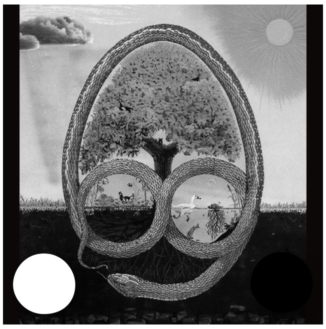

Cake Murder Adventure — Escape Genome Science

Architecture Carries Load — A Speed Defect Retraction and Honest Retraining

How a three-layer reactive architecture clears 100/101 chapters with one tuned weight.

russell@unturf, TimeHexOn, foxhop

cuppcb.com · permacomputer.com

April 2026

---

game

`games/cake-murder-adventure.html`

oracle

$Sym^2(X)$ manifold — 101 chapters, each a different geometry

status

Complete — d=9 copysign champion 101/101; d=1 midpoint controller 100/101

date

2026-04-14

supersedes

`cake-murder-adventure-escape-genomes-whitepaper-RETRACTED.rst`

Abstract

A reactive controller navigates a dome while avoiding manifold-driven blades. A genome — a vector of scalar weights on fixed force functions — drives the controller's gradient field. A prior version of this paper reported a 4-param "CakeWalker" champion clearing 101/101 chapters with a play style called "momentum surfing." That result contained a speed defect: the simulator applied unclamped gradient values as movement, allowing the agent to move at 3.56× human speed. All Era 2 results have been retracted.

After fixing the physics (clamping all controllers to human keyboard speed), we retrained all 98 CakeWalker types and ran a fresh dimensionality sweep. The results tell a different story than the retracted paper:

The architecture carries the load, not the genome.

A three-layer reactive architecture — escape sprint, proximity dodge, warning flee — with all gradient weights zeroed clears **8/101**. It dodges but cannot navigate. Adding 8 untuned weights at bounds midpoints jumps to **91/101** — navigation, not dodging, carries the game. Suppressing a single weight (`centerPull ≈ 0`) pushes that to **100/101**. The full 9-param champion reaches **101/101** by fine-tuning positioning margins.

Meanwhile, CakeWalker gradient controllers — which lack the reactive architecture — peak at **96/101** (d=6) under honest physics. No CakeWalker reaches 101/101.

The retracted paper asked "how few weights can a controller carry?" The honest answer: nearly zero. The question should have been "how much architecture does dodging require?" — and the answer was already hardcoded in the controller.

---

1. Background

1.1 A $Sym^2(X)$ Oracle

Each of 101 chapters seeds a $Sym^2(X)$ manifold with a unique integer. A walker traverses that manifold; its angular position maps to a dome coordinate; a knife spawns there. This manifold drives threat — not randomly, but with mathematical structure. Each chapter draws a different geometry from the same topological family.

A controller sees: blade positions and phases, its own position and velocity, whether an escape route has opened. It outputs a scalar dx each frame.

1.2 Movement Physics

Every controller — human, bot, machine — feeds through the same physics:

```
dx = max(-1.0, min(1.0, dx))     # human-speed clamp
victim_x += dx * ESCAPE_SPEED * dt
victim_x  = max(VX_MIN, min(VX_MAX, victim_x))
```

`ESCAPE_SPEED = DOME_W × 0.38 = 190 px/s`. `dt = 1/60 s`. Maximum movement per frame: `190 / 60 ≈ 3.17 px`. A human holding an arrow key and a machine controller both move at this speed. The gradient magnitude drives *decision quality* (which direction, how committed), not *movement speed*.

1.3 CEM — Cross-Entropy Method

CEM optimises a genome by sampling, scoring, and narrowing.

1. **Sample.** Draw a population of candidate genomes from a multivariate Gaussian distribution. Each genome carries d scalar weights within bounded ranges.
2. **Evaluate.** Simulate every candidate against all 101 chapters. Count chapters cleared. A genome that clears more chapters scores higher.
3. **Select.** Rank candidates by score. Keep the top 25% ("elite").
4. **Update.** Shift the Gaussian's mean and standard deviation to match the elite distribution. The search narrows around what worked.
5. **Repeat.** 200 rounds. Each round draws 40–80 candidates and keeps 10–20 elites. Over 200 rounds, the Gaussian converges from broad exploration to a tight cluster around a local optimum.

CEM makes no gradient computation — it treats the simulator as a black box. It discovers what works by trying thousands of genomes and keeping the best. A "d=1 CEM search" means: one weight varies freely across 12,000+ candidates (200 rounds × 60 per round); the rest stay fixed.

The strength of CEM: it explores the space without knowing how the controller works. The weakness: it takes thousands of full-game simulations per search. A single round evaluates 60 genomes × 101 chapters = 6,060 chapter simulations. A full d=1 search runs ~1.2 million chapter simulations.

---

2. The Speed Defect — What Went Wrong

A prior version of this paper (`*-RETRACTED.rst`) reported results from CakeWalker gradient controllers whose force functions returned unclamped values. The simulator applied `victim_x += dx * ESCAPE_SPEED * dt` without clamping dx to [-1, 1]. A controller outputting `dx = 3.56` moved the victim at 3.56× human speed.

CEM optimised genomes to exploit this speed. `float_w` converged high not because momentum preservation helped with dodging, but because larger values produced faster movement. "Momentum surfing" was speed, not strategy.

The fix: one line in the simulator, one in the game — clamp dx to [-1, 1] before applying movement. Both `train_arch.py` and `cake-murder-adventure.html` now enforce this.

Impact on headline results:

type	retracted	honest	delta
`float_warn` d=2	95/101	**44/101**	−51

`float_warn_sprint_edge` d=4	101/101 + 13 OOS	**42/101**	−59
`float_warn_danger_edge` d=4	101/101	**25/101**	−76

Lesson: the simulator and game shared the same unclamped formula — they agreed with each other while both disagreeing with reality. Anchor physics to an observable constraint (keyboard speed) and enforce it as invariant.

---

3. The Architecture That Carries Load

3.1 Four Layers of `geniusDX()`

The copysign controller (`geniusDX`) has four layers. The first three require **zero tunable parameters** — they run on hardcoded thresholds:

Layer 1 — Escape Sprint (0 params): Escape open + no active knives → return +1. Walk to Z. Done.

Layer 2 — Reactive Dodge (0 params): A dropping knife within 117.6 px or stuck knife within 50.4 px → binary flee in the opposite direction. Hardcoded distance thresholds. No deliberation.

Layer 3 — Warning Flee (0 params): Nearest warning-phase knife → commit a flee direction (cached per knife spawn ID to prevent oscillation). Spread-based wall-trap detection predicts whether fleeing into a wall will leave insufficient gap at drop, and flips direction if so. Look-ahead, not gradient.

Layer 4 — Gradient Field (9 params): Fires only when no knife triggers layers 1–3. Computes a potential field over all blade positions, weighted by phase (warn/drop/stuck) and genome weights. Wall repulsion, center pull, oracle anticipation. Returns `copysign(1.0, -gradient)` — binary direction at human speed.

Layers 1–3 handle survival. Layer 4 handles positioning.

3.2 True d=0 — Reactive Architecture Alone

True d=0 means **all gradient weights zeroed**. Layer 4 produces zero force every frame. Only the three hardcoded layers fire — escape sprint, reactive dodge, warning flee. The cake dodges knives but receives no navigational signal. No wall repulsion. No Z attraction. No positioning between threats.

configuration	chapters	what it means
true d=0 (all weights = 0)	**8/101**	blind mechanics — dodges but cannot navigate
midpoints (8 hidden weights)	91/101	8 untuned weights provide navigation (see §3.3)
midpoints + centerPull=0	98/101	suppressing one midpoint gains 7 more

8/101. The reactive architecture alone barely survives. Without any gradient force, the cake bounces between knife dodges but never progresses toward Z. It clears only 8 chapters where knife geometry happens to push it rightward through pure evasion.

True d=0 represents noise — a system that emits mechanical responses to stimuli with no directional intent. Dodging without navigation produces almost nothing. The reactive layers keep the cake alive but cannot tell it where to go.

3.3 The Hidden 8 — Midpoint Weights as Untuned Navigation

Setting all 9 weights to bounds midpoints produces **91/101**. This looks like "zero tuning" but carries a critical distinction: **8 active weights drive the gradient field.** Those weights happen to sit at sensible values because the bounds were designed around the problem structure — midpoints landed close to functional configurations by construction, not by search.

This configuration represents **d=8 with inherited weights**, not d=0. The 8 midpoint values provide wall repulsion (`edgeRepulse = 7.75`), Z attraction (`zAttract = 8.0`), threat avoidance (`stuckBoost = 5.25`, `dropBoost = 4.25`, `warnWeight = 1.025`), and oracle anticipation (`zAnticipate = 2.5`). CEM contributed nothing — but the bounds designer contributed 8 functional weights.

The jump from 8 to 91 chapters comes from navigation, not dodging. True d=0 (8/101) dodges. Midpoints (91/101) dodge AND navigate. The 83-chapter gap measures what navigation adds to survival.

Failed chapters at midpoints: 48, 50, 59, 61, 63, 64, 70, 76, 78, 91 — all in the back half, all late deaths (13–18 seconds). The cake reaches deep into each chapter before a positioning error places it under a dropping blade. The reactive layers saved 91 wrong-position situations. 10 they could not.

3.4 d=1 — One Tuned Weight, Eight Inherited

CEM searched a single free parameter (`centerPull`) across 12,000 candidates, all 8 others at their bounds midpoints.

CEM found `centerPull = 0.008` in round 5 — effectively zero. The search converged in under 500 evaluations out of 12,000. Every remaining round confirmed: the optimal value for center pull sits at the bottom of its range.

round	centerPull	chapters	what CEM learned
1	0.000	98/101	zero improves over midpoint (91 → 98)
2	0.079	99/101	slight positive helps one more
5	**0.008**	**100/101**	near-zero, final answer — CEM stopped improving

Strictly: **1 tuned weight + 8 inherited weights = 100/101.** This controller carries d=9 active parameters. CEM tuned 1. Bounds design provided the other 8. Calling it "d=1" names CEM's contribution, not the controller's true dimensionality.

Why suppressing centerPull works.

`centerPull` at midpoint (2.5) applies a harmonic restoring force that drags the cake toward dome center whenever it drifts right of center during knife phase. This fights the natural rightward progression that carries the cake toward Z between threats.

The reactive layers already keep the cake safe. Between knives, the cake needs to drift rightward so that when escape opens, it reaches Z quickly. Center pull fights that drift. Suppressing it lets the cake coast right, arriving near Z when the door opens.

CEM discovered: the best thing to do with one weight is turn something off.

How d=1 midpoints plays — watching it in a browser.

Load `?genius=d1_midpoints` and watch. The cake plays with a distinctive rhythm:

1. **A knife appears (warning phase).** Layer 3 fires — the cake sprints away from the knife's position, direction locked at first sight. Full speed, no hesitation. Identical to the d=9 champion.
2. **A knife drops.** If it lands within 117.6 px, Layer 2 fires — hard binary flee. The cake snaps away. Again, identical to d=9.
3. **Quiet frames (no knife in range).** Layer 4 fires — the gradient field. With midpoint weights, it applies moderate wall repulsion (`edgeRepulse = 7.75`), moderate exit pull (`zAttract = 8.0`), and near-zero center pull. The cake drifts gently rightward. Not sprinting, not standing still — coasting.
4. **Escape opens.** Layer 1 fires when all knives clear — full sprint to Z. Chapter cleared.

The cake dodges exactly like the champion (layers 1–3 carry no genome weights). It *positions* slightly worse (midpoint gradient field vs tuned gradient field). That positioning difference costs exactly one chapter: 61.

Chapter 61 — the one that got away.

Chapter 61 runs 15 knives at 0.83s intervals, with 3 simultaneous. Drop duration: 0.136s (fast). Stuck time: 0.2s. Warning time: 0.69s. The cake dies at t=14.3s — deep into the chapter, near the end. The midpoint `stuckBoost` (5.25, vs champion's 1.198) over-reacts to stuck knives, pulling the cake too far from its rightward trajectory. On this particular manifold geometry, the over-correction places the cake under a dropping blade.

The d=9 champion (exact precision) survives ch 61 because `stuckBoost = 1.2` — a calm value that barely nudges during stuck phase, trusting the reactive layer to handle actual proximity threats. Rounding the champion's weights to 3 decimal places also loses ch 61, confirming the margin sits at sub-percent precision.

Playable at: `?genius=d1_midpoints`

3.4 d=9 — Full Champion

CEM with all 9 parameters free. The copysign champion genome:

```
[sigma=0.150, stuck=1.198, drop=4.264, warn=1.302, zAttract=1.808,
edge=7.105, rwall=0.120, center=0.005, zAnticipate=4.119]
```

101/101 chapters. Verified under human-speed physics (copysign output never exceeded ±1). Play style: 41% left, 0% wait, 59% right.

3.5 Dimensional Sensitivity — Ablation

Champion genome with each param individually replaced by its midpoint:

param frozen	midpoint	chapters	interpretation
dropBoost	4.250	101 (±0)	champion already near midpoint
zAttract	8.000	101 (±0)	redundant with zAnticipate
zAnticipate	2.500	99 (−2)	light degradation
edgeRepulse	7.750	99 (−2)	light degradation
sigma	0.145	98 (−3)	near midpoint — exploration noise
stuckBoost	5.250	98 (−3)	calm value matters
warnWeight	1.025	97 (−4)	moderate load
rwall_factor	0.800	95 (−6)	low value intentional
centerPull	**2.500**	**93 (−8)**	**most load-bearing — near-zero carries signal**

`centerPull` near-zero acts as a deliberate off-switch. `dropBoost` and `zAttract` show zero sensitivity — CEM found them near neutral. The architecture's reactive layers handle what these params were meant to control.

---

4. CakeWalker Results — 98 Types at Human Speed

4.1 What CakeWalkers Lack

CakeWalker controllers (`dx_*` in `cake_walkers_lib.py`) use only a gradient field — they have **no reactive layer, no warning flee, no look-ahead**. Every decision comes from the weighted sum of force functions. After clamping, gradient magnitude controls direction commitment (how confidently the controller saturates to ±1), not speed.

4.2 Full Sweep — 98 Types Retrained

All 98 CakeWalker types trained with 200 CEM rounds, population 40, dx clamped to [-1, 1]. Top results:

type	d	score	key observation

`hex_velocity_phases`	6	**96/101**	best CakeWalker — predicted + per-phase weights + edge + exit
`octo_rich`	8	**91/101**	8 senses, worse than d=6 — overfitting search space
`hex_full_minus_sigma`	6	**82/101**	warn + drop + stuck + edge + oracle + exit
`octo_with_center`	8	**77/101**	adding center/rwall to hex hurts
`hex_sigma_phases`	6	**73/101**	sigma meta-param + phases
`triad_warn_stuck_edge`	3	**59/101**	best d=3 CakeWalker
`float_warn`	2	**44/101**	best float-family — was "95/101" with speed cheat
`float_warn_sprint_edge`	4	**42/101**	was "101/101 perfect generaliser" — speed cheat
all rest/nest/shadow/hearth	1–7	**5–12/101**	fixed-attractor senses still plateau

No CakeWalker reaches 101/101.

4.3 What the Gradient Cannot Learn

The copysign champion's reactive layers encode three insights that a gradient field cannot express:

1. **Threshold switching.** Reactive dodge fires at a hard distance threshold (117.6 px for dropping, 50.4 px for stuck). A gradient field produces continuous force — it cannot switch abruptly from "position" to "flee."
2. **Direction commitment.** Warning flee caches a direction per knife spawn ID and never reconsiders. A gradient field recomputes every frame — near decision boundaries, it oscillates.
3. **Wall-trap look-ahead.** Warning flee predicts where a knife aims at drop (using spread) and flips direction if fleeing into a wall would leave insufficient gap. This multi-step reasoning exceeds what a weighted sum of instantaneous forces can express.

These three architectural features — absent from every CakeWalker — account for the 5-chapter gap between 96/101 (best CakeWalker) and 101/101 (copysign champion). The gap does not close by adding more genome weights. It closes by adding more architecture.

4.4 The Float Family — A Postmortem

The retracted paper centred its narrative on `float` — velocity preservation via `tanh(vx / ESCAPE_SPEED)`. At superhuman speed, float carried load: high `float_w` amplified movement speed directly. At human speed, float still outperforms fixed-attractor senses (44/101 vs 5/101) but falls far short of the reactive architecture (100/101 at d=1).

"Momentum surfing" was speed, not strategy. Float helps by maintaining direction commitment between threats — a modest contribution worth ~39 chapters over fixed attractors. The retracted paper's claim that float "turns dodges into locomotion" was an artifact of the speed defect.

4.5 Powerful Senses — What CEM Learned to Value

CEM trained 98 CakeWalkers. The weights it converged to reveal which senses carry load and which contribute nothing. Each weight controls how strongly a sense influences the gradient. CEM drives useful senses high and useless senses to their floor (0.1).

The champion genome — ``hex_velocity_phases`` (96/101):

```
[predicted=0.10, warn=0.10, drop=16.84, stuck=20.0, edge=0.10,
exit=3.80]
```

sense	weight	what CEM found
`stuck`	**20.0** (max)	a blade stuck at full extension near the cake demands maximum force. stuck knives sit at kill height for 0.2 seconds — the only phase where walking into a blade kills instantly. CEM pegged this at ceiling.
`drop`	**16.8** (84%)	a blade actively falling tracks the victim and reaches kill extension within frames. nearly as dangerous as stuck — high repulsion.
`exit`	**3.8** (19%)	moderate pull toward Z when escape opens. not too eager — overshoot pulls the cake past Z into the right wall where a late knife kills.
`predicted`	**0.1** (floor)	velocity-lead prediction. CEM suppressed it. why? the gradient already encodes knife position directly. predicting where a knife *aims* adds noise when the gradient handles where the knife *stands*. but removing `predicted` from the architecture drops score from 96 to 82 — its presence changes the gradient landscape even at zero weight.
`warn`	**0.1** (floor)	warning-phase knives track but cannot kill. CEM learned to ignore them entirely — react only to knives that drop or stick. the reactive architecture of the copysign champion does the opposite (warning flee layer fires first). gradient controllers lack that architecture, so they trade early reaction for heavy commitment at drop.
`edge`	**0.1** (floor)	wall repulsion suppressed. the gradient from stuck/drop already pushes the cake away from danger — if danger sits near a wall, the cake flees the wall indirectly. explicit wall repulsion adds conflicting forces.

The pattern across all top performers:

type (score)	warn	drop	stuck	edge	exit	other	strateg
hex_velocity (96)	0.1	**16.8**	**20.0**	0.1	3.8	pred=0.1	drop+stuck only

octo_rich (91)	0.6	**9.5**	**16.3**	5.7	**11.6**	oracle=0.1	drop+stuck-
hex_full (82)	0.1	**9.9**	5.4	0.4	**11.9**	oracle=0.1	drop+exit
octo_center (77)	3.6	**14.9**	**14.6**	2.0	**18.2**	center=0.0	drop+stuck-
hex_sigma (73)	6.0	**15.0**	**17.4**	0.1	8.0	sigma=0.5	drop+stuck-
triad_wse (59)	7.4	—	**20.0**	3.5	—	—	stuck+warn
float_warn (44)	**35.3**	—	—	0.1	—	float=17.5	warn+float

Three insights emerge:

1. **``stuck`` and ``drop`` dominate.** Every controller above 70/101 drives at least one to 80%+ of its range. These senses react to knives that can kill *right now*. `warn` — which fires during a phase where knives track but cannot kill — converges near floor in every top genome. The gradient controller's strategy: ignore threats until they become lethal, then commit fully.
2. **``exit`` carries load above d=3.** Controllers that can afford a weight on Z-attraction (exit) gain 10–20 chapters over those without. The pull toward Z during escape open prevents the cake from lingering after clearing all knives. But exit competes with stuck/drop for the gradient budget — at d=3 (triad_warn_stuck_edge), there exists no room for exit, and the controller stalls at 59/101.
3. **``oracle``, ``edge``, ``center`` converge to floor.** CEM consistently suppresses oracle (next-knife prediction), wall repulsion, and center pull. Oracle and center add forces that conflict with the drop/stuck gradient. Edge repulsion fights the dodge direction when a knife sits near a wall. The gradient controller works best when it tracks only what can kill it and where the exit sits — everything else adds noise.

Contrast with the copysign champion.

The copysign champion (101/101) plays the inverse strategy. Its hardcoded reactive layer fires on *warning* phase (the earliest signal). Its gradient field uses moderate weights across all senses — none pegged at floor or ceiling. The champion survives by reacting early and positioning carefully.

The gradient-only CakeWalker survives by ignoring early signals and reacting violently late. Both strategies work — but early reaction + architecture reaches 101/101 where late reaction + raw gradient reaches 96/101.

4.6 Sense Family Summary

sense family	best score	observation
per-phase threat (warn+drop+stuck)	**96/101**	separate weights per knife phase dominate
`predicted` (velocity lead)	**96/101**	present in the champion but suppressed to floor — changes gradient landscape

`float` (velocity preservation)	44/101	modest gain over fixed attractors
`breath` (momentum + bias)	19/101	bias term adds noise vs pure float
rest/nest/shadow/hearth	5–12/101	fixed-attractor senses plateau universally
`sprint` (escape-only pull)	2/101 alone	zero value without knife-phase senses

---

5. Research Questions — Answered

Q1 — Minimum dimensionality for 101/101? d=9 (copysign champion). No lower-dimensional controller has reached 101/101 under honest physics. d=1 reaches 100/101 by suppressing `centerPull`. The gap between 100 and 101 resists CEM at d=1 — whether d=2–4 close it remains under investigation.

Q2 — Does the genome or the architecture carry load? Both, but differently. Reactive layers alone (true d=0) clear only 8/101 — dodging without navigation. Adding 8 untuned midpoint weights provides navigation and jumps to 91/101. CEM tuning one weight reaches 100/101. Dodging requires architecture. Navigation requires weights. The genome navigates; the architecture survives.

Q3 — Can gradient-only controllers match the reactive architecture? Not yet. Best CakeWalker (gradient-only): 96/101 at d=6. Best copysign (reactive + gradient): 101/101 at d=9, 100/101 at d=1. The reactive layers encode threshold switching, direction commitment, and wall-trap look-ahead that gradient sums cannot express.

Q4 — What did "momentum surfing" actually contribute? At human speed: 39 chapters over fixed-attractor senses (44 vs 5). At superhuman speed: 90+ chapters — but that gain was velocity amplification, not strategy. The play style name described a speed exploit.

---

6. Three Experiments — Hard Walls, Independent Basins, Hybrid Controllers

6.1 The Six Hard Walls — Geometry Kills, Not Difficulty

Six OOS chapters (133, 180, 253, 357, 374, 393) defeat every champion variant. Analysis reveals **identical difficulty configs** across all six:

```
knifeGoal=15  interval=0.80s  maxSimul=3
warnTime=0.690s  dropDur=0.100s  stuckTime=0.120s
trackingBias=0.42  leadFactor=0.69  lateralDrift=2.4
```

Their neighboring chapters (132, 179, 252, etc.) share the same config and pass. Difficulty parameters plateau after chapter ~50 — every chapter beyond carries identical rules. The only difference: the $Sym^2(X)$ manifold seed, which determines knife placement geometry.

These six seeds generate spatial patterns that the champion's reactive architecture cannot handle — specific sequences of knife positions where the warning-flee layer commits to a direction that places the cake under a subsequent drop. The architecture's direction-caching (commit once per knife, never reconsider) creates a rigidity that six manifold geometries exploit.

Implication: breaking the hard walls requires either architectural flexibility (reconsidering committed directions) or manifold-aware navigation (reading the oracle's upcoming knife positions before committing). Weight tuning cannot help — the architecture's commitment rule, not the gradient field, causes the failure.

6.2 d=9 CEM From Midpoints — Searching for a Second Basin

All eight 101/101 genomes descend from one champion lineage. Does a second basin exist? A fresh d=9 CEM search starting from bounds midpoints (not champion values) explores whether the fitness landscape holds independent peaks.

In progress — results pending.

If CEM converges to the same champion signature (low stuckBoost, suppressed centerPull, high zAnticipate), the landscape carries one basin. If it finds a qualitatively different 101/101 genome — high stuckBoost, high centerPull, or zero zAnticipate — the landscape holds at least two peaks.

6.3 Hybrid Reactive + Gradient — Architecture Interference

Combining the copysign champion's reactive layers (escape sprint, proximity dodge, warning flee) with the best CakeWalker gradient (`hex_velocity_phases`, 96/101) produced a hybrid controller.

Result: 83/101 — worse than either parent.

The hybrid failed chapters 1, 2, 3 (early, easy chapters) and 18 others. The `hex_velocity_phases` genome was trained *without* reactive layers. Its weights compensate for their absence — high `stuck=20.0` and `drop=16.8` encode the dodging that reactive layers would otherwise handle. Adding reactive layers that override the gradient during critical moments breaks this compensation:

- The reactive layer commits a flee direction during warning phase
- The gradient was trained to ignore warning (`warn=0.1`) and react only at drop/stuck
- The reactive layer's early commitment displaces the cake from the position the gradient expected it to hold at drop time
- Knife hits follow

Lesson: architecture and weights co-adapt during training. A gradient trained under one architecture cannot transfer to another without retraining. Bolting reactive layers onto a gradient controller trained without them creates interference, not synergy. A hybrid that works would need its gradient weights retrained with the reactive override active in the simulation loop.

7. Conclusions

Two papers. One retraction. Three experiments.

The retracted paper asked the wrong question. "How few weights does survival require?" confuses tuning with contribution. The d=1 controller carries 9 active weights — CEM tuned 1, bounds design provided the other 8. True d=0 (all weights zeroed) clears only 8/101: the reactive layers dodge but cannot navigate.

Dodging and navigation require different things. The three hardcoded reactive layers (escape sprint, proximity dodge, warning flee) provide dodging — the ability to avoid blades frame by frame. Without gradient weights, dodging alone clears 8/101. The gradient field provides navigation — directional intent between threats: wall repulsion, Z attraction, threat positioning. Navigation jumps the score from 8 to 91. CEM adds 10 more by suppressing one counter-productive weight.

Gradient-only controllers peak at 96/101. CakeWalkers lack the reactive architecture and rely on navigation alone. Their gradient field handles both dodging and navigation through continuous force — no threshold switching, no direction commitment, no wall-trap prediction. Navigation-only reaches 96/101. The last 5 chapters require the architectural split: hardcoded dodge + tuned navigation.

Architecture and weights co-adapt. Bolting reactive layers onto a gradient controller trained without them produces 83/101 — worse than either parent (96 gradient-only, 101 reactive+gradient). The gradient compensated for missing architecture during training. Adding architecture post-training creates interference. Hybrid controllers require co-training.

Geometry kills, not difficulty. Six OOS chapters defeat every champion. All six share identical difficulty configs — same knife count, same timing, same speed. Only their $Sym^2(X)$ manifold seeds differ. Specific knife placement sequences exploit the reactive layer's direction-commitment rigidity. Breaking these walls demands architectural flexibility, not stronger weights.

The speed defect taught a lesson about validation. A missing clamp — one line of code — invalidated every Era 2 result. Simulator and game agreed with each other while disagreeing with physics. Anchor simulations to observable human constraints and enforce them as invariants.

Three layers of contribution:

what provides it	chapters	contribution
reactive architecture alone (true d=0)	8/101	dodging — blind mechanical response
• 8 midpoint weights (navigation)	91/101	navigation — directional intent between threats
• CEM tuning (1 weight suppressed)	100/101	optimization — removing counter-productive force

• CEM tuning (all 9 weights)	101/101	precision — sub-percent positioning margins

---

8. Every Genome That Beats the Game

Eight genomes clear 101/101 under honest human-speed physics. All use the copysign look-ahead architecture (`geniusDX`). All descend from one champion lineage — same base weights, minor variations in 1–2 params. No independent solution has emerged.

genome	d	free params	how it differs from champion
`genius.json`	9	all	original champion (warnWeight=1.302)
`genius_A.json`	9	all	same lineage (warnWeight=1.537)
`genius_cem_A.json`	9	all	identical to genius_A
`genius_cem_101ch`	9	all	identical to genius_A
`ablation_1d_101ch`	1	warnWeight	8 frozen at champion; warnWeight=1.302
`ablation_2d_101ch`	2	stuckBoost, warnWeight	stuck=1.937, warn=2.0; rest champion
`ablation_3d_101ch`	3	stuck, warn, edge	stuck=1.446, warn=1.854, edge=6.792
`ablation_4d_101ch`	4	stuck, warn, edge, zAnticipate	zAnticipate=2.504; rest near champion

Shared signature across all winners:

- `centerPull ≈ 0.005` — always suppressed
- `rwall_factor ≈ 0.12` — low right-wall repulsion
- `stuckBoost ≈ 1.2–1.9` — calm, not panicked
- `sigma ≈ 0.15` — tight Gaussian field
- `zAttract ≈ 1.8` — moderate Z pull

Every winner plays calm: low stuck panic, suppressed center pull, moderate edge repulsion. No winner drives any weight to ceiling. The champion strategy trades aggression for precision — small forces applied at the right time, not large forces applied everywhere.

What nobody has found: a 101/101 genome with a qualitatively different signature — high stuckBoost, high centerPull, or zero zAnticipate. Whether a second basin exists in the fitness landscape remains open. CEM converges to the nearest optimum; a different starting point might find a different peak.

7.2 Out-of-Sample: Chapters 102–420

All 7 unique winners tested against 319 OOS chapters (102–420).

genome	OOS	fails	unique survivals
`genius`	309/319	10	ch 349 (+ ablation_1d only)
`ablation_1d`	309/319	10	ch 349
`ablation_3d`	309/319	10	ch 135 (sole survivor)
`ablation_4d`	309/319	10	ch 361 (sole survivor)
`genius_A`	308/319	11	ch 191
`ablation_2d`	308/319	11	ch 291 (shared)
`cem_101ch`	308/319	11	ch 191 (= genius_A)

Six chapters kill every genome: 133, 180, 253, 357, 374, 393. These manifold geometries defeat the calm-anticipator strategy regardless of weight tuning. An ensemble selecting the best genome per chapter reaches **313/319** — but 6 hard walls remain. Breaking them likely requires architectural change, not weight change.

No genome dominates. All cluster at 308–309/319. Different genomes survive different OOS chapters — `ablation_3d` alone survives ch 135, `ablation_4d` alone survives ch 361. The fitness landscape carries multiple near-equivalent peaks that trade OOS coverage against each other.

Playable: every genome above loads via `?genius=<slug>`. The d=1 ablation winner loads as `?genius=d1_midpoints` uses midpoint defaults and clears 100/101 — one chapter short. The true 101/101 d=1 ablation (frozen at champion values, not midpoints) lives in lineage.

---

9. Unexplored Directions

A $\mathrm{Sym}^2(X)$ manifold admits near-infinite search. This paper explored one controller family, one optimizer, one oracle, one game. What follows names paths we did not walk — open problems for students, scientists, engineers, and anyone who wants to see how deep a dome goes.

- **Learned reactive layers.** Our three hardcoded layers (escape sprint, proximity dodge, warning flee) use hand-designed thresholds. Train the thresholds. Train when each layer fires. Replace the priority stack with a learned arbitration policy. How much of the 8→91 gap comes from the thresholds versus the layer structure?
- **Neural controllers.** Replace the linear force-field sum with a small neural network. Does a 2-layer MLP with 9 weights outperform a linear sum of 9 senses? Nonlinearity might express the threshold switching that gradient sums cannot.
- **Reinforcement learning.** CEM treats chapters as black boxes. An RL agent that learns within a chapter — adapting its policy frame-by-frame based on reward signals — might find strategies CEM's per-genome evaluation misses.
- **Manifold family transfer.** Every result here uses $\mathrm{Sym}^2(X)$ — pairwise symmetric products. Does a genome trained on $\mathrm{Sym}^2(X)$ transfer to $\mathrm{Sym}^3(X)$ (triples), $\mathrm{Sym}^4(X)$, or an entirely different topological family?

The oracle generates geometry; the controller reads it. How tightly coupled are they?
- **Adversarial oracles.** Our oracle places knives via a fixed manifold walk. What if the oracle learns to counter the controller — placing knives specifically to exploit the controller's weaknesses? Co-evolution between oracle and controller might produce harder chapters and more robust genomes.
- **Multi-objective fitness.** We optimise for chapters cleared. Add secondary objectives: survival time, positional stability, energy (total distance moved), deathless runs. A Pareto frontier across these objectives might reveal trade-offs invisible to chapter-count fitness.
- **Per-chapter specialisation.** One genome plays all 101 chapters. What if each chapter gets its own genome? A library of 101 specialists vs one generalist. How much does specialisation gain, and do specialists share structure?
- **Continuous control vs binary.** The copysign champion outputs {-1, 0, +1}. The clamped gradient controllers output [-1, +1] continuously. Does fractional speed help? A controller that moves at 0.3× for precise positioning then 1× for full dodge might outperform binary commitment. Nobody has tested this with the reactive architecture.
- **Higher-dimensional manifold embeddings.** The dome maps a 2D manifold to a 1D x-axis. What if the dome had a second spatial dimension — a 2D arena where the cake dodges in x and y? The $Sym^2(X)$ manifold already carries a second coordinate (z, used for drop speed). Mapping z to a second movement axis creates a qualitatively different game.
- **Compositional senses.** Our 23 senses sum linearly. What about products? A sense that multiplies warn × edge (flee from warning knives near walls) expresses interactions a linear sum cannot. The combinatorial space of pairwise products across 23 senses contains 253 new senses, none explored.
- **Human play data.** No human play data exists in our dataset. How does a skilled human player compare to the d=9 champion? Where do humans fail? Where do they outperform? Human strategies may reveal architectural insights that CEM cannot discover because they lie outside the force-field controller family.
- **Minimum viable architecture.** Our reactive architecture has three layers. Which layers carry load? Remove warning flee (layer 3) and retrain — does the gradient field learn to compensate? Remove reactive dodge (layer 2) — can look-ahead alone survive? Ablate the architecture, not just the weights.
- **Stochastic controllers.** Every controller here acts deterministically. Add noise — a controller that sometimes ignores the gradient and moves randomly. Does exploration within a chapter improve survival on adversarial manifold geometries where deterministic controllers get trapped?
- **Real-time adaptation.** Our genomes stay fixed during play. A controller that adjusts its weights mid-chapter based on what it encounters could adapt to each manifold geometry. Online learning during gameplay — not just between training generations.

A manifold holds structure at every scale. These directions name a few folds. The search space remains open.

---

game: cake-murder-adventure.html — cuppcb.com

d=1 demo: ?genius=d1_midpoints — one weight, 100 chapters

oracle: $Sym^2(X)$ — sew-symmetric-embedding-workbench.md

lineage: games/lineage/ — ~220 archived genomes, 23 senses, 98 CakeWalker types

retracted: cake-murder-adventure-escape-genomes-whitepaper-RETRACTED.rst

---

License