

ILLUSTRATED STUDENT'S GUIDE · COMPUTER SCIENCE MEETS BIOLOGY

4 LANGUAGES + LEAN 4 PROVED

Self-Sorting Cells, “Platonic Space,” and Why a 6-Line Algorithm Sparked a Huge Scientific Debate

How 100 Kids in Gym Class Explain Morphogenesis, Optical Illusions in Math, and “Mathematical Free Lunches”

By Joel Dietz · Infinite Morphospace Project · Replication & Audits of Zhang, Goldstein & Levin (2025) and Hoel (2026)

1 The Gym-Class Puzzle: What If There Is No Gym Teacher?

Imagine 100 kids standing in a single line in gym class, each wearing a sports jersey with a random number from #1 to #100. Normally, the gym teacher (a central controller) walks up and down the line telling specific kids to trade places until everyone is lined up in order from 1 to 100.

In 2025, three scientists — Taining Zhang, Adam Goldstein, and Michael Levin — asked a fascinating question that connects computer science to how living embryos build bodies: What if there is no gym teacher? What if every kid is an independent “cell” who can only look at their neighbors and decide on their own whether to swap?

THE 3 ALGOTYPES Three Different “Cell Personalities” in the Same Line

Bubble Kids (B)

The Neighbor Swappers: Look only at the kid immediately to your left or right. If your jersey number is smaller than your left neighbor’s (or bigger than your right neighbor’s), trade places!

Insertion Kids (I)

The Careful Steppers: Only step left if every single kid to your left is already lined up in order.

Selection Kids (S)

The Long-Range Teleporters: Keep a mental finger pointing at slot #1, then #2, then #3... If your number is smaller than whoever is standing at that target slot, jump straight across the gym and swap with them!

When they ran this simulation, they noticed three wild behaviors: (1) Error Tolerance (if a few FROZEN #99 kids refuse to move, the line still sorts around them); (2) “Delayed Gratification” (the line sometimes looks messier before getting cleaner); and (3) “Secret Clique Clustering” (B kids group with B kids and S kids group with S kids, even though their jerseys hide their personality!).

Illustrated Guide 1: The Three 'Cell Personalities' (Algotypes) & How They Handle a Frozen Kid

1. Bubble Kids (B — Blue): Only Swap With Immediate Neighbors (Blocked Forever by a Frozen Wall!)

X BLOCKED! #2 can NEVER get past Frozen #99!

Neighbor swap OK! (#5 > #3)

#1 Bubble (B) #4 Bubble (B) #99 FROZEN #2 Bubble (B) #5 Bubble (B) #3 Bubble (B)

2. Insertion Kids (I — Green): Need a Sorted Left Line — Saved by 3 Lines of 'Reset' Code (if FREEZE: prev = -1)!

Special Code L74-76: if FREEZE: prev = -1 (Resets memory here!)

Because memory reset after #99, #7 sees [6, 9] is sorted & steps left!

#2 Insert (I) #99 FROZEN #6 Insert (I) #9 Insert (I) #7 Insert (I) #12 Insert (I)

3. Selection Kids (S — Orange): Use a Long-Range 'target' Pointer to Teleport Right Over the Frozen Wall!

TELEPORT! SWAP! Kid #1 points 'target = Slot 0' and swaps with #8 right over Frozen #99!

#8 Select (S) #5 Select (S) #99 FROZEN #9 Select (S) #1 Select (S) #11 Select (S)

ILLUSTRATED GUIDE 1: How the 3 personalities work — and why Bubble gets blocked by Frozen #99, Insertion needs a 3-line code rescue (if FREEZE: prev = -1), and Selection teleports right over the wall!

2

The Big Debate: “Platonic Ghost Minds” vs. “Nothing New Here!”

As soon as the paper came out, a big scientific argument broke out between two camps:

CAMP 1 · LEVIN (2025/2026)

“Unprogrammed Platonic Space Side-Quests!”

In *Ingressing Minds*, Michael Levin argued that because nobody wrote code for broken cells, going backwards around walls, or finding same-personality friends, these behaviors must be **cognitive “side-quests”** beaming in from a non-physical **Platonic Space of Patterns** (where π and prime numbers live).

CAMP 2 · CS BLOG CRITICS

“Just 1960s Textbook Bubble-Sort Stability!”

Several computer-science bloggers pushed back hard, claiming: “Come on! That clique clustering is just a 1960s textbook trick called *Bubble-sort stability* (when two kids wear the exact same number, Bubble sort never swaps them). There is nothing new here at all!”

To find out who was right, we translated their original Python code into **C++20**, **JavaScript**, and the **Lean 4** math theorem prover, ran **3 “what-if” code surgery experiments**, and tested the system with **Erik Hoel’s 2026 Causal Emergence 2.0** math. Turns out: **both camps got something right and something wrong!**

3

Discovery #1: The Sneaky 3-Line Code Rescue Inside Insertion Sort

DISCOVERY #1 · CODE SURGERY

Did They Really Add “Zero Code” for Frozen Kids?

Levin wrote that they “*did not add steps to the algorithm... to do specific things if an operation failed.*” For **Bubble (B)** kids, that’s 100% true! But **Insertion (I)** kids have a strict rule: *never move left unless everyone to your left is sorted*. If **FROZEN #99** is stuck near the front of the line, the left side is *never* sorted! When we checked lines 74–76 of `InsertionSortCell.py`, we found **3 lines of special code** (if status == FREEZE: prev = -1) that tell Insertion kids to reset their memory right after a frozen kid. When we deleted those 3 lines, Insertion Sort froze up and its error jumped **4.5× higher** (4.9 → 21.95)!

4

Discovery #2: Why “Delayed Gratification” Was an Optical Illusion of the Ruler!

ILLUSTRATED GUIDE 2: Why ‘Delayed Gratification’ Was an Optical Illusion of the Ruler!
Swapping 4 ↔ 1 in [2, 4, 1, 3] → [2, 1, 4, 3] Goes STRICTLY DOWNHILL on Ruler 1 (I: 3 → 2), While Ruler 2 Goes Up (E: 1 → 2)!

BEFORE SWAP: [2, 4, 1, 3]

RULER 1 — Total Out-of-Order Pairs I(v) = 3 (Hoel CE 2.0 = +0.119, 100% Deterministic)

4 > 3

2 > 1

#2

Bubble (B)

#4

Bubble (B)

#1

Bubble (B)

#3

Bubble (B)

1 Big Cliff: 4 > 1

RULER 2 — Only Neighbor Steps E(v) = 1 (Hoel CE 2.0 = -0.253, Non-Lumpable Illusion!)

AFTER SWAP (4 ↔ 1): [2, 1, 4, 3]

RULER 1 — Total Out-of-Order Pairs I(v) = 2 (Hoel CE 2.0 = +0.119, 100% Deterministic)

4 > 3

2 > 1

#2

Bubble (B)

#1

Bubble (B)

#4

Bubble (B)

#3

Bubble (B)

Step 1: 2 > 1 Step 2: 4 > 3

RULER 2 — Only Neighbor Steps E(v) = 2 (Hoel CE 2.0 = -0.253, Non-Lumpable Illusion!)

ILLUSTRATED GUIDE 2: Watch 4 kids swap 4 ↔ 1 in [2, 4, 1, 3] → [2, 1, 4, 3]. Under Ruler 1 (Total Out-of-Order Pairs I, green arches), messiness goes **STRICTLY DOWN** (3 → 2). Under Ruler 2 (Neighbor Steps E, red arrows), it goes **UP** (1 → 2) only because one big cliff (4 > 1) split into two small steps (2 > 1 and 4 > 3)!

DISCOVERY #2 · HOEL'S CAUSAL EMERGENCE 2.0

How Erik Hoel’s 2026 Math Proves Which Ruler Is Real

- Ruler 1 — Total Out-of-Order Pairs I(v) (Green Arches):** Counts *every* pair of kids where a bigger number stands left of a smaller number. Every single Bubble swap reduces I(v) by −1 (0 uphill steps out of 192 swaps!). In Erik Hoel’s 2026 *Causal Emergence 2.0* test, Ruler 1 is 100% **deterministic with positive Causal Emergence** ($\Delta\text{CP} = +0.119$).
- Ruler 2 — Only Neighbor Steps E(v) (Red Arrows):** Only looks at immediate neighbors. Because it throws away long-range pair information, it has **negative Causal Emergence** ($\Delta\text{CP} = -0.253$). Zero Bubble kids ever cross a wall — Ruler 2 just had a blind spot!

github.com/fractastical/morpholib

Page 2 of 4

5

Discovery #3: Why the Computer-Science Blog Critics Were Wrong Twice!

Remember how the online blog critics claimed: “*The Bubble and Selection kids only cluster together because of textbook Bubble-sort stability on duplicate numbers!*”? Our experiments proved the critics wrong **two different ways**:

DISCOVERY #3 · TRAFFIC WAVES & THE $C = 2$ SMOKING GUN

Sports Cars vs. Pedestrians — Plus “Anti-Clustering” at $C = 2$!

1. **Proof 1 — Zero Duplicate Numbers ($1 \dots 80$, All Unique):** When every kid wears a different number, “duplicate-key stability” is impossible because there are zero duplicates! Yet halfway through sorting, **B** and **S** kids still cluster together ($0.50 \rightarrow 0.624$)! Why? Because **Selection (S)** kids move like **sports cars** (teleporting small numbers straight to the front of the line) while **Bubble (B)** kids move like **pedestrians**, shoving a wave of Bubble kids into the right half of the gym!

2. **Proof 2 — The $C = 2$ “Anti-Clustering” Smoking Gun on Duplicate Numbers:** If textbook Bubble-sort stability merely preserved the starting random order of equal numbers, clustering would stay flat at **0.50** (random) for every group size C . Instead, because Selection kids always scan from the left ($0, 1, 2 \dots$), 75% of Selection kids grab the **LEFT** half of every number’s group ($S \dots S \mid B \dots B$):

- With $C = 2$ copies per number, putting S on the left and B on the right of each pair creates $SB \mid SB \mid SB$, driving clustering **BELOW** random ($0.477 < 0.50$, **Anti-Clustering!**)!
- With $C = 8\text{--}10$ copies per number, each group forms $SSSSS \mid BBBB$, driving clustering **way UP** to 0.659!

Illustrated Guide 3: Why ‘Secret Cliques’ Form — And How $C = 2$ Copies Proves the Blog Critics Wrong!

Case A: $C = 2$ Copies of Each Number (#1, #1, #2, #2, #3, #3, #4, #4) → Selection Grabs the LEFT Spot of Each Pair ($S \mid B \mid S \mid B \mid S \mid B$) → **ANTI-CLUSTERING** ($0.477 < 0.50$)

Group #1: S then B!

Group #2: S then B!

Group #3: S then B!

Group #4: S then B!

Case B: Many Copies of the Same Number (e.g. Eight #1s) → Selection Packs the Left Half ($S \mid S \mid S \mid S \mid B \mid B \mid B \mid B$) → **BIG CLIQUES** ($0.659 > 0.50$)!

Left Half of Group #1: All Selection (‘Sports Cars’ Arrive First!)

Right Half of Group #1: All Bubble (‘Pedestrians’ Displaced Right!)

ILLUSTRATED GUIDE 3: Why Selection (S, orange) packs into the left half of each number’s group and pushes Bubble (B, blue) into the right half — creating $SB \mid SB \mid SB$ **Anti-Clustering** ($0.477 < 0.50$) when $C = 2$ (top) and $SSSS \mid BBBB$ **Big Cliques** (0.659) when $C = 8\text{--}10$ (bottom)!

Illustrated Guide 4: Seeing the Proof in the Real Computer Experiments!

A. Same Kids, Two Rulers: Smooth Slide vs. Bumpy Illusion

Ruler 1: Total Pairs $I(v)$ — 0 Uphill Steps! (Smooth Slide)

Ruler 2: Neighbor Steps $E(v)$ — 47 Uphill Bumps! (Illusion)

B. How $C = 2$ Anti-Clustering Refutes the Blog Critics!

Blog Critics’ ‘Stability’ Prediction (Stays at 0.50 Random)

Ideal Left-Right Split Formula ($S \dots S \mid B \dots B$)

Actual Computer Simulation (Final Clustering)

ILLUSTRATED GUIDE 4: The real computer simulation curves! Left (A): Ruler 1 ($I(v)$, green) slides smoothly downhill with 0 bumps while Ruler 2 ($E(v)$, red) has 47 illusory bumps. Right (B): Across duplicate copies C , clustering dips below 0.50 at $C = 2$ (0.477) before climbing to 0.659 at $C = 10$ — refuting the flat 0.50 blog-critic prediction!

6

The Final Scoreboard: Who Got What Right?

Mystery / Claim	Camp 1 (Levin's Platonic Claim)	Camp 2 (CS Blog Critics)	What Our Code & Math Audit Proved
1. Sorting Around Frozen Kids	“Zero special code was added for broken cells!”	(Did not check the source code)	True for Bubble & Selection, but Insertion Sort has 3 lines of reset code (if FREEZE: prev = -1)!
2. “Delayed Gratification” (Messiness Going Up)	“Cells temporarily de-sort to walk around a barrier!”	(Did not test the ruler)	Optical illusion of Ruler 2 (<i>E</i>)! Zero Bubble kids cross a barrier, and Ruler 1 (<i>I</i>) goes strictly downhill ($\Delta I = -1$) on every swap!
3. Clique Clustering (All Unique Numbers)	“An unprogrammed social side-quest!”	“Impossible without duplicate numbers!”	Critics wrong! Clustering still hits 0.624 because Selection “sports cars” outrun Bubble “pedestrians”!
4. Clique Clustering (Duplicate Numbers)	“Cells seek out same-personality neighbors!”	“Just 1960s textbook Bubble-sort stability!”	Critics wrong again! Selection grabs the left half (<i>S</i> ... <i>S</i> <i>B</i> ... <i>B</i>), causing Anti-Clustering (0.477 < 0.50) at $C = 2$ and Cliques (0.659) at $C = 10$!

THE BIG TAKEAWAY · WHY BOTH BIOLOGY AND MATH WIN

What Are “Mathematical Free Lunches” in Living Things?

1. What Michael Levin Got Right (Mathematical “Free Lunches”): You really *do* get “more out than you put in” when living cells follow simple local rules! Without any central gym teacher or friend-detector code, the geometry of permutations and differences in movement speed automatically give embryos self-sorting, error resistance, and group separation for free.

2. What Erik Hoel’s Causal Emergence 2.0 Taught Us: Sometimes a system looks spooky or chaotic only because we are measuring it with a blurry ruler! When we switch from Neighbor Steps $E(v)$ ($\Delta CP = -0.253$) to Total Out-of-Order Pairs $I(v)$ ($\Delta CP = +0.119$), the mystery vanishes and the macro-law becomes 100% deterministic.

3. Why Checking the Actual Code Always Matters: By reading all 6 lines of cell rules, testing them across Python, C++20, JavaScript, and Lean 4, and sweeping $C = 2 \dots 10$ copies per number, we uncovered both the 3-line `prev = -1` rescue in Insertion Sort and the $C = 2$ Anti-Clustering dip that neither camp had noticed!

