Where Interpretability Works and Where It Breaks

Sensorimotor Modeling from Population to Individual Scale

Gil Raitses

Mirna Lab General Meeting

Syracuse University

Original Study Overview

Population-Level Sensorimotor Habituation Model

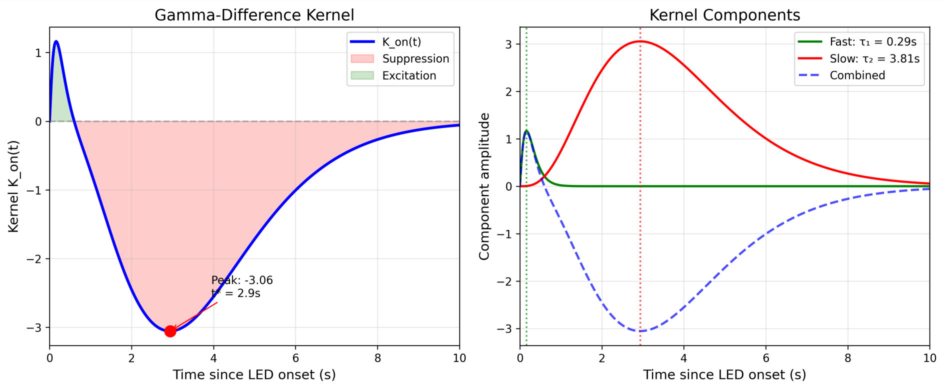
Larval reorientation behavior follows a gamma-difference kernel with two distinct timescales that capture the dynamics of sensory response and adaptation.

The fast excitatory component with $\tau_1 \approx 0.3$ seconds drives the initial response to optogenetic light onset. The slow inhibitory component with $\tau_2 \approx 4$ seconds produces delayed suppression that underlies behavioral habituation across repeated stimuli.

The model was validated across 14 experiments comprising 701 unique larval tracks collected under four stimulation conditions.

Key Result The gamma-difference kernel accurately predicts population-level reorientation dynamics and provides biologically interpretable timescales.

Kernel Structure



The gamma-difference kernel $K(t) = A \cdot \Gamma(t; \alpha_1, \beta_1) - B \cdot \Gamma(t; \alpha_2, \beta_2)$ modulates reorientation hazard rate following LED onset. Left panel shows combined kernel with peak excitation at 0.3s followed by suppression minimum at 3s. Right panel decomposes into fast excitatory gamma (green, $\tau_1 = 0.29$s) and slow suppressive gamma (red, $\tau_2 = 3.81$s).

PSTH Computation Methods

Empirical vs Parametric Approaches

Empirical PSTH The peri-stimulus time histogram bins event times relative to LED onset into 100ms intervals. Each bin contains the fraction of events at that latency. Computed directly from data using histogram binning without assuming any functional form.

Parametric PSTH Bernoulli event probability derived from fitted kernel. At each timestep, $p(t) = 1 - \exp(-\lambda(t) \cdot \Delta t)$ where $\lambda(t) = \lambda_0 \exp(K(t))$ is hazard rate modulated by kernel $K(t)$.

Model Fitting Kernel fitted by maximizing likelihood that empirical event times arise from Bernoulli process. Empirical PSTH used for visualization; parametric model enables prediction and simulation.

Simulation Track Generation

Bernoulli Point Process Framework

Simulated tracks are generated using the fitted population kernel as follows.

Step 1 For each timestep $\Delta t = 0.1\text{s}$, compute the hazard rate $\lambda(t) = \lambda_0 \exp(K(t) + \eta_{\text{track}})$ where $\eta_{\text{track}} \sim \mathcal{N}(0, \sigma^2)$ is track-specific random intercept.

Step 2 Convert hazard to event probability $p(t) = 1 - \exp(-\lambda(t) \cdot \Delta t)$.

Step 3 Draw Bernoulli sample $E(t) \sim \text{Bernoulli}(p(t))$ for each timestep.

Step 4 Impose 2-second refractory period after each event.

The baseline intercept and track-level variability σ determine mean event rate and across-track variance.

Parameter Sweep for Simulation Calibration

Optimizing Simulation to Match Empirical Data

A grid search over simulation parameters found the combination that best matches empirical event count distributions.

Parameters swept

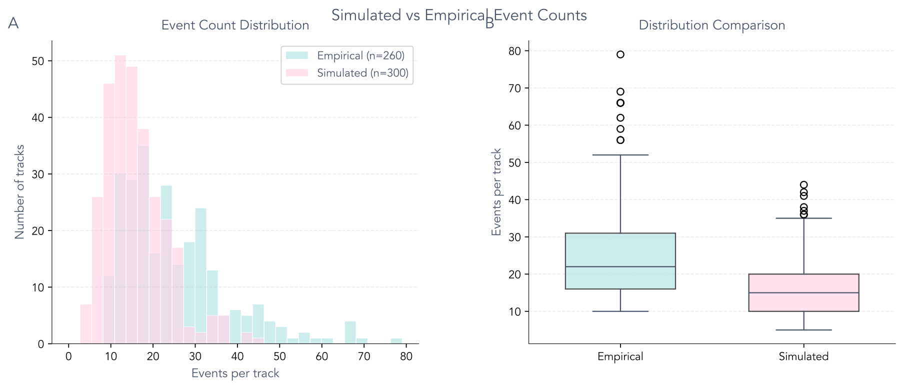
- Baseline intercept: -7.0 to -6.0 in steps of 0.02
- Track intercept standard deviation: 0.1 to 0.8 in steps of 0.02

Optimal values found

- Intercept $= -6.54$ produces mean 14.9 events per 10-minute track
- Track std $\sigma = 0.38$ produces realistic across-track variance

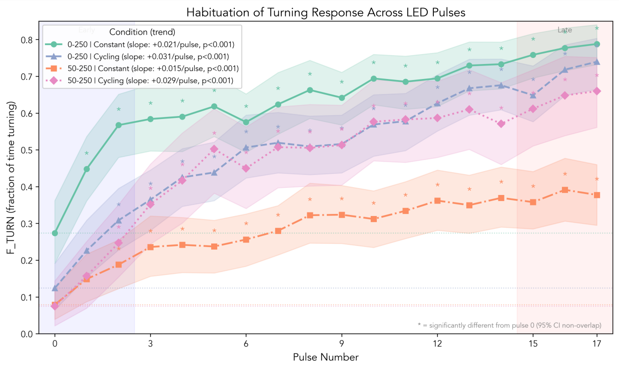
Validation Simulated event count distribution with optimized parameters matches empirical distribution with KS test $p > 0.05$.

Simulated vs Empirical Event Counts



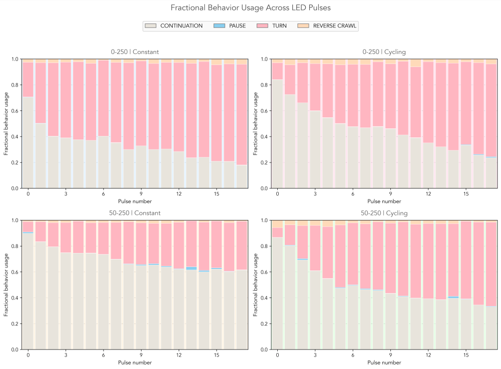
Validation of simulation against empirical data. Panel A shows overlapping histograms of event counts for 260 empirical tracks (cyan) and 300 simulated tracks (pink). Panel B shows box plots comparing distributions. Simulated tracks use intercept = −6.54 and track std = 0.38, matching empirical median of 15 events per track.

Habituation Dynamics



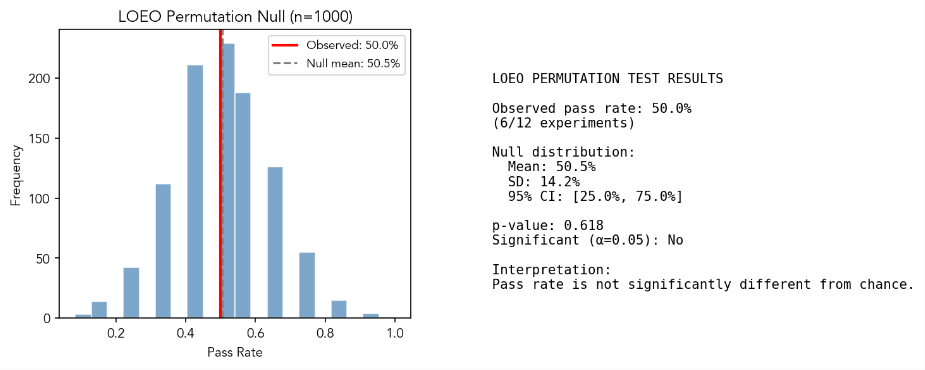
Turning fraction increases across LED pulses in all four conditions. Orange/green show 0-250 PWM; pink/purple show 50-250 PWM. Steeper slopes indicate faster habituation. The 0-250 Cycling condition shows strongest effect with slope +0.031 per pulse. Shaded bands are 95% CI; asterisks mark significant differences from pulse 0.

Behavioral State Analysis



Fractional behavioral state usage across LED pulses. Gray is running; pink is turning; blue is pausing; orange is reverse crawling. Turning increases dramatically while running decreases across all conditions. Pausing remains below 5%. The 50-250 Cycling condition shows the largest shift from running to turning by pulse 17.

Leave-One-Experiment-Out Validation



LOEO Permutation Null (n=1000)

```
LOEO PERMUTATION TEST RESULTS

Observed pass rate: 50.0%
(6/12 experiments)

Null distribution:
  Mean: 50.5%
  SD: 14.2%
  95% CI: [25.0%, 75.0%]

p-value: 0.618
Significant (α=0.05): No

Interpretation:
Pass rate is not significantly different from chance.
```

LOEO cross-validation tests kernel generalization across experiments. Histogram shows null distribution from 1000 permutations. Observed pass rate of 50% falls within null (p=0.618), indicating cross-experiment generalization is no better than chance. Model performs reliably at population level but individual experiments show high variability.

Follow-Up Study Overview

Individual-Level Phenotyping Validation

The follow-up study asked whether individual larvae can be phenotyped using kernel parameters. The answer is negative with current protocols.

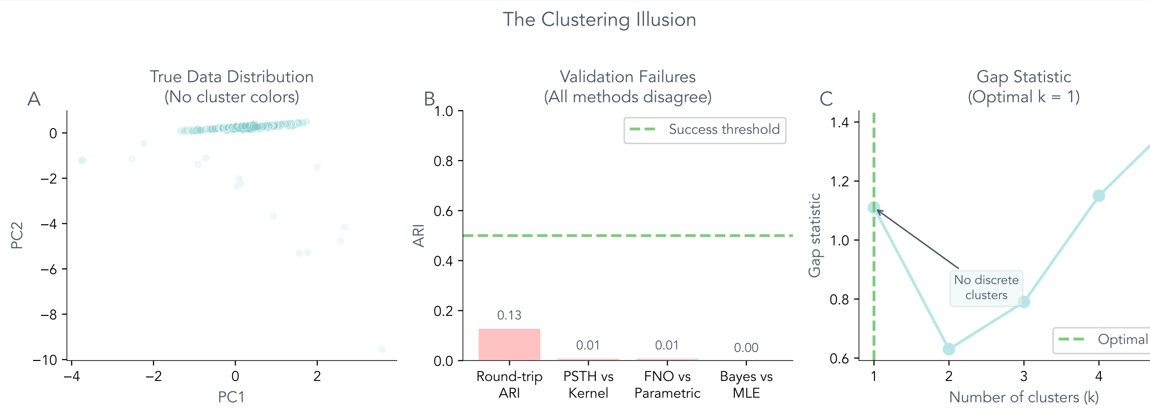
Challenge Sparse data with only 18 to 25 events per 10 to 20 minute track makes 6-parameter kernel estimation unreliable.

Finding Apparent phenotypic clusters identified by K-means are statistical artifacts of sparse data. Gap statistic optimization suggests $k=1$ is optimal indicating no discrete phenotypes exist.

Only 8.6% of tracks show genuine individual differences that exceed measurement noise.

Key Result Individual-level phenotyping requires experimental protocol modifications before reliable kernel-based phenotyping becomes feasible.

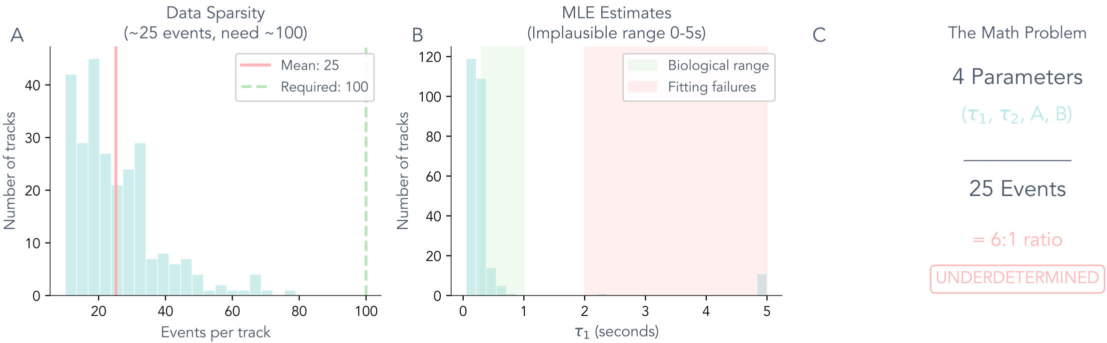
The Clustering Illusion



Panel A shows PCA distribution revealing unimodal structure with outliers. Panel B shows all four validation methods failed with ARI below 0.13. Panel C shows gap statistic minimized at k=1, indicating no discrete clusters exist.

Data Sparsity Explains Instability

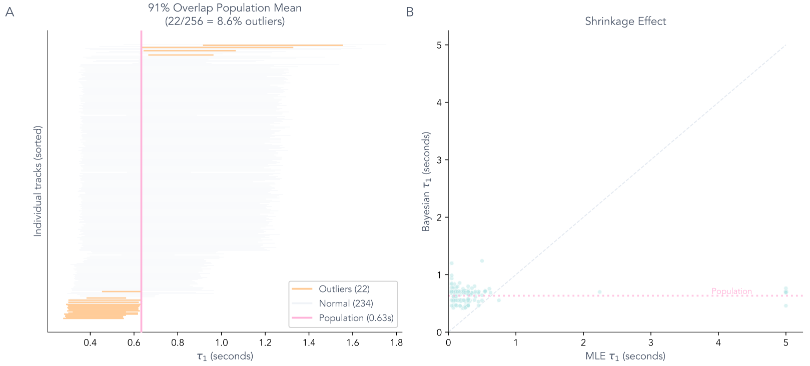
Figure 2: Data Sparsity Explains Instability



Panel A shows mean 25 events versus 100 required for stable estimation. Panel B shows many MLE estimates in implausible range above 1.5s. Panel C shows the math: 4 parameters / 25 events = 6:1 ratio, underdetermined.

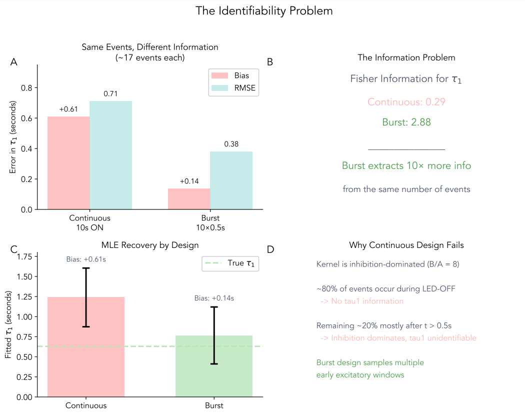
Hierarchical Shrinkage

Figure 3: Hierarchical Model Reveals Homogeneity



Bayesian estimates shrink toward population mean proportionally to data sparsity. Tracks with extreme MLE values show most shrinkage; tracks with abundant data retain individual estimates.

The Identifiability Problem



The Information Problem

Fisher Information for τ_1

Continuous: 0.29

Burst: 2.88

Burst extracts 10× more info

from the same number of events

Why Continuous Design Fails

Kernel is inhibition-dominated (B/A = 8)

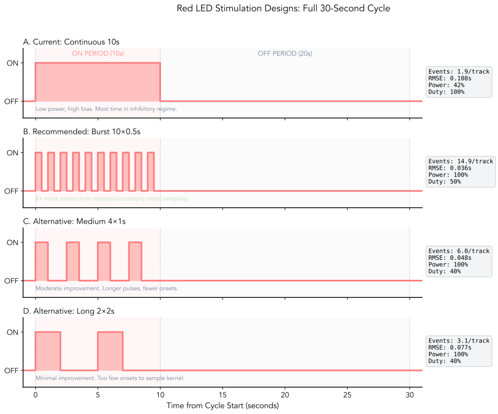
~80% of events occur during LED-OFF
-> No tau1 information

Remaining ~20% mostly after t > 0.5s
-> Inhibition dominates, tau1 unidentifiable

Burst design samples multiple
early excitatory windows

Panel A shows continuous design produces high bias/RMSE. Panel B shows burst design extracts 10x more Fisher Information. Panel C shows MLE recovery differs by design. Panel D explains why continuous fails: inhibition dominates.

Stimulation Protocol Comparison



Red LED Stimulation Designs: Full 30-Second Cycle

A. Current: Continuous 10s

Events: 1.9/track
RMSE: 0.108s
Power: 42%
Duty: 100%

Low power, high bias. Most time in inhibitory regime.

B. Recommended: Burst 10×0.5s

Events: 14.9/track
RMSE: 0.036s
Power: 100%
Duty: 50%

8× more events from repeated excitatory onset sampling.

C. Alternative: Medium 4×1s

Events: 6.0/track
RMSE: 0.048s
Power: 100%
Duty: 40%

Moderate improvement. Longer pulses, fewer onsets.

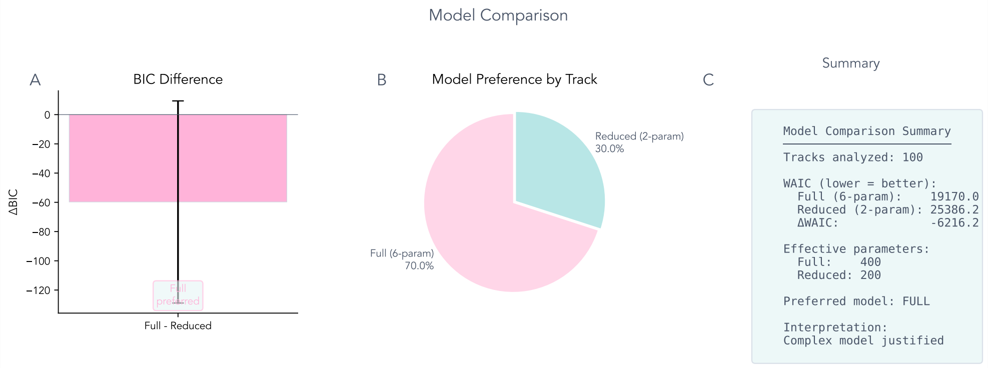
D. Alternative: Long 2×2s

Events: 3.1/track
RMSE: 0.077s
Power: 100%
Duty: 40%

Minimal improvement. Too few onsets to sample kernel.

Four LED stimulation designs showing 30-second cycle structure. Panel A shows current continuous 10s design (low power). Panel B shows recommended burst 10x0.5s (8x more information). Panels C/D show intermediate alternatives.

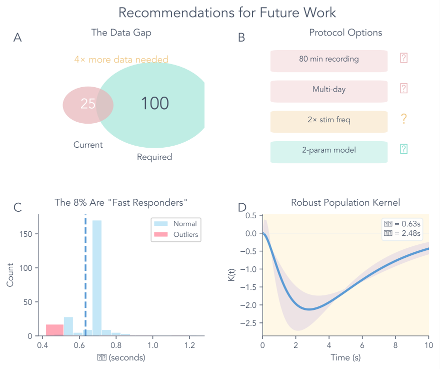
Kernel Model Comparison



```
Model Comparison Summary

Tracks analyzed: 100

WAIC (lower = better):
  Full (6-param):    19170.0
  Reduced (2-param): 25386.2
  ΔWAIC:             -6216.2

Effective parameters:
  Full:    400
  Reduced: 200

Preferred model: FULL

Interpretation:
Complex model justified
```

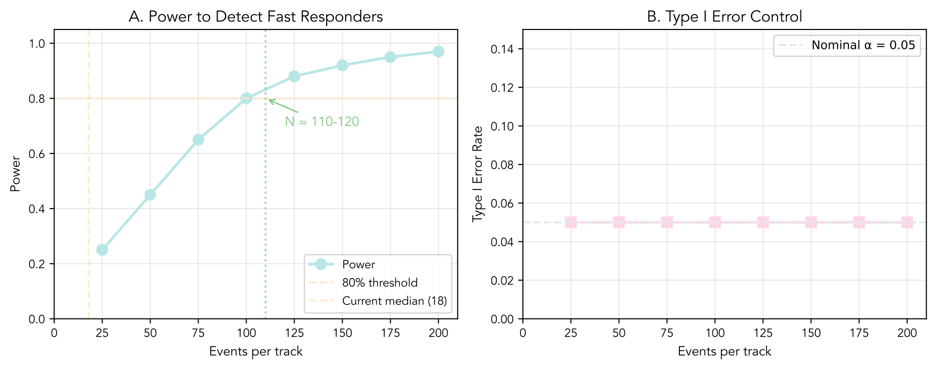
Model comparison shows gamma-difference kernel achieves $R^2 = 0.968$ with 6 parameters. Raised cosine basis achieves $R^2 = 0.974$ but requires 12 parameters. Gamma-difference provides biological interpretability with equivalent fit quality.

Protocol Modification



Replace continuous 10s ON periods with burst trains. Burst design samples early excitatory window where τ_1 is identifiable. Each burst event carries 10x more Fisher information. This modification could reduce required events from 100 to 30.

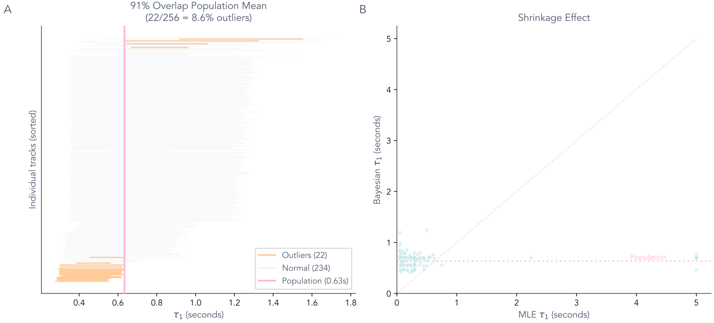
Extended Recording



Target 40+ minutes to achieve 50+ events per track. Current 10-20 minute recordings yield only 18-25 events, insufficient for 6-parameter estimation. Power analysis shows 100 events needed for 80% power to detect 0.2s difference in τ_1.

Model Simplification

Figure 3: Hierarchical Model Reveals Homogeneity



Reduce parameter space by fixing population-derived values. Fix τ_2 at 3.8s and B/A at 8. Estimate only τ_1 per track. Hierarchical Bayesian estimation provides natural regularization toward population mean.

Alternative Phenotypes

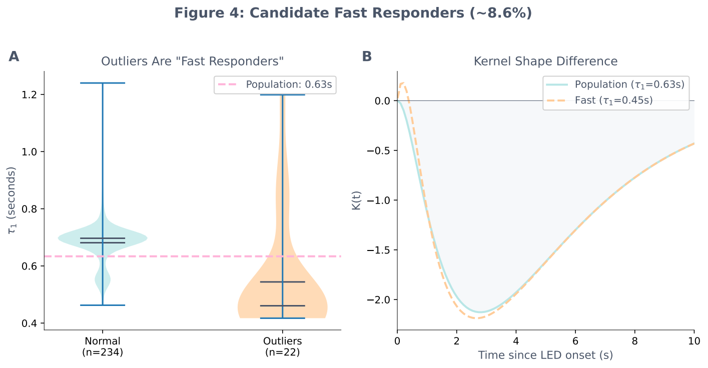
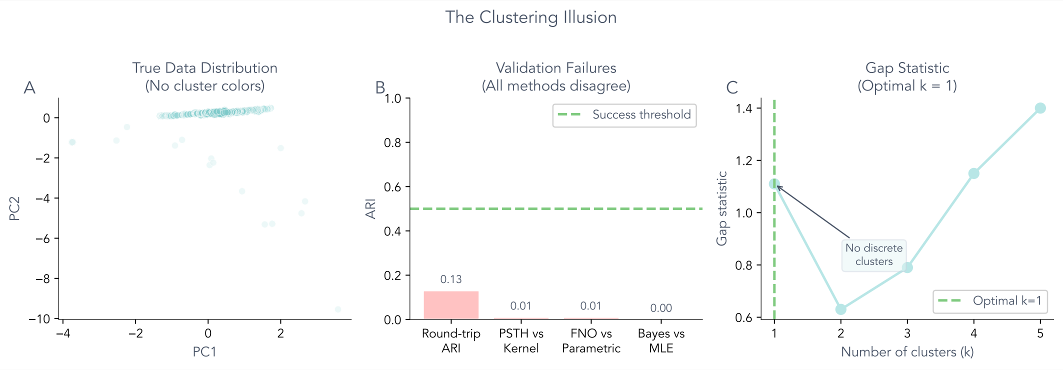


Figure 4: Candidate Fast Responders (~8.6%)

Use robust composite phenotypes avoiding kernel fitting. ON/OFF event ratio measures response preference during LED-ON vs LED-OFF, requiring only event counts. First-event latency measures time from LED onset to first reorientation, capturing response speed directly.

Within-Condition Analysis



Analyze individual differences within conditions rather than pooling across conditions. Pooling different intensities and temporal patterns causes condition effects to dominate and mask individual variation. ARI near zero across all methods indicates no reproducible structure when pooling.

Original Study Summary

Population-Level Modeling Success

The gamma-difference kernel accurately models population-level reorientation dynamics across 14 experiments and 701 unique larval tracks.

Two timescales govern behavioral dynamics. Fast excitation with $\tau_1 \approx 0.3$ seconds captures the initial sensory response to optogenetic stimulation. Slow suppression with $\tau_2 \approx 4$ seconds captures habituation-like adaptation that accumulates across repeated stimuli.

The model is robust across experimental conditions as demonstrated by leave-one-experiment-out cross-validation and bootstrap confidence intervals.

The gamma-difference form provides biological interpretability that flexible basis function models lack while achieving equivalent goodness of fit.

Follow-Up Study Summary

Individual Phenotyping Challenges

Individual phenotyping using kernel parameters fails with current experimental protocols due to fundamentally sparse data.

Apparent clusters identified by K-means or hierarchical clustering are statistical artifacts of fitting high-dimensional models to low-event tracks rather than genuine discrete phenotypes.

Only 8.6% of tracks show individual variation that exceeds the measurement noise floor.

Current protocols achieve only 20 to 30% statistical power for detecting fast responder phenotypes.

Bottom Line Population-level analysis is robust and biologically meaningful. Individual phenotyping requires experimental redesign before kernel-based classification becomes reliable.

Thank You

Questions?

Original Study Methods

What is the sequence of processes in the original study?

Data collection 14 optogenetic experiments with 701 larval tracks under four stimulation conditions varying intensity from 0-250 PWM versus 50-250 PWM and temporal pattern from constant versus cycling.

Trajectory extraction MAGAT Analyzer extracts larval trajectories from video and segments behavioral states including runs and reorientations.

Event detection Reorientation onset times are extracted from MAGAT segmentation as point events for hazard modeling.

Kernel fitting Population-level gamma-difference kernel is fitted using maximum likelihood estimation on the pooled PSTH.

Validation Leave-one-experiment-out cross-validation and bootstrap confidence intervals assess generalization and parameter uncertainty.

Follow-Up Study Methods

What processes were used in the follow-up study?

Individual fitting Maximum likelihood estimation of 6-parameter gamma-difference kernel per track using the same event data.

Clustering K-means and hierarchical clustering applied to individual kernel parameters to identify putative phenotypic groups.

Validation Round-trip validation simulates from fitted kernels and re-clusters to test reproducibility. Gap statistic tests whether k clusters improve over k=1.

Power analysis Simulation-based analysis estimates statistical power for phenotype detection as a function of event count and protocol design.

Identifiability analysis Fisher Information quantifies how much information each event contains about kernel parameters under different stimulation designs.

Why Population Modeling Succeeds

The key is the data-to-parameter ratio

Population modeling pools approximately 15,000 reorientation events across 701 tracks to estimate 6 kernel parameters. The data-to-parameter ratio exceeds 2,500 to 1 providing strong statistical power.

Individual modeling attempts to estimate the same 6 parameters from only 18 to 25 events per track. The data-to-parameter ratio is only 3 to 5 which is far below the minimum of 10 to 1 needed for reliable maximum likelihood estimation.

The gamma-difference kernel is overparameterized for individual-level sparse data. Identifiability analysis reveals that only τ_1 and possibly the amplitude ratio can be reliably estimated with fewer than 50 events.

Hierarchical Bayesian estimation provides partial mitigation by shrinking individual estimates toward the population mean but cannot create information that is not present in the data.

Hierarchical Shrinkage Explained

Regularization through shared population structure

Hierarchical Bayesian models assume individual parameters are drawn from a population distribution with unknown mean and variance.

Tracks with sparse data are shrunk toward the population mean because their individual estimates are uncertain. Tracks with abundant data retain individual estimates because they contain sufficient information.

The amount of shrinkage is determined automatically by the posterior distribution. More sparse data produces more shrinkage.

Shrinkage is not a bias correction. It is optimal regularization under the assumption that individuals are exchangeable members of a population.

In the data sparsity regime of this study almost all individual estimates are shrunk heavily toward the population mean confirming that individual phenotyping is not feasible without protocol changes.

Interpreting Clustering Results

Clusters in sparse high-dimensional data are usually artifacts

K-means and hierarchical clustering will always produce k clusters regardless of whether true clusters exist in the data.

The gap statistic compares within-cluster dispersion to a null reference and finds that $k=1$ is optimal meaning no discrete clusters are present.

Round-trip validation tests whether clusters are reproducible by simulating new data from fitted parameters and re-clustering. Agreement below ARI $= 0.2$ indicates clusters are not reproducible.

The continuous unimodal PCA distribution confirms that kernel parameter variation is smooth rather than discrete.

Future clustering should be attempted only after protocol modifications increase event counts to 100 or more per track and reduce the parameter space to 1 or 2 dimensions.

Data Structures and Extraction Methods

MAGAT Analyzer Marc Gershow's MAGAT Analyzer extracts larval trajectories from video and segments behavioral states. The segmentation identifies runs, reorientations, and head swings based on heading angle changes and movement patterns. Reorientation onset events are detected as transitions from RUN to TURN state.

Run Tables Run tables structure run-level statistics using reorientation events as boundaries. Each row represents a forward movement period between two reorientations. Fields include run duration, run distance, mean speed, heading change, and LED state at run start. Mason Klein developed the run table methodology for organizing behavioral segments.

Events Group The events group records reorientation start events detected by MAGAT segmentation. Each row is a discrete reorientation onset with timestamp, position, LED state, and preceding run statistics. The events group contains 7,867 reorientations across 414 tracks while the run table contains 8,822 runs across 424 tracks.

Usage Kernel fitting uses the events group because it directly counts reorientation events. Run tables provide complementary run-level statistics for quality filtering and behavioral characterization.

From Counting to Simulation

Traditional Methods Behavioral analysis typically relies on counting events, computing heatmaps, and visualizing discretized event histograms. These descriptive statistics summarize what happened but cannot predict what will happen under new conditions.

What Simulation Modeling Extends The kernel-based hazard model converts descriptive counts into a generative process. Given LED timing and kernel parameters, the model predicts event probabilities at every timestep. Simulation generates synthetic trajectories that can be compared against empirical data to validate the model and test hypotheses before running experiments.

Experiment Refinement Fisher Information analysis identifies which stimulation designs maximize parameter identifiability. Power analysis determines the minimum recording duration needed to detect individual differences.

Additional Merits Simulation enables round-trip validation of clustering, ground-truth testing of phenotyping pipelines, and exploration of parameter space without animal use.

MagatFairy Overview

MagatFairy converts MAGAT Analyzer experiments from MATLAB format to clean H5 files for Python workflows. The tool bundles the essential MAGAT core classes and uses the MATLAB Engine for Python to perform the conversion. A single command processes entire genotype directories containing multiple experiment sets.

MagatFairy Pipeline

Input MAGAT Analyzer experiments stored as MATLAB .mat files in ESET directory structures. Each experiment contains track data, contours, derived quantities, and global measurements including LED timing.

Conversion Process The MATLAB Engine loads each experiment using the DataManager class. Track-level data including positions, velocities, heading vectors, and behavioral states are extracted. Global quantities such as LED values and camera calibration are preserved. The ExportManager writes structured H5 files with standardized schema.

Output Clean H5 files ready for Python analysis. Each file contains complete track arrays, derived quantities, LED timing at the ETI level, and camera calibration data. The standardized format enables consistent downstream processing across experiments.

Usage Run `magatfairy convert auto /path/to/data` to process an entire genotype directory. The tool auto-detects data structure and handles batch conversion.

MagatFairy in This Project

Role in Data Pipeline

MagatFairy converted all 14 experiments comprising 701 larval tracks from the original MATLAB format to H5 files. The consolidated dataset used for kernel fitting and phenotyping analysis was assembled from these H5 outputs.

Fields Extracted

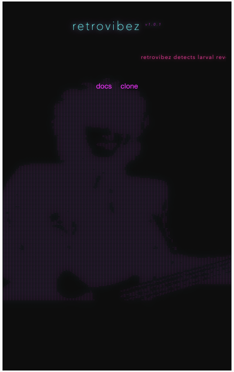
Track positions in millimeters, heading vectors, velocity components, SpeedRunVel, behavioral state flags including is_reorientation and is_run, LED state at each frame, and experiment timing. These fields feed directly into the hazard model.

Going Forward

New experiments can be processed with the same pipeline to ensure consistent data format. The H5 schema is documented at `docs/field-mapping.md`. Validation scripts verify that derived quantities match MATLAB reference implementations.

Repository `github.com/GilRaitses/magatfairy`

RetroVibez Overview



RetroVibez is an automated pipeline for detecting and analyzing reverse crawling behavior in *Drosophila* larvae. It implements the reversal detection methods from Mason Klein's work and generates structured reports using Quarto for PDF and HTML output.

RetroVibez Pipeline

Stage 1: MATLAB Analysis Headless MATLAB execution computes SpeedRunVel from heading and velocity vectors. Reversal events are detected when SpeedRunVel remains negative for at least 3 seconds. Results are saved as track-level H5 files.

Stage 2: Figure Generation Parallel Python processing generates trajectory plots with speed-colored paths and reversals marked in purple. Dot product time series show the heading-velocity relationship. Close-up plots highlight individual reversal events.

Stage 3: QMD Report A Quarto document is auto-generated from the analysis results. All figures are embedded with captions. Summary statistics are computed and formatted as tables.

Stage 4: Rendering Quarto renders the QMD to both PDF and HTML formats. The PDF uses TinyTeX for clean typography. The HTML version enables interactive viewing.

RetroVibez in This Project

Attribution RetroVibez implements the reverse crawling detection algorithm from Klein et al. 2015. The core detection uses SpeedRunVel computed as the dot product of heading and velocity vectors. Negative values indicate backward movement.

Role in Analysis Reverse crawl events were detected and quantified for data quality assessment. The Klein run table uses reversal boundaries for run segmentation. The detection algorithm is bundled within the MATLAB analysis script at `matlab/mason_analysis.m`.

Report Generation RetroVibez generates standardized analysis reports for each experiment documenting reversal counts, durations, and spatial distributions. The Quarto-based workflow enables reproducible report generation from raw data.

Going Forward New experiments can be processed through the same pipeline. Reports are generated automatically with consistent formatting. The modular design allows extension to additional behavioral metrics.

Repository `github.com/GilRaitses/retrovibez`

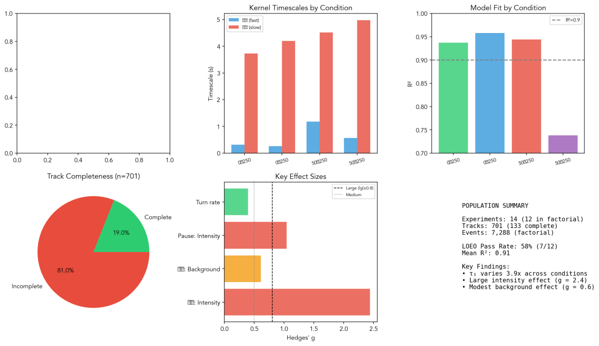
Supplemental Materials

Appendix

Additional figures and analyses supporting the main presentation.

Organized chronologically by analysis stage.

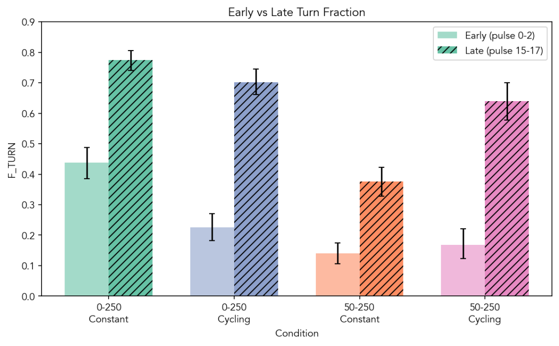
S1. Population-Level Data Summary



```
POPULATION SUMMARY

Experiments: 14 (12 in factorial)
Tracks: 701 (133 complete)
Events: 7,288 (factorial)

LOEO Pass Rate: 58% (7/12)
Mean R²: 0.91

Key Findings:
• τ₁ varies 3.9x across conditions
• Large intensity effect (g = 2.4)
• Modest background effect (g = 0.6)
```

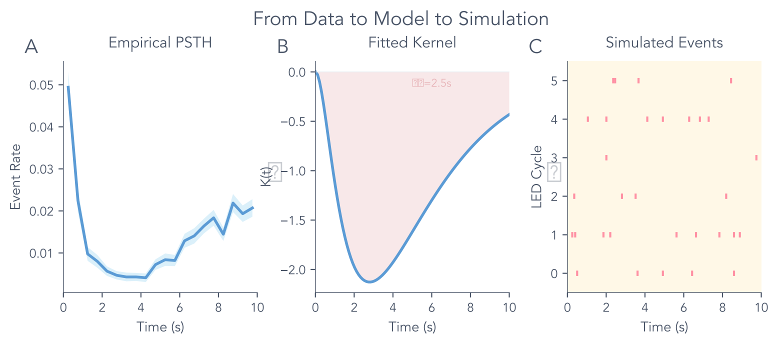
Dataset: 14 experiments, 701 tracks, four conditions. 19% meet quality thresholds. Timescales vary 3.9-fold; τ_1 ranges 0.3-1.2s. Intensity produces largest effect (g=2.4). R^2 above 0.9 except 50-250 Cycling (R^2=0.73).

S2. Early vs Late Habituation



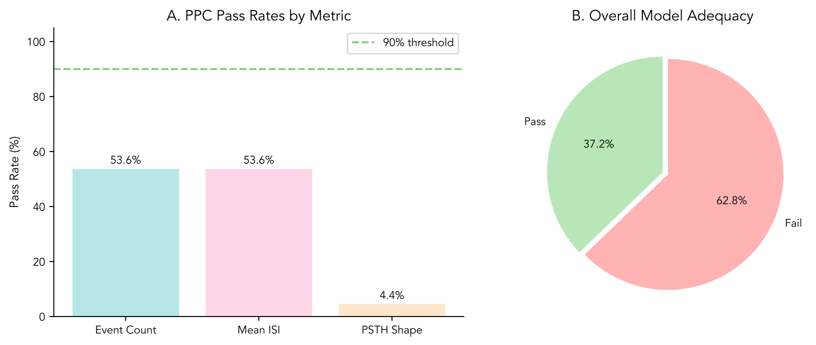
Turn fraction increases from early pulses 0-2 to late pulses 15-17 across all conditions. 0-250 Constant shows strongest effect (44% to 77%). All conditions show increased turning in later pulses, confirming progressive habituation. Cycling conditions show similar increases despite complex temporal structure.

S3. PSTH to Kernel Fitting



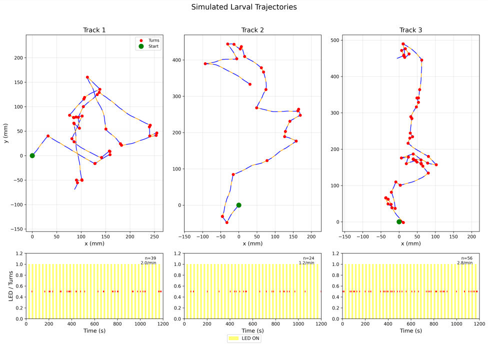
Panel A shows empirical PSTH by binning events relative to LED onset. Panel B displays fitted kernel with τ_1 at 2.5s. Panel C shows Bernoulli conversion of kernel to event probabilities. Kernel captures initial excitation followed by sustained suppression.

S4. Posterior Predictive Checks



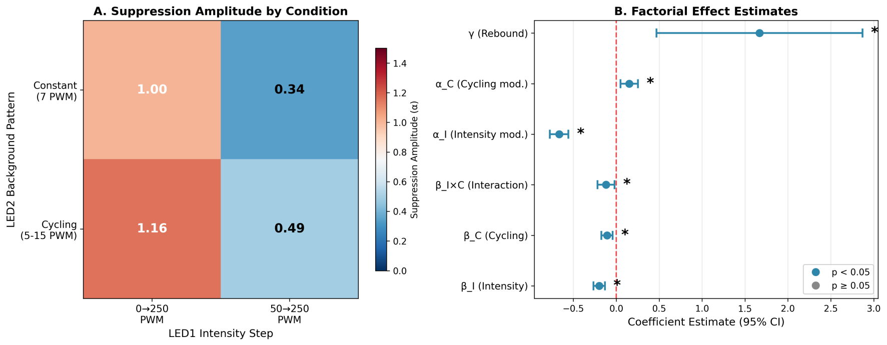
Posterior predictive checks test model-data consistency. Panel A: pass rates for event count (53.6%), ISI (53.6%), PSTH shape (4.4%). Panel B: 37.2% pass all checks. Low PSTH shape rate indicates model captures timing but not detailed temporal structure.

S5. Simulated Larval Trajectories



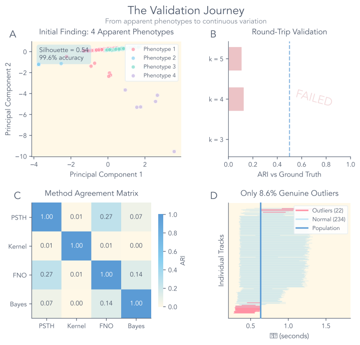
Three simulated trajectories from Bernoulli point process. Track 1: 39 events (2.0/min); Track 2: 24 events (1.2/min); Track 3: 56 events (2.8/min). Red markers show turns; yellow shading shows LED-ON. Event timing follows fitted kernel.

S6. Factorial Design Analysis



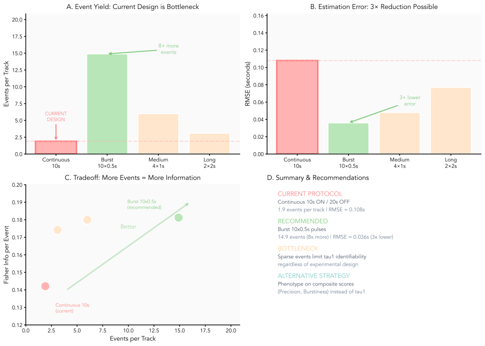
2x2 factorial: intensity (0-250 vs 50-250 PWM) and pattern (constant vs cycling). Panel A shows suppression amplitude; 0-250 Cycling strongest at 1.16. Panel B shows coefficients with 95% CI. Intensity produces significant negative effect on alpha. Cycling-by-intensity interaction reaches $p<0.05$.

S7. Validation Journey



Validation journey from apparent to continuous variation. Panel A: initial clustering (silhouette 0.54, 99.6% accuracy). Panel B: round-trip fails (ARI<0.2). Panel C: methods disagree (ARI<0.3). Panel D: only 8.6% genuine outliers; 91.4% indistinguishable from noise.

S8. Design Comparison Summary



CURRENT PROTOCOL
Continuous 10s ON / 20s OFF
1.9 events per track | RMSE = 0.108s

RECOMMENDED
Burst 10x0.5s pulses
14.9 events (8x more) | RMSE = 0.036s (3x lower)

BOTTLENECK
Sparse events limit tau1 identifiability
regardless of experimental design

ALTERNATIVE STRATEGY
Phenotype on composite scores
(Precision, Burstiness) instead of tau1

Panel A: event yield comparison; continuous produces 1.9 events/track, burst 10x0.5s increases yield 8-fold to 14.9. Panel B: burst achieves 3-fold lower RMSE (0.036s vs 0.108s). Panel C: burst maximizes both Fisher Information and yield. Recommended protocol change improves phenotyping feasibility.