

Sensorimotor Habituation in *Drosophila* Larvae

Population-Level Modeling and Individual Phenotyping Validation

Gil Raitses

Bhattacharya Lab General Meeting

Syracuse University

Original Study Overview

Population-Level Sensorimotor Habituation Model

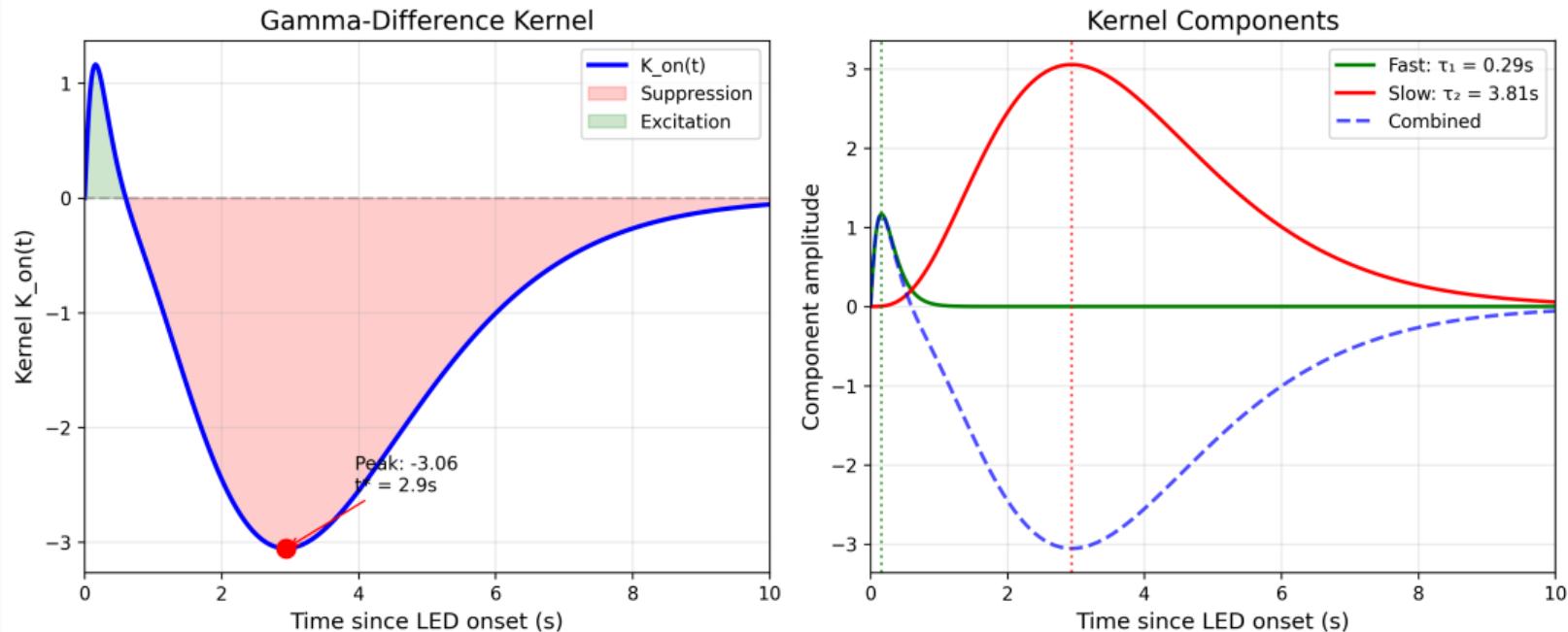
Larval reorientation behavior follows a gamma-difference kernel with two distinct timescales that capture the dynamics of sensory response and adaptation.

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. The left panel shows the combined kernel with peak excitation at 0.3 seconds followed by suppression reaching minimum at 3 seconds. The right panel decomposes this into fast excitatory gamma with $\tau_1 = 0.29s$ and slow

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 occurring at that latency across all trials. The empirical PSTH is computed directly from data using histogram binning without assuming any functional form.

Parametric PSTH The Bernoulli event probability is derived from the fitted kernel. At each timestep, event probability $p(t) = 1 - \exp(-\lambda(t) \cdot \Delta t)$ where $\lambda(t) = \lambda_0 \exp(K(t))$ is the hazard rate modulated by kernel $K(t)$. The parametric PSTH evaluates this probability function at each time point.

Model Fitting The kernel is fitted by maximizing the likelihood that empirical event times arise from the Bernoulli process. The empirical PSTH is used for visualization and validation. The 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

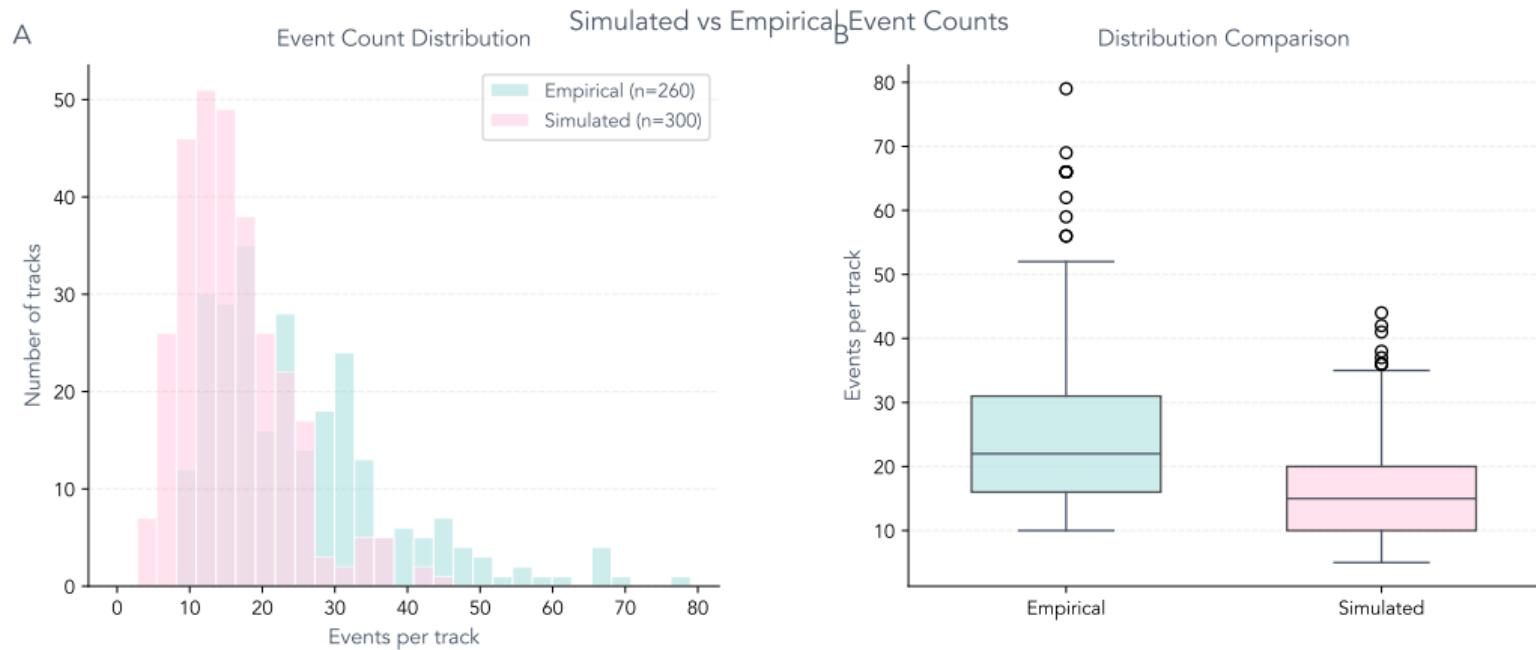
- Baseline intercept: -7.0 to -6.0 in steps of 0.02
- Track intercept standard deviation: 0.1 to 0.8 in steps of 0.02

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

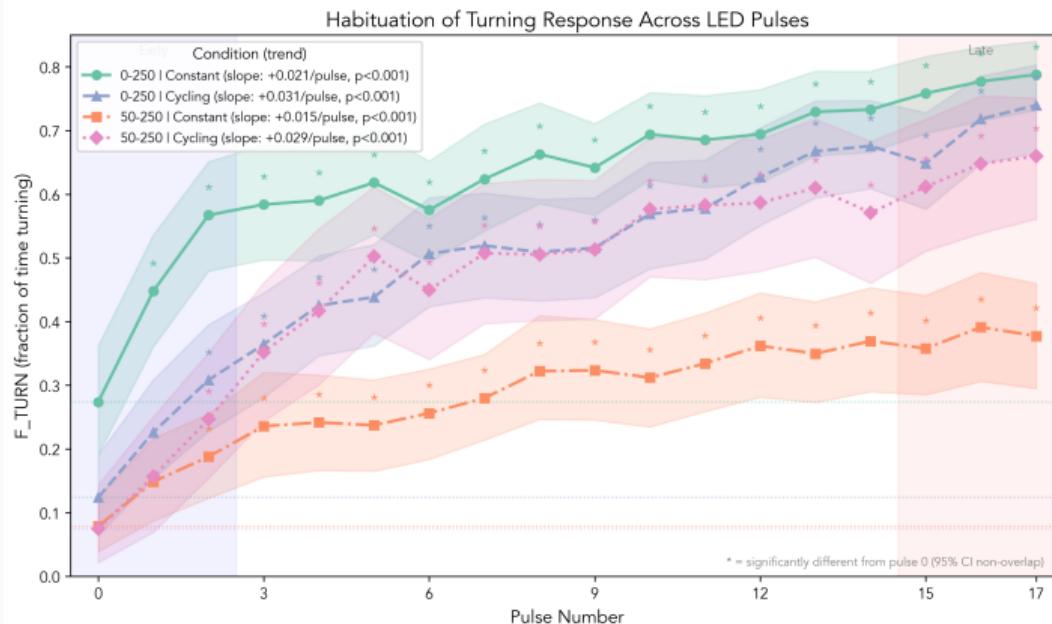
Validation Simulated event count distribution with optimized parameters matches

Simulated vs Empirical Event Counts



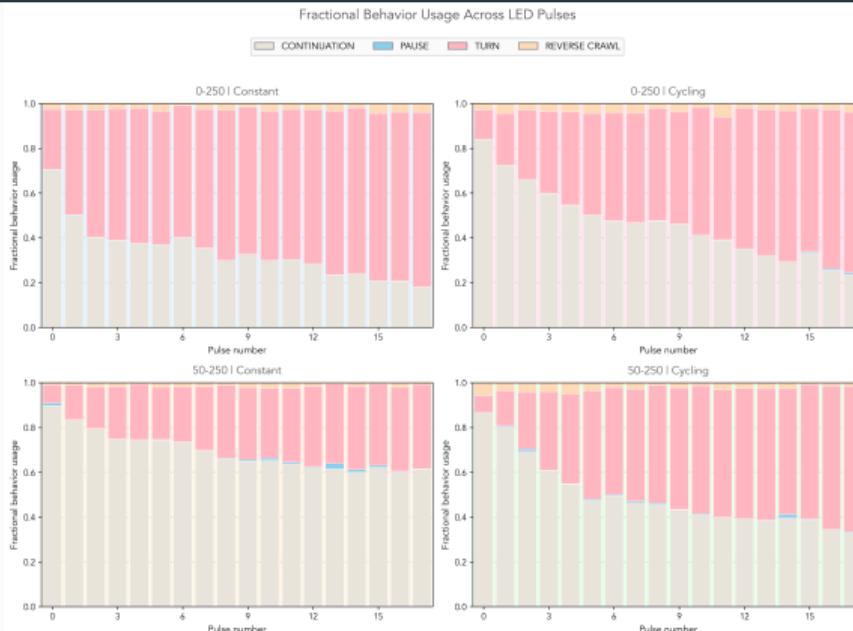
Validation of the simulation model against empirical data. Panel A shows overlapping histograms of event counts per track for 260 empirical tracks in cyan and 300 simulated tracks in pink. Panel B shows box plots comparing the distributions. The simulated tracks were generated using the fitted population kernel with intercept = -6.54 and track std = 0.38 and

Habituation Dynamics



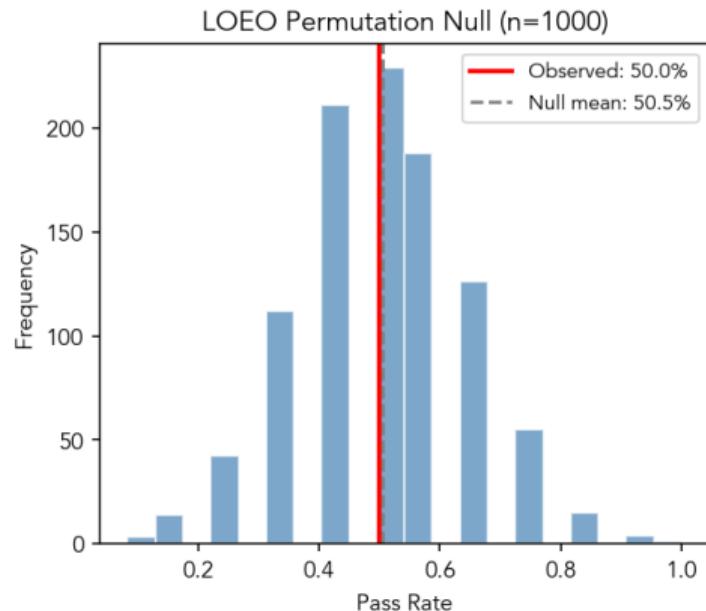
Fraction of time spent turning increases across successive LED pulses in all four experimental conditions. Orange and green lines show 0-250 PWM conditions while pink and purple show 50-250 PWM conditions. Steeper slopes indicate faster habituation. The 0-250 Cycling condition shows the strongest habituation effect with slope +0.031 per pulse. Shaded bands

Behavioral State Analysis



Fractional behavioral state usage across LED pulses for all four experimental conditions. Gray indicates forward running. Pink indicates turning behavior. Blue indicates pausing. Orange indicates reverse crawling. Turning fraction increases dramatically across pulses in all conditions while running fraction decreases. Pausing remains below 5% throughout. The

Leave-One-Experiment-Out Validation



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 ($\alpha=0.05$): No

Interpretation:
Pass rate is not significantly different from chance.

LOEO cross-validation assesses whether kernel parameters generalize across experiments. The histogram shows the null distribution generated from 1000 permutations of experiment labels. The observed pass rate of 50% falls within the null distribution with $p=0.618$ indicating that cross-experiment generalization is no better than chance. Model performs reliably at the

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.

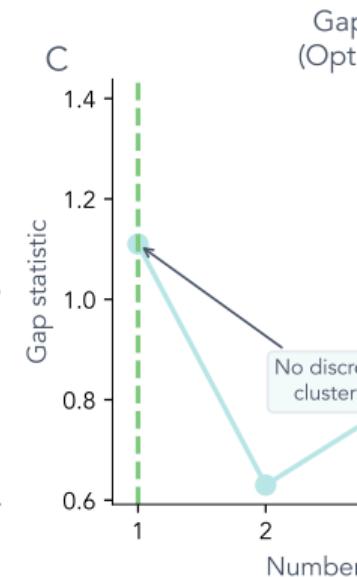
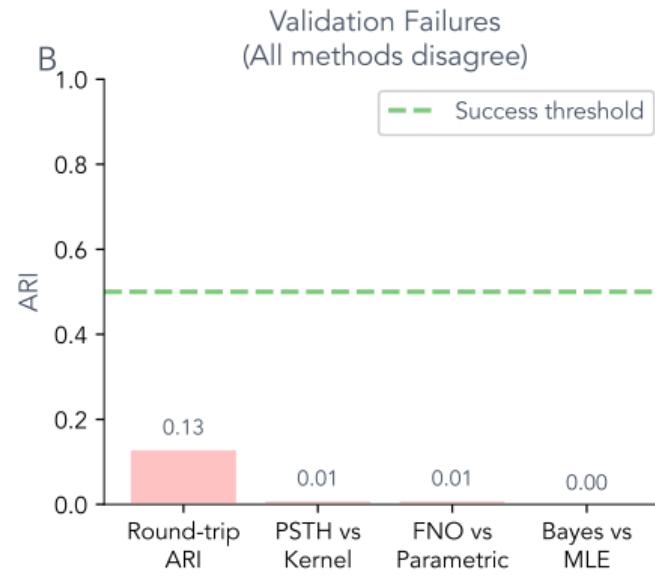
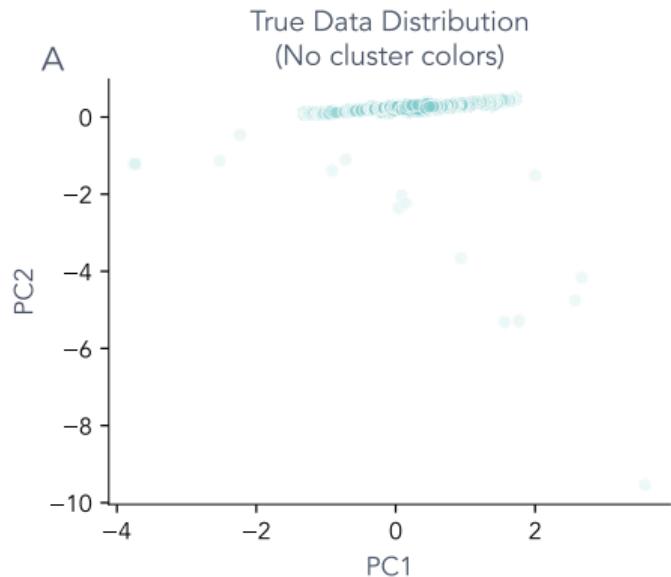
Challenge Sparse data with only 18 to 25 events per 10 to 20 minute track makes 6-parameter kernel estimation unreliable.

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.

The Clustering Illusion

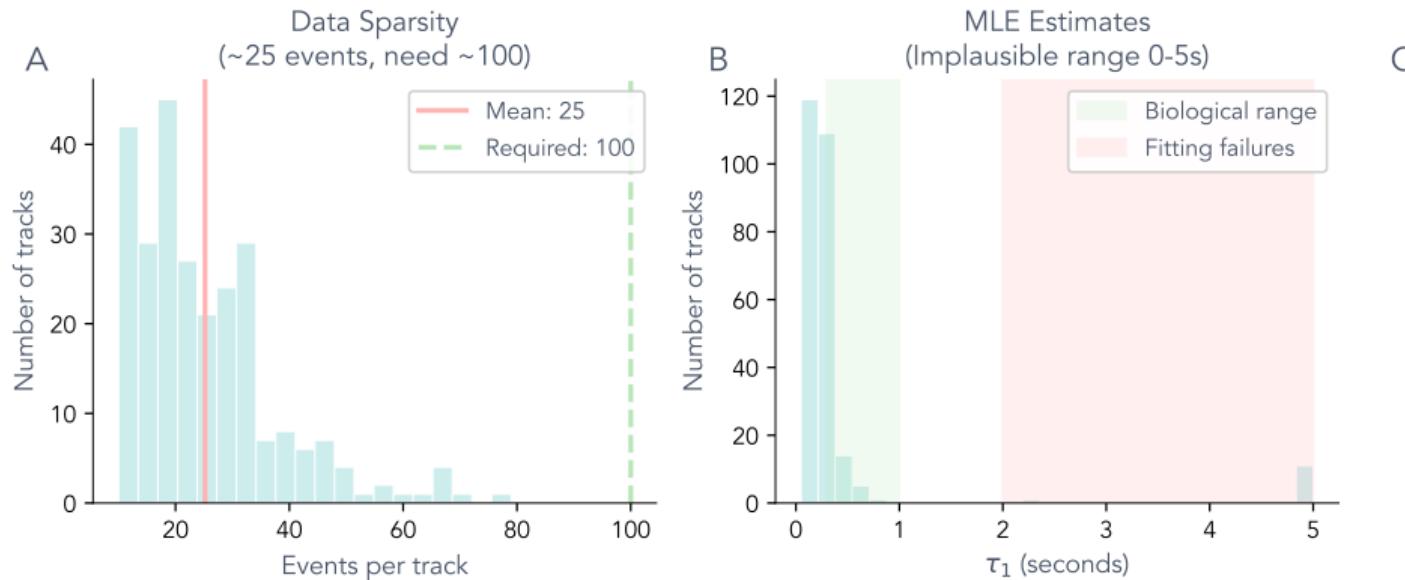
The Clustering Illusion



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

Data Sparsity Explains Instability

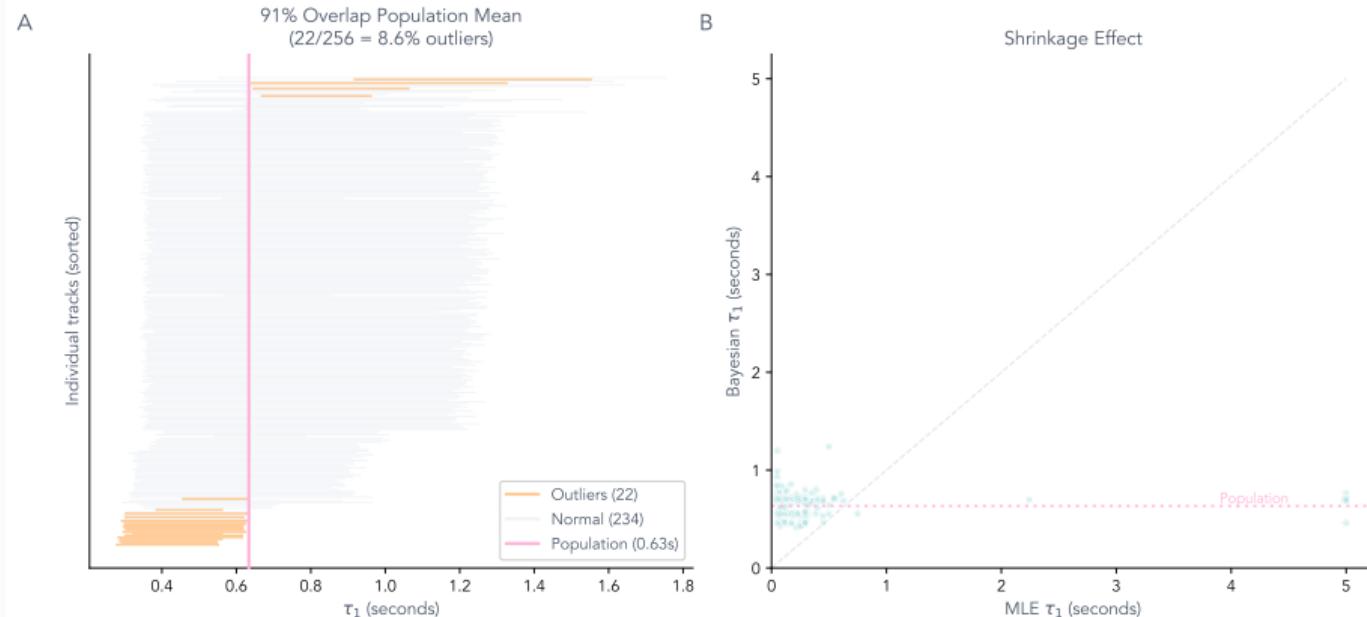
Figure 2: Data Sparsity Explains Instability



Panel A shows mean 25 events per track versus the 100 required for stable estimation. Panel B shows many MLE estimates fall in the implausible range above 1.5 seconds. Panel C shows the math problem: 4 parameters divided by 25 events equals a 6:1 ratio which is underdetermined.

Hierarchical Shrinkage

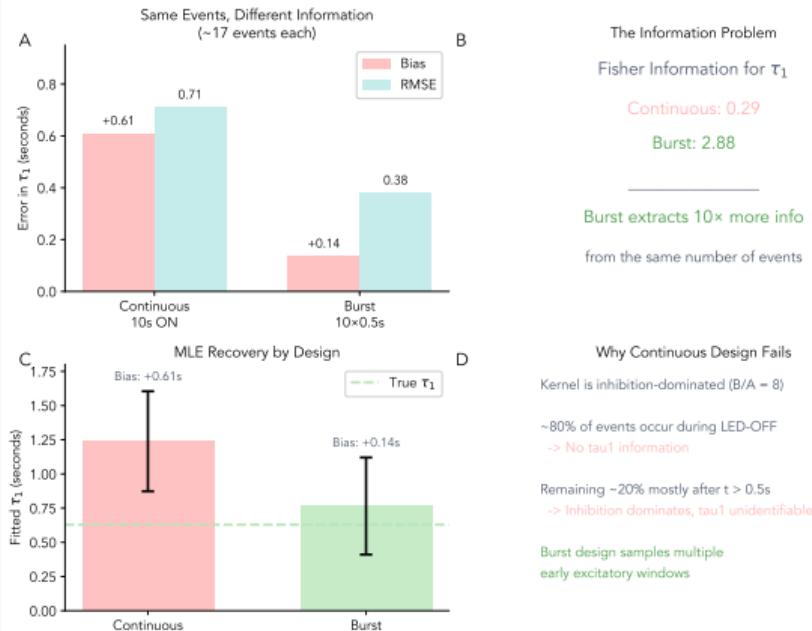
Figure 3: Hierarchical Model Reveals Homogeneity



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

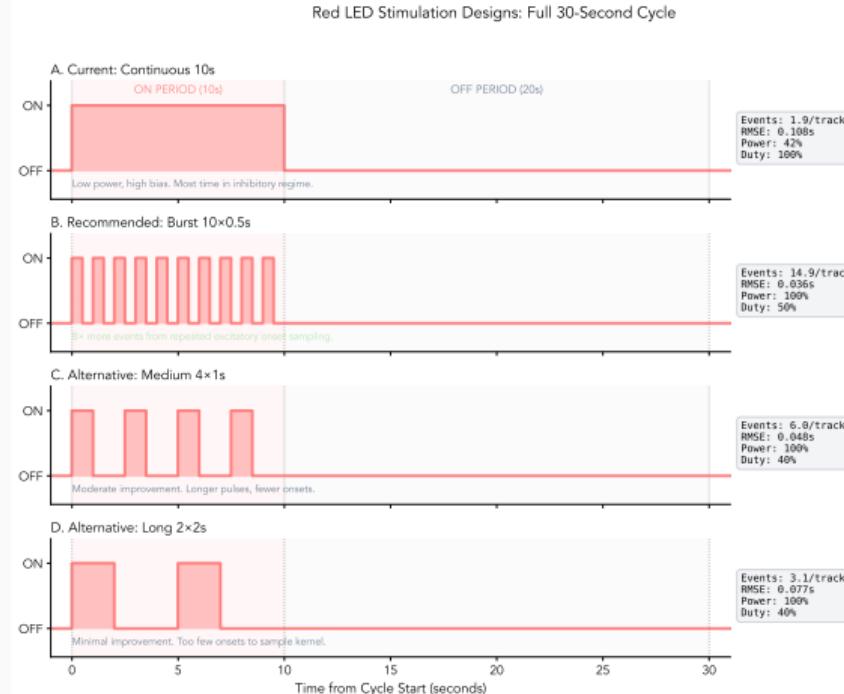
The Identifiability Problem

The Identifiability Problem



Panel A shows continuous design produces high bias and 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.

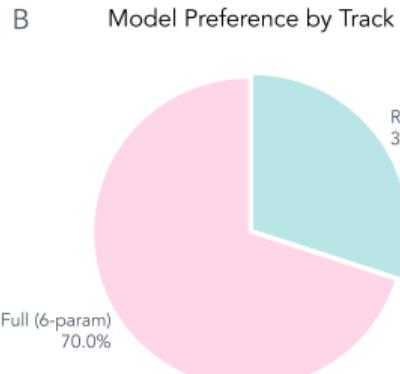
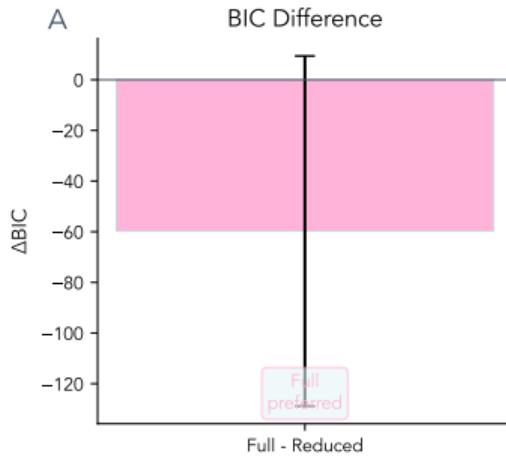
Stimulation Protocol Comparison



Four LED stimulation designs showing the 30-second cycle structure. Panel A shows current continuous 10s design with low power. Panel B shows recommended burst 10 \times 0.5s with 8x more information. Panels C and D show intermediate alternatives.

Kernel Model Comparison

Model Comparison



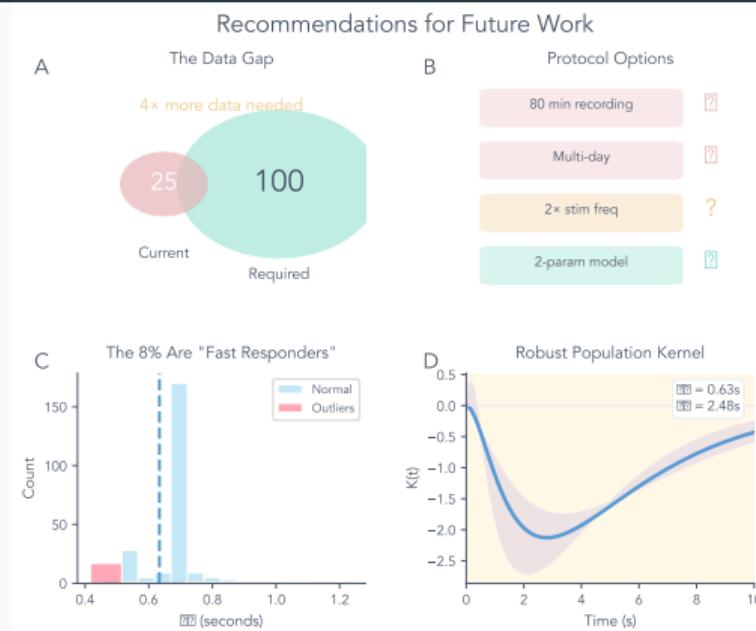
C

Summary

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	

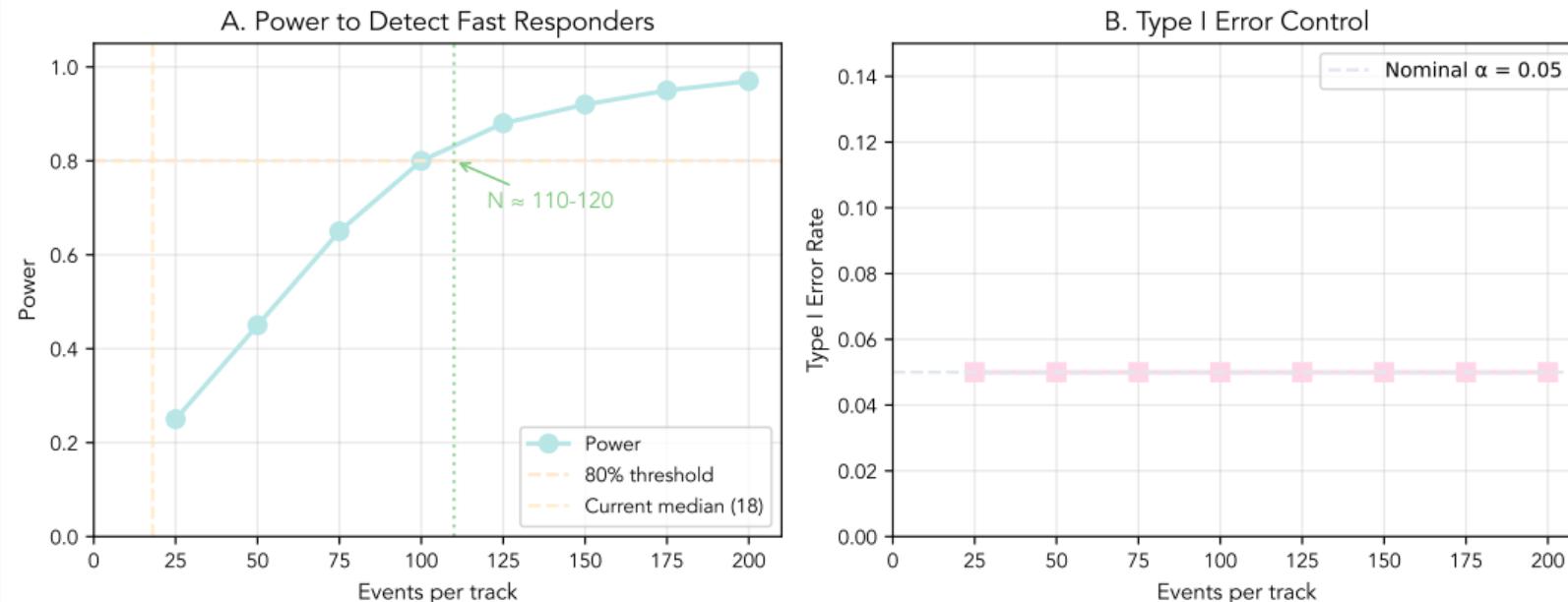
Model comparison shows the gamma-difference kernel achieves $R^2 = 0.968$ with 6 parameters. The raised cosine basis achieves $R^2 = 0.974$ but requires 12 parameters. The gamma-difference form provides biological interpretability with equivalent fit quality.

Protocol Modification



Replace continuous 10-second ON periods with burst trains. Burst design samples the early excitatory window where τ_1 is identifiable. Each burst event carries 10 times more Fisher information than continuous events. This modification alone could reduce the number of events required for reliable estimation from 100 to 30.

Extended Recording

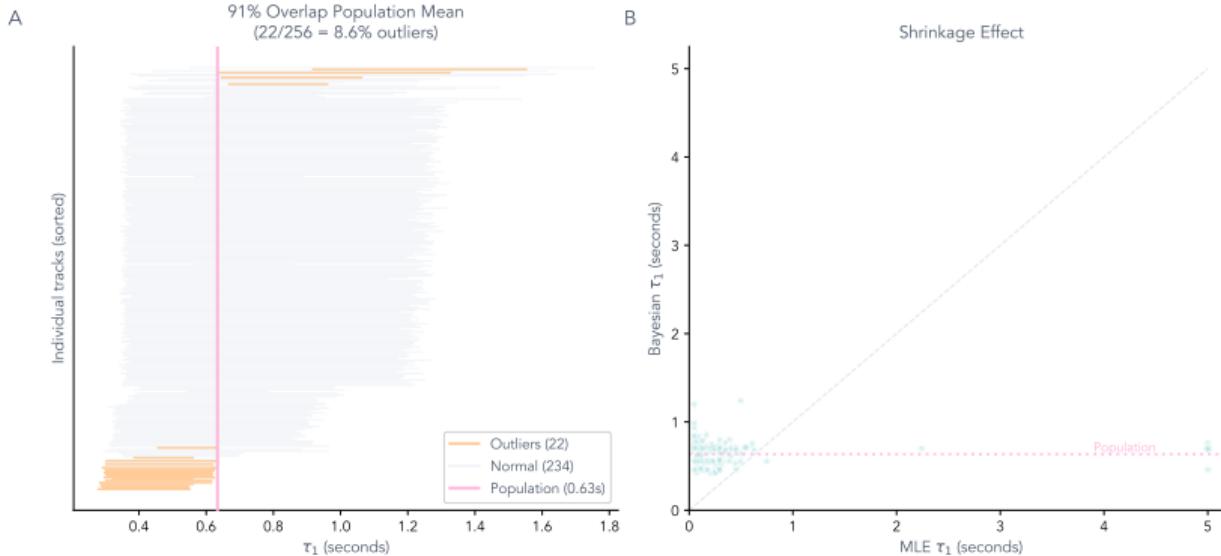


Target 40 minutes or more of recording to achieve at least 50 reorientation events per track.

Current 10 to 20 minute recordings yield only 18 to 25 events which is insufficient for 6-parameter estimation. Power analysis indicates that 100 events are required for 80% power to detect a 0.2 second difference in τ_1 at the individual level.

Model Simplification

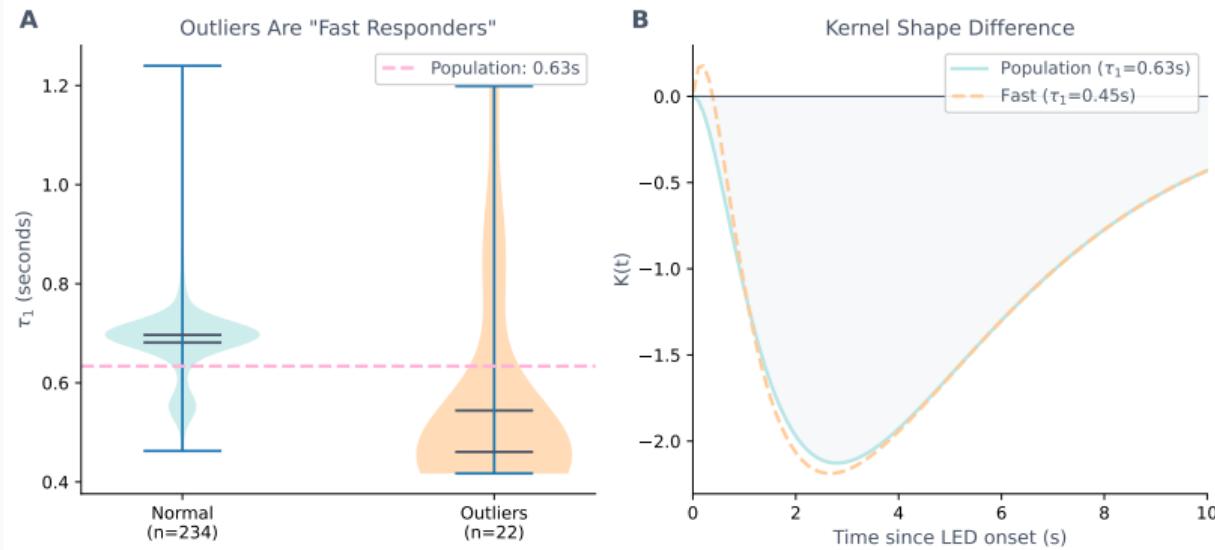
Figure 3: Hierarchical Model Reveals Homogeneity



Reduce the parameter space by fixing population-derived parameters. Fix τ_2 at the population estimate of 3.8 seconds and fix the amplitude ratio B/A at the population value of 8. Estimate only the fast timescale τ_1 per individual track. Hierarchical Bayesian estimation provides natural regularization toward the population mean.

Alternative Phenotypes

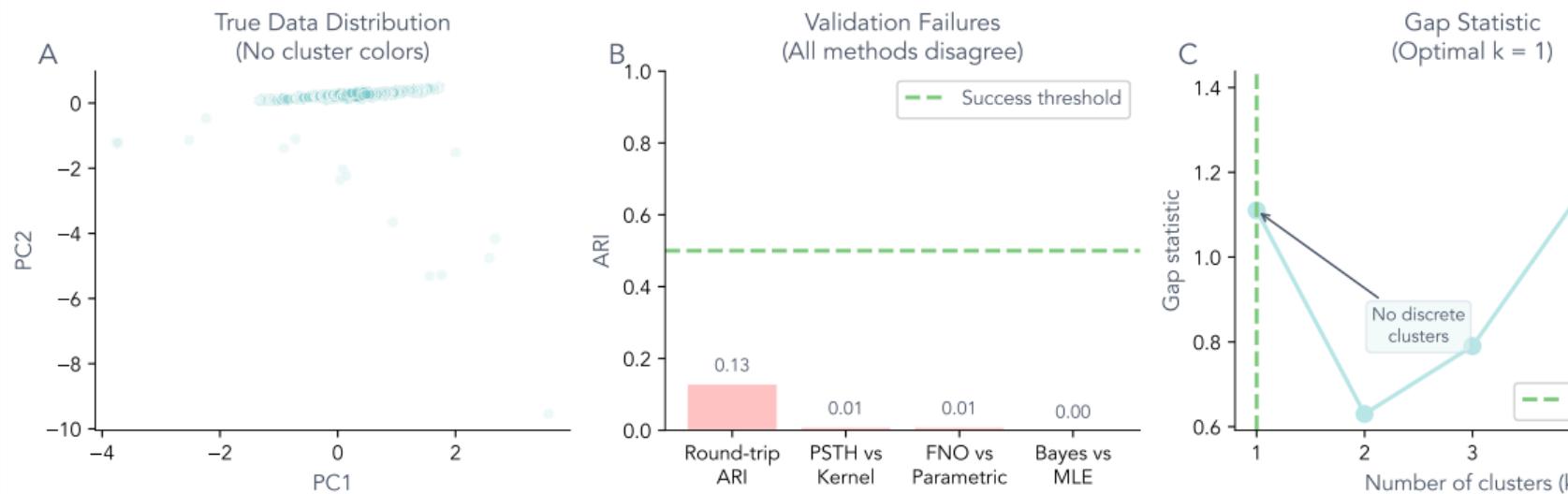
Figure 4: Candidate Fast Responders (~8.6%)



Use robust composite phenotypes that avoid kernel fitting entirely. The ON/OFF event ratio measures whether larvae respond preferentially during LED-ON versus LED-OFF periods and requires only event counts not full kernel estimation. First-event latency measures the time from LED onset to first reorientation and captures response speed directly.

Within-Condition Analysis

The Clustering Illusion



Analyze individual differences within experimental conditions rather than pooling across conditions. When data from different stimulation intensities and temporal patterns are pooled the condition effects dominate and mask genuine individual variation. The ARI near zero across all validation 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.

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

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

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.

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 $\text{ARI} = 0.2$ indicates clusters are not reproducible.

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

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. These discrete events serve as the dependent variable for hazard modeling.

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 into analyzable units.

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

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

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.

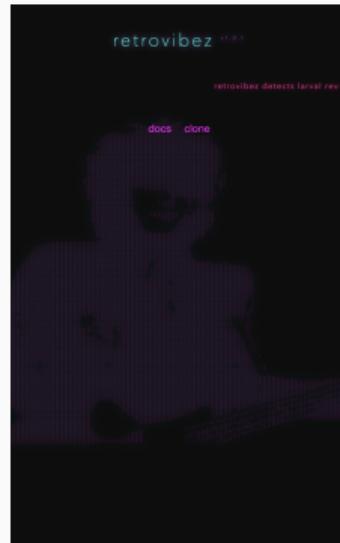
Fields Extracted

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.

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.

RetroVibez in This Project

Attribution

RetroVibez implements the reverse crawling detection algorithm described in 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. These reports document reversal counts, durations, and spatial distributions. The Quarto-based workflow enables reproducible report generation from raw data.