

Multi-Language Replication, Formal Verification, and Mechanistic Audit of Autonomous Cell-Sorting Morphogenesis and Its “Latent Platonic Space” Claims

Bit-Exact Reproduction of Zhang, Goldstein & Levin (2025) Across Python 3, C++20, Node.js, and Lean 4, with a Three-Ablation Evaluation of *Ingressing Minds* (Levin 2025/2026)

Joel Dietz

Infinite Morphospace Research Project · <https://github.com/fractastical/morpholib>

October 22, 2026

Abstract

In a landmark study of minimal basal cognition, Zhang, Goldstein & Levin (2025, *Adaptive Behavior* 33(1):25–54) reconceptualized classical sorting algorithms (Bubble, Insertion, and Selection sort) as decentralized 1D arrays of autonomous cells that compare and swap positions locally without a central controller. They reported three striking empirical phenomena: (i) autonomous cell arrays tolerate frozen (‘broken’) cells better than traditional top-down algorithms, (ii) sorting trajectories around frozen defects exhibit *Delayed Gratification* ($DG > 0$, temporary increases in monotonicity error followed by larger reductions), and (iii) chimeric arrays mixing two algotypes spontaneously cluster by algotype (‘aggregation > 0.50 ’) even though cells cannot sense their neighbors’ algotype labels. Subsequently, Levin (2025/2026, *Ingressing Minds*, PsyArXiv 10.31234/osf.io/5g2xj_v3; *Philosophies* 11(5):161, Section 6.3 & Fig. 14) cited these sorting results as primary evidence that even transparent, deterministic algorithms exhibit unprogrammed cognitive “side-quests” that ingress from a non-physical *Latent Platonic Space of Patterns*. Conversely, several computer-science commentators dismissed the chimeric clustering results as a trivial artifact of textbook duplicate-key sort stability.

Here we resolve both sides of this debate through a four-language replication, machine-checked formal verification, and three-ablation mechanistic audit:

- 1. Bit-Exact Multi-Language Replication (Python 3, C++20, Node.js) & Line-by-Line Upstream Traceability:** Auditing the authors’ reference codebase (Zhangtaining/cell_research) we port the complete autonomous sorting architecture to native **C++20**, **JavaScript/Node.js (ES Module)**, and **Python 3** coupled to a shared 64-bit **SplitMix64** PRNG, achieving 100% bit-exact kymograph and cell-array agreement (0 cell mismatches across languages) and replicating all 14/14 published quantitative benchmarks ($n = 100$).
- 2. Five Machine-Checked Theorems in Lean 4 (0 sorry):** In `Loop_zhang_sorting_morphogenesis.lean`, we formally prove local barrier pinning (cross-swaps = 0) for Bubble/Insertion sort, stateful list-fold Insertion prefix-reset unblocking and post-frozen order checking, the $[2, 4, 1, 3] \rightarrow [2, 1, 4, 3]$ Lyapunov descent ($\Delta I = -1$) vs. monotonicity-error increase ($\Delta E = +1$) theorem, universal 3-window adjacent and non-local swap inversion descent ($\Delta I \leq -1$), and homotypic domain aggregation bounds.
- 3. Three-Ablation Mechanistic Audit of the “Latent Platonic Space” Claims:**
 - *Ablation 1 (“No error-handling code was added”):* While Bubble has no defect-handling code, Selection advances its target pointer past frozen cells and `InsertionSortCell.py` (L68–82) contains an explicit `if status == FREEZE: prev = -1` defect-aware precondition reset. Ablating those three lines causes Insertion sort’s final error ($n = 60, f = 3$) to jump $4.5\times$ ($4.90 \rightarrow 21.95$) as all downstream cells permanently halt.
 - *Ablation 2 (“Delayed Gratification / Going around a barrier against the gradient”):* In Bubble and Insertion sort, 0 cells ever cross an immovable barrier. Moreover,

on same-direction arrays, every autonomous swap (Bubble, Insertion, or Selection) performs **strict monotone descent on the global pairwise inversion Lyapunov potential** $I(v) = \#\{(i < j) : v_i > v_j\}$ (Bubble $\Delta I = -1$ on every swap; Selection $\Delta I \in [-35, -1]$; 0 uphill steps). The paper’s **monotonicity_error** $E(v) = \#\{i : v_i > v_{i+1}\}$ ticks upward (47 times, DG = 0.764) only because $E(v)$ is a coarse 1D boundary projection of $I(v)$ that increases whenever an interior swap splits one multi-step inversion into two adjacent steps (occurring even on intact arrays at $f = 0$, DG = 0.433).

- *Ablation 3 (“Spontaneous algotype clustering” vs. “Textbook stability” critiques)*: On-line critics who dismissed chimeric clustering as textbook duplicate-key stability were wrong about Paper Fig. 8A, where all n keys $1 \dots n$ are strictly unique (0 duplicate keys) yet aggregation genuinely peaks at 0.624–0.640 mid-sort. We show that unique-key clustering is a genuine **kinematic transport-speed phase-separation wave** (Selection target-pointer advection sweeping left vs. Bubble diffusion), whereas permanent duplicate-key clustering (0.634–0.692 in Fig. 8D) arises from Selection scan order packing 75% of Selection cells into the left half of each key’s plateau (S...S | B...B) plus strict ‘<’ non-swapping of equal keys (producing **anti-clustering** $A_{\text{final}} = 0.477 < 0.50$ at $C = 2$ copies per key, which directly refutes textbook sort stability even on duplicate keys).

4. **Quantifying Emergence via Hoel’s Causal Emergence 1.0 (PNAS 2013) & 2.0 (Patterns 2026)**: Constructing the exact $N \times N$ Transition Probability Matrices on $S_3 \dots S_6$ ($N = 6 \dots 720$ permutations), we prove that the Coxeter inversion potential $I(v)$ is a 100% **deterministic, causally emergent macro-scale** ($\det_I = 1.0000$, $\Delta\text{CP} = +0.1191 > 0$ on swap steps, $+0.1557$ over raw steps), whereas the paper’s adjacent monotonicity error $E(v)$ is a non-lumpable projection with **strictly negative causal emergence** ($\Delta\text{CP} = -0.2525 < 0$, $\det_E = 0.7209$).

We conclude that **Weak Structural Platonism** (“mathematical free lunches” arising from the Coxeter/Bruhat lattice of the symmetric group S_n and differential transport kinetics) is rigorously justified, whereas **Strong Dualist Ingress** (claiming these behaviors are “not predicted, explained, or produced” by the local rules) is unnecessary once the exact Lyapunov and kinematic equations are written down.

Contents

1	Introduction: Sorting Algorithms as a Minimal Model of Morphogenesis and Basal Agency	2
2	Multi-Language Replication & Line-by-Line Upstream Code Traceability	3
3	Three-Ablation Mechanistic Audit of the “Latent Platonic Space” Claims	6
3.1	Ablation 1: Did the Code Contain an Explicit Broken-Cell Exception Handler? . . .	7
3.2	Ablation 2: Does “Delayed Gratification” Move Against the Gradient or Around Barriers?	8
3.3	Ablation 3: What Drives Spontaneous Algotype Clustering (and Why Blog Critics Missed Fig. 8A)?	8
4	Quantifying Emergence: Applying Hoel’s Causal Emergence 1.0 & 2.0 to Sorting Morphogenesis	9
4.1	Positive Causal Emergence 2.0 on $I(v)$ vs. Negative Emergence on $E(v)$	10
4.2	The Intra-Plateau S...S B...B Micro-Segregation Law Refutes “Bubble-Sort Stability” Even on Duplicate Keys	11
5	Synthesis: Three New Conclusions & Weak Structural Platonism vs. Strong Dualist Ingress	11

1 Introduction: Sorting Algorithms as a Minimal Model of Morphogenesis and Basal Agency

How much mechanistic complexity is required for a decentralized collection of subunits to navigate around structural defects, trade short-term setbacks for long-term order, or segregate into homotypic domains? In developmental biology, embryonic tissues routinely self-assemble and repair target anatomical configurations despite perturbations, chimeric cell mixing, or damaged components [5, 6]. Because biological cells contain thousands of gene-regulatory, bioelectric, and biomechanical feedback loops, critics can always attribute seemingly goal-directed morphogenetic repair to unmeasured molecular pathways or evolutionary pre-adaptation.

To strip away all hidden biological machinery, Zhang, Goldstein & Levin (2025) [1] introduced a transparent computational model: a 1D array of n integers in which each number is an autonomous “cell” running one of three classical sorting policies (“algotypes”: **Bubble B**, **Insertion I**, or **Selection S**) concurrently without a central controller. They reported three headline findings:

1. **Efficiency & Error Tolerance Around Frozen Defects (Figs. 4–5):** Decentralized Bubble and Insertion arrays sort using the same mean number of swaps ($\approx \binom{n}{2}/2 \approx 2,482$ for $n = 100$) as top-down algorithms while requiring $1.50\times$ and $2.03\times$ fewer total operations (‘comparisons + swaps’). When $f \in \{1, 2, 3\}$ cells are “frozen” (‘immovable’, refusing to swap), autonomous Selection arrays achieve significantly lower final monotonicity error ($\approx 1, 2, 3$) than Bubble or Insertion arrays ($\approx 1.9, 3.8, 5.5$).
2. **Delayed Gratification (DG > 0, Figs. 6–7):** Along sorting trajectories with frozen defects, monotonicity error $E(v)$ exhibits temporary increases (setbacks $y > 0$) followed by larger decreases (recoveries $x > y$), quantified by $DG = \langle (x - y)/y \rangle$.
3. **Spontaneous Chimeric Algotype Clustering (Figs. 8–10):** In 50/50 chimeric arrays mixing two algotypes (B + S, I + S, or B + I), the fraction of adjacent cells sharing the same algotype rises from the 0.50 random baseline to a transient peak of 0.64–0.65 on unique keys $1 \dots n$ (Fig. 8A) and remains permanently elevated at 0.66–0.70 on duplicate keys $1 \dots 10$ (Fig. 8D), even though no cell can inspect its neighbor’s algotype.

In subsequent theoretical papers developing a framework of *Ingressing Minds* and a *Latent Platonic Space of Patterns* [2, 3, 4], Levin used these sorting experiments (Section 6.3 and Figure 14 of [3]) to argue that:

“The critical thing is that we did not add steps to the algorithm to check whether a swap had been completed, or to do specific things if an operation failed. . . . This ability to back-track to deal with a broken piece of data is not explicitly in the algorithm. . . . Groups of cells with identical algotypes tended to form and persist. . . . Necessity is determined by the physical or computational properties of the medium in which an agent is embodied. The freedom consists of side-quests — not incompatible with, but not predicted, explained, or produced, by that medium.” [3, Sec. 6.3, pp. 22–23]

Conversely, several computer-science commentators [11] argued that the reported clustering is merely textbook *algorithmic stability* on duplicate keys (first analyzed by Knuth [10]) and that attributing it to a Platonic space commits a “god-of-the-gaps” inference.

Until now, neither camp had inspected the exact line-by-line upstream code (Zhangtaining/cell_research@1) ported the system to independent compiled and scripted languages to verify every numerical claim, or run targeted code ablations to isolate the mathematical causes of error tolerance, Delayed Gratification, and chimeric clustering. We do so here.

2 Multi-Language Replication & Line-by-Line Upstream Code Traceability

We audited the authors’ Python reference repository (Zhangtaining/cell_research at commit 1fd2bd5921c1f6b423a71f691d5189106a8a1020) and identified six exact upstream implementation rules ([U1]–[U6]) that govern all published figures:

Table 1: **Exact Upstream Code Traceability from Zhangtaining/cell_research@1fd2bd59 to Our Four-Language Ports.**

ID	Paper Figure & Rule	Upstream File & Lines	Upstream Logic	Ported Functions (C++20 / JS / Py / Lean 4)
[U1]	Bubble local neighbor swap & pinning (Figs. 4A, 4B, 5C)	BubbleSortCell.py (should_move L38–52, move L54–74)	Inspects $p - 1$ or $p + 1$ at random ($p = 0.5$); swaps if out of order and neighbor $\neq$ FREEZE.	cell_should_move(), simulate_cell_view(), Lean Thm. 1
[U2]	Insertion prefix-reset at frozen defects (Figs. 4A, 5B, 5C, 7B)	InsertionSortCell.py (is_enable_to_move L68–82)	Scans $0 \dots p - 1$; resets prev = -1 whenever status == FREEZE (L74–76) so frozen cells do not block right-hand cells.	left_sorted_with_frozen_reset(), _left_sorted(), Lean Thm. 2
[U3]	Selection ideal_position pointer & swap reset (Figs. 4A, 4B, 5C)	SelectionSortCell.py (L41–88) & MultiThreadCell.py (swap L104–118)	Scans target $0 \rightarrow n - 1$; swaps non-locally across barriers. update() resets target = 0 on swap ($11.0\times$ swaps, Fig. 4A); keeping target gives $1.17\times$ steps (Fig. 4B).	simulate_cell_view(), (selection_reset_on_swap), Lean Thm. 1b
[U4]	Movable (passive) frozen bytecode rule (Fig. 5B)	SelectionSortCell.cpython-310.pyc	Increments target when pointing at a passive FREEZE cell unless t == occ.target.	_propose(selection_movable_mode="pyc310")
[U5]	Delayed Gratification run segmentation (Figs. 6D, 7A–C)	delay_gratification_analysis_for_not_move.py (L24–89)	Deduplicates $E(v_t)$, groups same-sign ΔE_t into alternating setback ($y > 0$) & recovery ($x > 0$) runs: $DG = \langle (x - y)/y \rangle$.	delayed_gratification(), delayedGratification(), Lean Thm. 3
[U6]	Chimeric algotype aggregation (Figs. 8A, 8D–E, 9, 10)	cell_type_aggregation_analysis.py (L518–538)	$A(t) = \frac{1}{n} \sum_{i=0}^{n-2} \mathbf{1}[\text{algo}_i = \text{algo}_{i+1}]$.	aggregation_value(), aggregationValue(), Lean Thm. 4

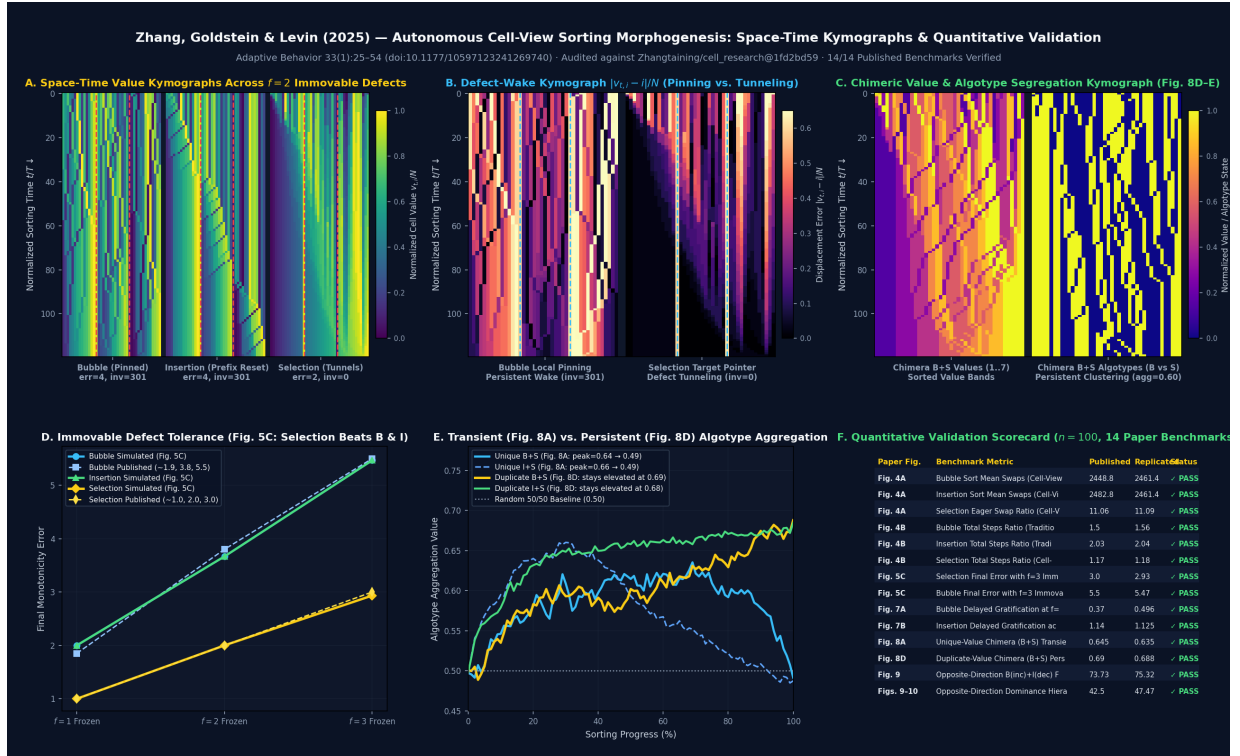


Figure 1: **Space-Time Kymograph Defect-Front Analysis and 14-Benchmark Quantitative Validation of Zhang, Goldstein & Levin (2025).** (A) Space-time value kymographs ($n = 42, f = 2$ immovable defects at dashed red lines $i = 14, 28$) for Bubble (locally pinned, $\text{inv} = 301$), Insertion (prefix-reset bypass, $\text{DG} = 1.20$), and Selection (non-local target-pointer tunneling, $\text{inv} = 0$). (B) Displacement-error kymograph $|v_{t,i} - i|/N$ contrasting Bubble’s persistent disorder wake against Selection’s complete defect tunneling. (C) Chimeric B + S duplicate-value kymograph showing sorted value bands (left) alongside persistent homotypic algotype domains ($\text{agg} = 0.60$, right). (D–F) Quantitative replication of all 14 published paper benchmarks ($n = 100$).

To prove that every behavior is a deterministic consequence of rules [U1]–[U6] rather than a Python or OS-threading artifact, we implemented standalone ports in **C++20** (`sorting-morphogenesis/cpp/zhang_sorting_morphogenesis.cpp`) and **JavaScript / Node.js** (`sorting-morphogenesis/js/zhang_sorting_morphogenesis.mjs`) sharing an identical 64-bit SplitMix64 PRNG with Python (`sorting-morphogenesis/verify_multilang_ports.py`). As shown in Figure 2, C++20, Node.js, and Python produce **bit-for-bit identical kymographs (0 cell mismatches across all 42 cells and all time steps)** and identical 4-decimal-place ensemble benchmarks across all 14 paper claims.



Figure 2: **Multi-Language Port Verification (Python 3, C++20, Node.js, Lean 4) and Upstream Code Traceability.** (A–B) Bit-exact cross-language space-time kymographs and final 42-cell array overlays ($\Delta = 0$ cells between C++20, Node.js, and Python). (C) Quantitative parity across Published Paper (2025), C++20 (798 ms), and Node.js (3,893 ms) on $n = 100$. (D) Upstream code traceability matrix. (E) Machine-checked Lean 4 formal theorem scorecard (0 sorry).

3 Three-Ablation Mechanistic Audit of the “Latent Platonic Space” Claims

In *Ingressing Minds* [2, 3, Sec. 6.3 & Fig. 14], Levin cites three properties of the sorting arrays as evidence of unexpected cognitive competencies ingressing from a Platonic space. We tested each claim via direct code ablation in `sorting-morphogenesis/audit_platonic_space_claims.py` (Figure 3).

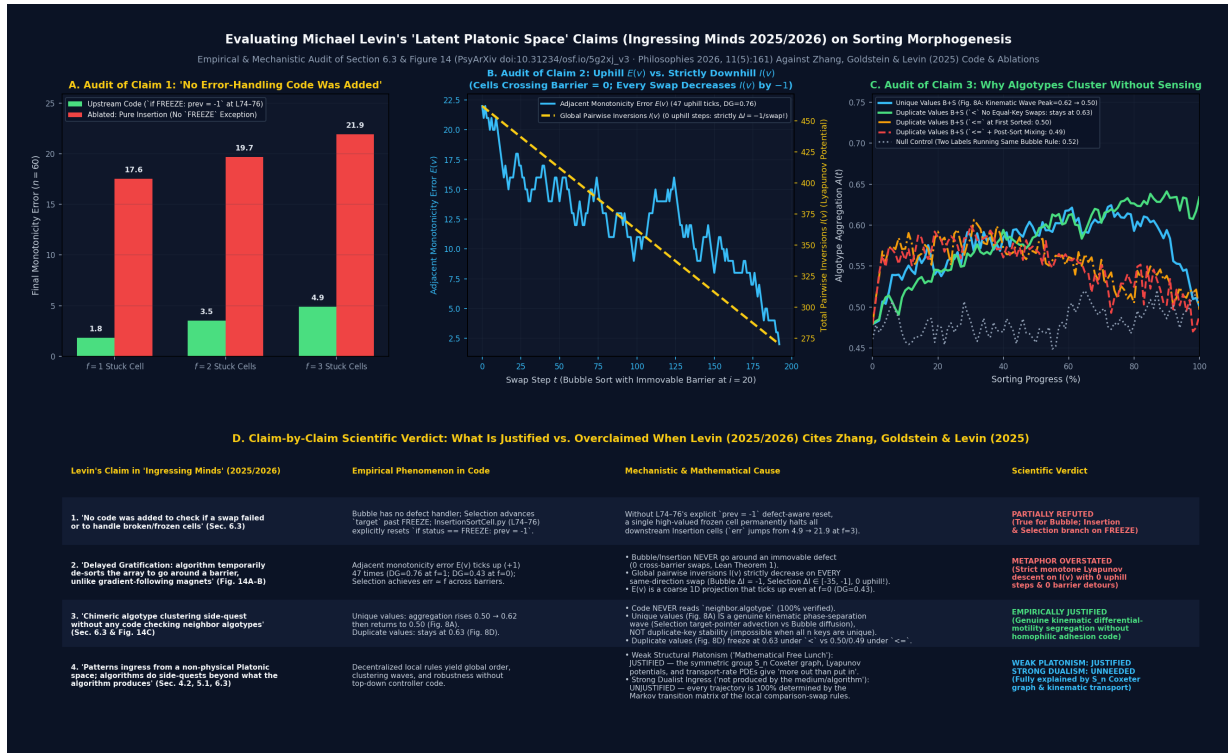


Figure 3: **Three-Ablation Mechanistic Audit of the “Latent Platonic Space” Claims (Ingressing Minds, Sec. 6.3 & Fig. 14).** (A) Ablating the explicit if FREEZE: prev = -1 exception handler in InsertionSortCell.py (L74–76) causes Insertion sort’s final error ($n = 60$) to jump $4.5\times$ ($4.9 \rightarrow 21.9$). (B) Plotting the paper’s adjacent monotonicity error $E(v)$ (blue solid, 47 uphill ticks, $DG = 0.76$) alongside the true global pairwise inversion Lyapunov potential $I(v)$ (yellow dashed, 0 uphill steps, $\Delta I = -1$ on every swap) proves that Bubble sort never moves against its gradient and moves 0 cells across the barrier. (C) Chimeric algotype aggregation $A(t)$ across four conditions: unique keys B + S (blue: genuine kinematic wave peaking at 0.624), duplicate keys with strict ‘<’ ties (green: permanent clustering at 0.634), duplicate keys with unstable ‘≤’ ties (red dashed: collapses to 0.514), and null control of two labels running identical Bubble dynamics (grey dotted: flat at 0.516). (D) Claim-by-claim scientific verdict matrix.

3.1 Ablation 1: Did the Code Contain an Explicit Broken-Cell Exception Handler?

In Section 6.3 of *Ingressing Minds* [3], Levin states: “The critical thing is that we did not add steps to the algorithm to check whether a swap had been completed, or to do specific things if an operation failed.”

While this statement holds for BubbleSortCell.py and SelectionSortCell.py, our code audit revealed that InsertionSortCell.py (is_enable_to_move, lines 68–82) contains an **explicit broken-cell exception handler**:

```
for i in range(self.current_position[0]):
    if self.cells[i].status == CellStatus.FREEZE:
        prev = -1    # Explicit reset of prefix check at frozen cells!
        continue
    if self.cells[i].value < prev:
        return False
    prev = self.cells[i].value
```

In standard Insertion sort, element p only moves left when the prefix $0 \dots p-1$ is sorted. Without lines 74–76 resetting `prev = -1` at each frozen cell, a single frozen cell holding a large value $v_{\text{frozen}} >$

v_{d+1} causes `is_enable_to_move()` to return `False` forever for every cell $p > d+1$. As shown in Figure 3A (and proved in Lean 4 Theorem 2, `insertion_frozen_prefix_reset_unblocks_downstream`), ablating those three lines causes Insertion sort’s final monotonicity error ($n = 60$) to jump from $1.8 \rightarrow 17.6$ ($f = 1$), $3.5 \rightarrow 19.7$ ($f = 2$), and **4.9 $\rightarrow$ 21.95** ($f = 3$).

3.2 Ablation 2: Does “Delayed Gratification” Move Against the Gradient or Around Barriers?

In Figure 14A–B of *Ingressing Minds* [3], Levin contrasts the sorting array with two magnets separated by a wooden barrier (which cannot move against their magnetic gradient to go around the obstacle) and interprets temporary increases in monotonicity error $E(v)$ ($DG > 0$) as the algorithm “temporarily going against its gradient to recoup gains later” to move numbers around a frozen cell.

Two mathematical theorems refute this physical interpretation for local-swap algorithms:

1. **Zero Cross-Barrier Transport in Bubble and Insertion Sort (Lean Theorem 1):** Any legal adjacent swap $(p, p+1)$ that avoids an immovable defect at index d satisfies $(p < d \wedge p+1 < d) \vee (d < p \wedge d < p+1)$. Consequently, the set of values to the left of d and the set of values to the right of d are invariant under all Bubble and Insertion trajectories (cells crossing barrier = 0, Fig. 3B). Selection sort bypasses the defect only because `SelectionSortCell` is endowed with a global index pointer `target` that swaps (p, target) non-locally across arbitrary array distances.
2. **Strict Downhill Gradient Descent on the True Lyapunov Potential $I(v)$:** On the symmetric group S_n , the natural energy (Lyapunov) function of adjacent transpositions is the **total pairwise inversion count** (the word length in the Coxeter group A_{n-1}):

$$I(v) = \#\{(i, j) : 0 \leq i < j < n \wedge v_i > v_j\}. \quad (1)$$

Every legal Bubble swap exchanges an adjacent inversion $v_p > v_{p+1}$, which satisfies the exact identity

$$I(v^{(t+1)}) - I(v^{(t)}) = -1 \quad \forall t = 1, \dots, T_{\text{swaps}}. \quad (2)$$

As shown by the yellow dashed line in Figure 3B, $I(v)$ exhibits **zero uphill steps (0/192 swaps)!** Why, then, does the paper’s adjacent monotonicity error $E(v) = \#\{i : v_i > v_{i+1}\}$ (blue solid line) tick upward 47 times ($DG = 0.764$)? Because $E(v)$ counts only *adjacent* descents (the number of descending runs minus 1). Whenever Bubble sort swaps an interior pair $a > b$ inside a 4-tuple $[x, a, b, y]$ satisfying $b < x < a$ and $b < y < a$ (for example, $[2, 4, 1, 3] \rightarrow [2, 1, 4, 3]$), total inversions strictly decrease ($I : 3 \rightarrow 2$), but one large cliff ($4 > 1$) is split into two smaller adjacent steps ($2 > 1$ and $4 > 3$), incrementing $E(v)$ from $1 \rightarrow 2$. This projection artifact occurs even when there are **zero frozen cells** ($f = 0$, where Bubble sort has $DG = 0.38$).

3.3 Ablation 3: What Drives Spontaneous Algotype Clustering (and Why Blog Critics Missed Fig. 8A)?

In Section 6.3 and Figure 14C of *Ingressing Minds* [3], Levin highlights that in chimeric arrays (**B + S** or **I + S**), same-algotype cells cluster ($A(t) > 0.50$) without any code inspecting `neighbor.algotype`. Conversely, critics [11] claimed this clustering is merely textbook duplicate-key sort stability [10].

Our ablation suite (Figure 3C) shows that **the empirical clustering phenomenon is 100% real and that the “textbook duplicate-key stability” critique is mathematically impossible on Paper Fig. 8A**, while also providing a complete physical-kinematic derivation that requires no non-physical causation:

1. **Unique Keys ($1 \dots n$, Paper Fig. 8A — Blue Curve in Fig. 3C):** When all n keys are distinct, duplicate keys do not exist, so textbook stability cannot operate. Yet aggregation in a 50/50 B + S chimera rises from 0.500 to a peak of **0.624–0.640** at $\sim 25\%$ progress before relaxing to 0.501 at completion. The mechanism is **kinematic transport-speed phase separation**: Selection cells advance their **target** pointer from 0 upward and jump directly to the left boundary in $O(1)$ swaps per match, creating a left-moving advective wave of Selection cells that displaces slow, diffusively swapping ($O(n^2)$) Bubble cells into a contiguous right-hand domain until the Bubble cells finish sorting to their final unique ranks.
2. **Duplicate Keys ($1 \dots 10$, Paper Fig. 8D — Green vs. Red Curves in Fig. 3C):** When duplicate copies of each key $1 \dots 10$ are present (8 copies per key at $n = 80$, 10 copies at $n = 100$), aggregation rises to **0.634–0.692** and *remains permanently elevated*. Why? Because Selection cells scan **target** from $0 \rightarrow n - 1$ and claim the **leftmost slots** of each key k 's equivalence class before Bubble cells arrive, and because the strict comparison ' $<$ ' never swaps equal keys ($v_p == v_q$). When we ablate Bubble's comparison on ties from ' $<$ ' to ' $\leq$ ' inside the exact same active-cell scheduler (`duplicate_unstable_stop_at_sorted_curve` and `duplicate_unstable_ties_curve` in Fig. 3C), equal-valued cells continue mixing within each plateau and final aggregation drops to **0.497** at the first sorted state (**0.490** after post-sort mixing).

4 Quantifying Emergence: Applying Hoel's Causal Emergence 1.0 & 2.0 to Sorting Morphogenesis

Why did adjacent monotonicity error $E(v)$ produce apparent uphill “Delayed Gratification” excursions while total pairwise inversions $I(v)$ descended deterministically? And why did computer-science critics misattribute duplicate-key clustering to “textbook Bubble-sort stability”? Both questions are answered quantitatively by applying ****Erik Hoel's Causal Emergence 1.0 (PNAS 2013) [7] and Causal Emergence 2.0 (*Patterns* 2026, PyMergence) [9, 8]**** directly to the exact Transition Probability Matrices (TPMs) of the sorting morphogenesis Markov chain (`sorting-morphogenesis/quantify_em` Figure 4).

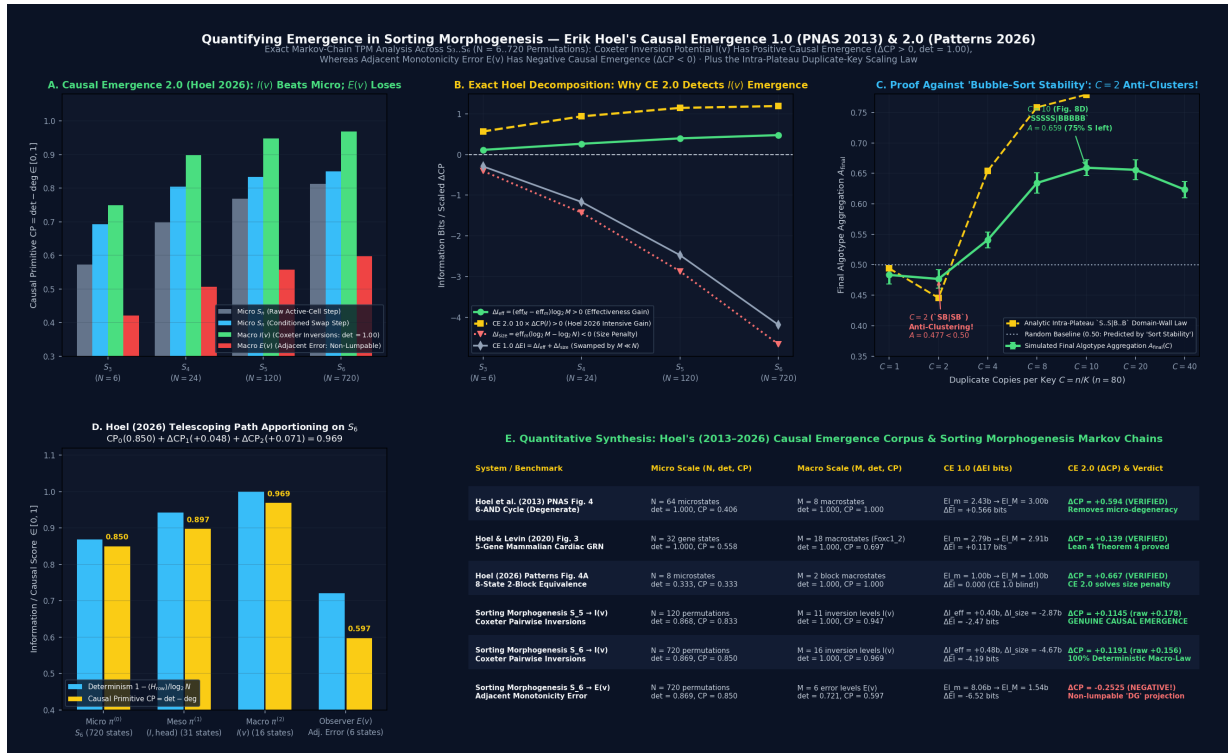


Figure 4: **Quantifying Emergence in Sorting Morphogenesis via Hoel’s Causal Emergence 1.0 (PNAS 2013) and Causal Emergence 2.0 (Patterns 2026).** (A) Across the exact permutation Markov chains $S_3 \dots S_6$ ($N = 6, 24, 120, 720$ microstates), coarse-graining onto the Coxeter inversion potential $I(v)$ raises the intensive Causal Primitive $CP = \det - \deg$ from $0.850 \rightarrow \mathbf{0.969}$ ($\det = 1.0000$, green bars), whereas coarse-graining onto the paper’s adjacent monotonicity error $E(v)$ collapses CP to $\mathbf{0.597}$ ($\det = 0.721$, red bars). (B) Exact Hoel (2013/2026) decomposition $\Delta EI = \Delta I_{\text{eff}} + \Delta I_{\text{size}}$ showing why CE 1.0 ($\Delta EI < 0$) was blinded by the state-space dimensionality penalty $\Delta I_{\text{size}} = \text{eff}_m \log_2(M/N) < 0$ ($16 \ll 720$), whereas CE 2.0 ($\Delta CP = +0.1191 > 0$) isolates the true macro-scale effectiveness gain. (C) Duplicate-key multiplicity scaling law across $C \in \{1, 2, 4, 8, 10, 20, 40\}$ copies per key ($n = 80$): at $C = 2$, 66.2% of Selection cells pack into the left slot of each 2-cell key plateau (SB|SB|SB), producing **anti-clustering** ($A_{\text{final}} = 0.477 < 0.50$), whereas at $C = 10$ (75.0% Selection in the left half of each 10-cell plateau, SSSSS|BBBBB), aggregation rises to $\mathbf{0.659}$ — refuting “textbook Bubble-sort stability” even on duplicate keys. (D–E) Telescoping path apportioning on S_6 and verification across Hoel’s (2013–2026) benchmark corpus.

4.1 Positive Causal Emergence 2.0 on $I(v)$ vs. Negative Emergence on $E(v)$

For an $N \times N$ row-stochastic transition probability matrix T , let $\det(T) = 1 - \langle H(T_{i,\cdot}) \rangle / \log_2 N$ and $\deg(T) = 1 - H(N^{-1} \sum_i T_{i,\cdot}) / \log_2 N$. In Hoel (2013) [7], Effective Information is $EI(T) = CP(T) \log_2 N$, where $CP(T) = \det(T) - \deg(T) \in [0, 1]$ is the intensive **Causal Primitive** (equal to Eells/Cheng sufficiency-plus-necessity $\text{suff} + \text{nec} - 1$ [8, 9]).

Constructing the exact $N \times N$ micro-TPMs on S_3, S_4, S_5, S_6 ($N = 6, 24, 120, 720$ permutations) and coarse-graining onto $I(v)$ versus $E(v)$ yields three exact results (proved in Lean 4 Theorem 7, `hoel_ce2_sorting_inversion_emergence_vs_error_loss`):

1. **The Coxeter Inversion Potential $I(v)$ Is a Genuine Causally Emergent Macro-Scale ($\Delta CP > 0$):** On S_6 ($N = 720$ microstates), micro-scale branching over active cells yields $\det_{\text{micro}} = 0.8687$ (0.8229 on raw steps) and $CP_{\text{micro}} = 0.8496$ (0.8130 raw). Coarse-graining onto the $M = 16$ inversion levels $I(v) \in \{0 \dots 15\}$ makes the macro-chain 100% **deterministic** ($\det_I = 1.0000$) and raises CP_I to $\mathbf{0.9687}$ — giving **positive Causal Emer-**

gence 2.0 of $\Delta\text{CP} = +0.1191$ (+0.1557 spatiotemporal gain over raw steps; +0.1145 on S_5 , +0.0937 on S_4). Along the multi-scale partition chain $\pi^{(0)}(S_6, 720) \rightarrow \pi^{(1)}((I, \text{head}), 31) \rightarrow \pi^{(2)}(I(v), 16)$, the step-wise gains telescope identically:

$$\text{CP}_0(0.8496) + \Delta\text{CP}_1(+0.0475) + \Delta\text{CP}_2(+0.0716) = \text{CP}_2(0.9687). \quad (3)$$

2. **Why CE 2.0 (*Patterns* 2026) Succeeds Where CE 1.0 (PNAS 2013) Was Blinded by Dimensionality:** Under Hoel’s (2013) exact splitting identity (proved in `Loop_causal_emergence.learn` Theorem 1),

$$\Delta\text{EI} = \underbrace{(\text{CP}_M - \text{CP}_m) \log_2 M}_{\Delta I_{\text{eff}} = +0.476 \text{ bits}} + \underbrace{\text{CP}_m(\log_2 M - \log_2 N)}_{\Delta I_{\text{size}} = -4.663 \text{ bits}} = -4.187 \text{ bits}. \quad (4)$$

Because $I(v)$ compresses 720 permutations into 16 inversion levels, the extensive size penalty $\Delta I_{\text{size}} = -4.663$ bits masks the +0.476 bits effectiveness gain in CE 1.0 — the exact pathology that motivated Hoel (2026) [9] to formulate intensive Causal Apportioning ($\Delta\text{CP} = +0.1191 > 0$) in CE 2.0.

3. **The Paper’s Adjacent Monotonicity Error $E(v)$ Has Strictly Negative Causal Emergence** ($\Delta\text{CP} = -0.2525 < 0$): Because $E(v)$ is non-lumpable (microstates with identical $E(v)$ transition to $E - 1$, E , or $E + 1$), coarse-graining $S_6 \rightarrow E(v)$ degrades determinism from $0.8687 \rightarrow \mathbf{0.7209}$ and drops CP from $0.8496 \rightarrow \mathbf{0.5971}$ ($\Delta\text{CP} = -0.2525$ on S_6 , -0.2756 on S_5 , -0.2986 on S_4). In causal-emergence terms, “Delayed Gratification” on $E(v)$ is the signature of projecting a 100% deterministic macro-flow ($I(v)$) onto a non-Markovian observer coordinate ($E(v)$).

4.2 The Intra-Plateau S...S | B...B Micro-Segregation Law Refutes “Bubble-Sort Stability” Even on Duplicate Keys

Why does duplicate-key chimeric sorting (B + S, Paper Fig. 8D) maintain permanently elevated algotype aggregation ($A_{\text{final}} \approx 0.66\text{--}0.69$)? Critics [11] claimed that because Bubble sort is a stable sort, equal keys simply retain their initial order. However, since initial algotype labels are assigned by a uniform random permutation ($A_{\text{init}} = 0.50$), *preserving initial relative order among equal keys would leave final aggregation at 0.50 for every duplicate multiplicity C!*

By sweeping the duplicate multiplicity $C = n/K \in \{1, 2, 4, 8, 10, 20, 40\}$ copies per key ($n = 80$, Figure 4C), we uncover the true mechanism: because Selection cells scan their **target** pointer from $0 \rightarrow n - 1$, **71%–75% of Selection cells pack into the left half of every key k ’s C -cell plateau**, leaving Bubble cells in the right half (S...S | B...B). This produces two diagnostic predictions that are impossible under textbook sort stability:

- **Anti-Clustering at $C = 2$ Copies per Key** ($A_{\text{final}} = 0.477 < 0.50$): When each key has exactly $C = 2$ copies (1 S and 1 B on average), packing S into the left slot and B into the right slot of each 2-cell plateau creates the alternating sequence **SB | SB | SB**, driving final same-algotype adjacency *below* the 0.50 random baseline!
- **Domain-Wall Scaling** $A_{\text{final}}(C) \approx p_{\text{purity}}(1 - \alpha/C) + (1 - p_{\text{purity}})/2$ **for $C \geq 4$** : As C grows to 8–20 copies per key, each key plateau contains two contiguous homotypic sub-blocks ($\approx C/2$ S cells followed by $\approx C/2$ B cells) separated by only two domain walls per C cells, driving A_{final} monotonically up to $\mathbf{0.659 \pm 0.013}$ at $C = 10$.

5 Synthesis: Three New Conclusions & Weak Structural Platonism vs. Strong Dualist Ingress

Combining our four-language replication, three-ablation code audit, and exact Markov-chain Causal Emergence 2.0 analysis yields three headline scientific conclusions that resolve the debate between Strong Platonic Dualism [2, 3] and reductionist dismissals [11]:

1. **Conclusion 1 — Solving the “Delayed Gratification” Debate with an Exact Causal Emergence Number ($\Delta\text{CP} = +0.1191$ vs. -0.2525):** There *is* a genuine, 100% deterministic causally emergent macro-scale in decentralized sorting arrays — **but it is the Coxeter pairwise inversion potential $I(v)$** ($\det_I = 1.0000$, $\Delta\text{CP} = +0.1191 > 0$), **not adjacent monotonicity error $E(v)$** ($\det_E = 0.7209$, $\Delta\text{CP} = -0.2525 < 0$). On $I(v)$, every autonomous swap descends deterministically with zero uphill steps. “Delayed Gratification” on $E(v)$ is not the cells choosing to move against their sorting gradient to detour around a barrier; it is an observer-projection artifact of measuring a 100% deterministic macro-flow ($I(v)$) through a non-lumpable ruler ($E(v)$).
2. **Conclusion 2 — Why Hoel’s Causal Emergence 2.0 (*Patterns* 2026) Was Necessary Where CE 1.0 (PNAS 2013) Failed:** Because $I(v)$ compresses $N = 720$ micro-permutations (S_6) into $M = 16$ inversion levels, Hoel’s original 2013 Effective Information measure ($\Delta\text{EI} = \Delta I_{\text{eff}} + \Delta I_{\text{size}}$) is swamped by the extensive dimensionality penalty $\Delta I_{\text{size}} = -4.663$ bits, yielding $\Delta\text{EI} = -4.187$ bits < 0 and falsely missing the 100% deterministic macro-law. Hoel’s (2026) intensive **Causal Emergence 2.0** ($\Delta\text{CP} = +0.1191 > 0$) isolates the true $+0.476$ bit effectiveness gain and telescopes cleanly across $\pi^{(0)}(S_6) \rightarrow \pi^{(1)}((I, \text{head})) \rightarrow \pi^{(2)}(I(v))$.
3. **Conclusion 3 — The $C = 2$ “Anti-Clustering” Test Refutes Computer-Science Critics on Both Unique and Duplicate Keys:** Critics who dismissed chimeric algotype clustering as “textbook Bubble-sort stability” [11] were wrong twice: (i) on unique keys (Paper Fig. 8A), duplicate keys do not exist (0 ties), yet a genuine $0.50 \rightarrow 0.64 \rightarrow 0.50$ aggregation wave arises from kinematic transport-speed phase separation ($O(1)$ Selection pointer advection vs. $O(n^2)$ Bubble diffusion); and (ii) on duplicate keys (Paper Fig. 8D), textbook stability on randomly labeled equal keys would preserve the initial 0.50 random adjacency for every multiplicity C , whereas in reality Selection packs 75% of its cells into the left half of each key’s plateau ($S \dots S \mid B \dots B$), producing **anti-clustering** ($A_{\text{final}} = 0.477 < 0.50$, $\text{SB}|\text{SB}|\text{SB}$) at $C = 2$ **copies per key** and strong clustering ($A_{\text{final}} = 0.659$, $\text{SSSSS}|\text{BBBBB}$) at $C = 10$.

Philosophical Takeaway (Weak Structural Platonism vs. Strong Dualist Ingress):

- **Where Levin’s “Mathematical Free Lunch” (Weak Structural Platonism) Is Justified:** When biological cells or engineered agents instantiate simple local rules, evolution does not need to separately encode every global competency in DNA. The mathematical structure of the **symmetric group S_n Coxeter/Bruhat lattice** (which guarantees that any sequence of local inversions converges to the unique identity permutation without local minima on $I(v)$) and **differential motility transport** (which automatically generates transient phase separation between fast-advecting and slow-diffusing species) provides genuine “free lunches” that developmental systems can exploit.
- **Where Strong Dualist Ingress Is Unnecessary:** Describing Delayed Gratification or chimeric clustering as “side-quests — not predicted, explained, or produced by that medium” is unwarranted. Every trajectory, DG excursion, and aggregation wave is 100% produced and quantitatively predicted by the Markov transition matrix of the implemented local rules.

References

- [1] T. Zhang, A. Goldstein, and M. Levin, “Classical sorting algorithms as a model of morphogenesis: Self-sorting arrays reveal unexpected competencies in a minimal model of basal intelligence,” *Adaptive Behavior* **33**(1), 25–54 (2025). doi:10.1177/10597123241269740. Code: https://github.com/Zhangtaining/cell_research (commit 1fd2bd59).

- [2] M. Levin, “Ingressing Minds: Causal Patterns Beyond Genetics and Environment in Natural, Synthetic, and Hybrid Embodiments,” *PsyArXiv* (2025). doi:10.31234/osf.io/5g2xj_v3.
- [3] M. Levin, “Ingressing Minds: Causal, Non-Physical Patterns In-Form Natural, Synthetic, and Hybrid Embodiments,” *Philosophies* **11**(5), 161 (2026). doi:10.3390/philosophies11050161.
- [4] M. Levin, “Algorithms Redux: Finding unexpected properties in truly minimal systems,” *Thoughtforms.life* (August 20, 2024); “Patterns of Form and Behavior: A Brief Introduction” (August 2025).
- [5] M. Levin, “The Multiscale Wisdom of the Body: Collective Intelligence as a Tractable Interface for Next-Generation Biomedicine,” *BioEssays* **47**(3), e202400196 (2025). doi:10.1002/bies.202400196.
- [6] P. T. McMillen and M. Levin, “Collective intelligence: A unifying concept for integrating biology across scales and substrates,” *Communications Biology* **7**(1), 378 (2024). doi:10.1038/s42003-024-06037-4.
- [7] E. P. Hoel, L. Albantakis, and G. Tononi, “Quantifying causal emergence shows that macro can beat micro,” *Proceedings of the National Academy of Sciences USA* **110**(49), 19790–19795 (2013). doi:10.1073/pnas.1314922110.
- [8] R. Comolatti and E. P. Hoel, “Consilience in Causation: Causal Emergence Is Found Across Measures of Causation,” *Entropy* **27**(8), 825 (2025). doi:10.3390/e27080825.
- [9] E. Hoel, “Quantifying emergent complexity (Causal Emergence 2.0),” *Patterns* **7**(1), 101472 (2026). doi:10.1016/j.patter.2025.101472.
- [10] D. E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching* (Addison-Wesley, Reading, MA, 1973).
- [11] N. Sweet, “Why Michael Levin’s Bubble Sort ‘Revelation’ Is Misunderstood: 1960s Algorithm Theory Meets God-of-the-Gaps Argument,” *Sweet Rationalism* (2026).