference frame selection is a dependency**
    - Logs are only as good as the frames. Early phases should condition on human annotations to avoid compounding swing-detection error.
    - If frames are unstable, generation will chase noise; this is why the discretizer is the first falsification surface.
2. **Too many levels vs too few**
    - Too many boundaries increases parameters and risks overfitting; too few makes decision-zone behavior unrepresentable.
    - OS1 starts with the North Star's explicitly named "special" ratios; if that explodes complexity, the fallback is to collapse `0.9/1.0/1.1` and `0/0.1` into zones (not remove them entirely).
3. **Chop realism may require explicit attempt semantics**
    - If pure band transitions feel too "teleporty," we'll promote `LEVEL_TEST/ATTEMPT` to first-class events (still within HSMLM) rather than growing a pattern zoo.
4. **Tail modeling can become an excuse**
    - Tail overrides must remain rare and logged; if we start needing tails for everyday behavior, the core transition logic is wrong.
5. **News driver weakness**
    - Assumption: "news" is an exogenous stream of (time, polarity, intensity, TTL) that *tilts* transition and timing distributions.
    - If the news component is weak or absent, the model remains valid: internal structural tension (frustration, target stacking, exhaustion) still produces regime-like accelerations; "news" becomes just one of several modifiers rather than the sole source of motion.

Appendix 1: Document synthesis

Common threads across drafts

- **Interpretability is non-negotiable** (all drafts converge): the generator must be debuggable via explicit structural decisions, not latent vectors.
- **Fib levels are the coordinate system**: discretization should happen in ratio space relative to active swings, not in raw price space.
- **Hierarchy is causal (XL→L→M→S)**: child behavior is constrained by parent structure; "upward causation" must be explicit and rare.
- **Canonical game record + reversibility**: the discrete log should be the auditable truth; OHLC is a rendering.
- **Small-data reality**: designs that require massive training corpora are misaligned with current ground truth volume.

The main disagreements (and why they matter)

1. **Grammar-first vs state-machine-first**
   - `discretization_approaches_c1.md` pushes a pure stochastic grammar (L-system) as the whole generator.
   - `*_o2.md`, `*_c2.md`, `*_c3.md`, `*_o1.md` prefer a Fib band/level machine with stochastic transitions.
   - Practical difference: grammars are elegant for recursion but tend to drift on precise arithmetic constraints and make OHLC inversion harder; level machines align directly with "price crossed level X."
2. **Atomic unit: band transition vs intent/attempt vs motif**
   - Some drafts emphasize **level-to-level steps** (band transitions) as the primitive.
   - Others emphasize **intent/attempt loops** to capture decision-zone chop.
   - Others emphasize **motif templates** as macro-moves for realism.
   - OS1 treats band transitions as the canonical primitive and upgrades intent/attempt to explicit counters/events (not a separate architecture), with motifs reserved for rendering variety.
3. **How "hard" the Fib grid is**
   - `discretization_approaches_c2.md` explicitly frames Fib levels as attractors (soft), suggesting probabilistic boundaries.
   - `product_north_star.md` defines hard termination semantics (completion/invalidation) but soft preferences elsewhere.
   - OS1 resolves this by making *termination and legal transitions hard*, and everything else soft via weights.

Blind spots / missing pieces across drafts

- **Reference frame selection and multi-reference stacking**: most drafts assume a clean "one swing per scale," while the North Star and interview notes suggest multiple salient references and target stacking dynamics.

- **No-lookahead discipline in calibration**: several drafts gesture at "use real OHLC to learn probabilities" but don't specify how to avoid using future behavior; OS1 makes the forward discretizer (with annotated frames) the first-class, no-lookahead artifact.
- **Drift detection and fractal assumption checks**: drafts mention realism but rarely propose concrete drift metrics and invariants as regression gates.
- **Integration with the existing repo**: some proposals are architecture-level but don't leverage that the code already emits multi-scale swings and structural events; OS1 treats existing `ActiveSwing`/`StructuralEvent`-style semantics as the natural substrate.

What each draft uniquely contributes (and what it misses)

- `Docs/Proposals/Drafts/discretization_approaches_o2.md`
  - **Unique**: HSMLM framing; unified oriented ratio frame; "log is the product"; maximum-entropy stance; clear layering (core vs attempt vs motif).
  - **Misses**: A sharper operational definition of episode termination and explicit drift/falsification tests (OS1 adds these).
- `Docs/Proposals/Drafts/discretization_approaches_o1.md`
  - **Unique**: Crisp engineering sequencing and "done" checklist; pragmatic Grove-style decision discipline.
  - **Misses**: Less explicit about semi-Markov timing and second-order rule mechanisms.
- `Docs/Proposals/Drafts/discretization_approaches_c2.md`
  - **Unique**: Factorization into independently learnable components; explicit "reversibility over compression"; suggests Bayesian priors and softening where needed.
  - **Misses**: T2 ("Fib as attractors") risks undermining hard invariants unless carefully scoped; less concrete on recursion wiring and tail logging.
- `Docs/Proposals/Drafts/discretization_approaches_c3.md`
  - **Unique**: "Swing bookkeeper overlay" separation (generation vs lifecycle bookkeeping); includes a concrete implementation sketch and additional level granularity options.
  - **Misses**: More levels can expand parameters quickly; needs stronger anti-overfitting controls and clearer criteria for adding/removing boundaries.
- `Docs/Proposals/Drafts/discretization_approaches_g1.md`
  - **Unique**: The "corridor" intuition (moves define bounded paths, not straight vectors); emphasizes "struggle at levels" (attempt/reject/break) as the meaningful unit.
  - **Misses**: Less explicit about canonical log schema, replay contract, and how to keep a corridor model from becoming a hidden physics

engine.

- `Docs/Proposals/Drafts/discretization_approaches_c1.md`
  - **Unique**: Strong argument for fully traceable named rules; separates grammar structure from probability tables; clear "tune by inspection" workflow.
  - **Misses**: Pure grammar as the primary engine is high risk for precise Fib arithmetic + OHLC inversion; it can also become a large handcrafted library (a different kind of overfitting) unless tightly constrained by a canonical level machine.

How the ideas complement (and undermine) each other

- The **level/band machine** gives a shared, auditable coordinate system and makes inversion straightforward; the **intent/attempt lens** restores decision-zone realism; **motifs** add variety in rendering *after* causality is decided.
- Pure **grammar-first** approaches complement the system as a way to author higher-level "macro move" templates later, but undermine OS1 if used as the canonical representation because they blur arithmetic constraints and complicate regression testing.