Relativistic Ecological Dynamics: An Empirical Investigation of its Geometric Properties

Sebastián Vallejo
Servicio Académico Universitario y Centro de Estudios Paleontológicos
(SAUCE-P), Universidad de la República, Uruguay
Email: svallejo@grupomas.software
Richard A. Fariña
Facultad de Ciencias, Universidad de la República, Uruguay
Servicio Académico Universitario y Centro de Estudios Paleontológicos
(SAUCE-P), Universidad de la República, Uruguay
Email: dogor@netgate.com.uy

Abstract

The foundational models of population dynamics, such as those by Lotka and Volterra, presuppose a static, Euclidean phase space where interactions are governed by fixed forces. The theory of Biorelativity challenges this, positing that dynamics are better described as geodesics on a manifold whose geometry is actively shaped by the system's state and external forcings. This study gives operational form to this concept by framing the choice between classical and biorelativistic models as an empirical question about the underlying geometry of ecological interactions. A model tournament was conducted across three predator-prey datasets, confronting classical models with a key formalisation of Biorelativity: a climate-driven ecological clock that modulates the metric of the phase space. While simple Euclidean models proved sufficient for systems dominated by internal dynamics, the climate-driven ecological clock (BR-τZ) was the decisive winner for the externally forced Isle Royale system. This result provides strong empirical evidence that a geometric, non-static description of ecological dynamics is not only more accurate but also more parsimonious, vindicating the core tenets of Biorelativity.

1 Theoretical Framework: From Flat Space to a Dynamic Manifold

1.1 The Classical View: A Euclidean Arena

The pioneering work of Lotka [14] and Volterra [24] and subsequent foundational analyses by authors like May [16] and Murdoch [18], established the traditionestablished the tradition of modelling ecological dynamics within a static, Euclidean phase space. In this Newtonian view, populations are treated as points, and their interactions (predation, competition) are modelled as eco-evolutionary forces governed by fixed parameters, deflecting trajectories from otherwise linear paths. This framework implicitly assumes that the underlying space of ecological possibilities is flat, unchanging and absolute. While powerful, this view relegates environmental effects to external perturbations acting upon the system, rather than being intrinsic to the fabric of the interaction space itself.

However, this classical Newtonian viewpoint is not the sole conceptual possibility. An alternative, inspired by geometric theories of physics, posits that the ecological phase space is not merely a static arena but rather a dynamic structure in its own right – mathematically, a differentiable manifold $\mathcal{M}$. In this geometric framework, the interactions and environmental influences typically modelled as forces manifest instead as the curvature and evolving metric g_{ij} of the manifold itself. The trajectories of populations are then hypothesised to follow geodesics, the paths of least 'effort' through this structured space.

This conceptual shift from forces to geometry is not without precedent. For several decades, pioneering work, particularly by Antonelli and colleagues, has explored the geometrisation of ecological and evolutionary dynamics. This research has demonstrated that classical population models, including variants of the Lotka-Volterra equations, can be rigorously reformulated as geodesic flows, often utilising the even more general framework of Finsler geometry, where the 'metric' of the space can depend on the direction of travel [2].

1.2 The Biorelativity Paradigm: A Geometric Re-interpretation

The theory of Biorelativity, as proposed by Fariña [8], offers a paradigm shift analogous to Einstein's move from Newtonian physics to general relativity. It reframes the phase space itself as a dynamic, high-dimensional differentiable manifold [26]. There are several core concepts that provide the foundation for this view, and they are as follows.

Formally, the phase space is conceived as an n-dimensional differentiable manifold, denoted $\mathcal{M}$, a topological space that locally resembles Euclidean space, allowing calculus to be performed. Coordinate systems on $\mathcal{M}$ are realised through local charts (mappings from subsets of $\mathcal{M}$ to open sets in R^n), and the transition maps between overlapping charts are required to be smooth (infinitely differentiable in this context).

Hypervolumes and Projections: An organism or a population is not a simple, dimensionless point but a complex hypervolume existing in a high-dimensional eco-morphological spacetime. This hypervolume is defined by a vast number of axes representing genetic, morphological, behavioural and physiological traits, as well as abiotic factors (salinity, rainfall, substrate). The population dynamics we observe in a typical two-dimensional phase space (prey vs. predator) are therefore a lower-dimensional projection of this much richer reality. The act of modelling is an act of projection, which involves collapsing this complexity into a simplified representation.

This new "point" is an abstraction whose meaning is contingent upon the subspace chosen for the projection. For example, a projection onto an average morphological subspace yields a point representing the archetypal individual (the centroid of the hypervolume, defined by traits such as average body size or diet). An analysis at this level compares interactions between these archetypes. Alternatively, a projection onto an ecological functional subspace yields a point representing the species' niche role (e.g., "apex predator," "large herbivore"). Here, the analysis concerns the structure of the ecosystem itself, independent of the specific taxa filling those roles. Finally, a projection onto a phylogenetic subspace can represent an entire clade, enabling the study of macroevolutionary trajectories. The models employed herein, therefore, do not capture the complete state, but a shadow of it, projected onto the chosen subspace of analysis.

Curvature as Interaction: In this framework, interactions do not manifest as forces but as the curvature of the phase space itself, in direct analogy to General Relativity where mass-energy tells spacetime how to curve and curved spacetime tells mass how to move. The ecological mass-energy of a species is not its physical weight, but its capacity to modify its environment and the relationships within it. An entity with high ecological mass-energy drastically alters the landscape of possibilities and probabilities for others, effectively deforming the rules of the game. The prey’s trajectory, in response, follows a geodesic, the straightest possible path through this curved geometry.

This curvature can be induced in multiple ways. An ecosystem engineer like a beaver (*Castor canadensis*) imparts extreme curvature by transforming a lotic (running water) system into a lentic (still water) one through dam construction. This action blocks the geodesics of riverine fish while creating new, low-resistance paths for amphibians and aquatic insects. An apex predator like the now extinct sabretoothed cat (*Smilodon populator*) generated a “landscape of fear” that curved the behavioural space of its prey (and other species), triggering cascading trophic effects. An invasive species like buffelgrass (*Pennisetum ciliare*) curves the environmental axes by introducing a frequent fire cycle, creating conditions lethal to native flora. In each case, a biological entity redefines reality for others by altering the structure of the environment itself.

Mathematically, the intrinsic curvature of the manifold is fully encoded in the Riemann curvature tensor, R^{k}_{lij} (often denoted K^{k}_{lij} in some notations). This type (1, 3) tensor is constructed from the connection coefficients Γ^{k}_{ij} and their first derivatives. Its non-vanishing components quantify the extent to which the manifold deviates locally from Euclidean geometry.

This connection is foundational. The field of information geometry, formalised by Amari, provides a robust mathematical framework for treating the space of probability distributions as a statistical manifold equipped with a Riemannian metric (the FIM). This framework has been successfully applied to foundational models in theoretical biology, such as the Wright-Fisher model of population genetics, where the manifold's curvature describes the effects of selection and drift [3]. Our hypothesis—that the empirical covariance structure Σ^{-1} approximates the metric g_{ij}—is therefore a direct ecological application of this established principle.

In the biorelativistic interpretation, non-zero curvature directly reflects the intensity and nature of ecological interactions, governing phenomena such as the convergence or divergence of nearby system trajectories.

The Metric Tensor, Mahalanobis Distance, and Fisher Information: The geometry of this space is defined by a metric tensor [15], g_{ij} [21]. This can be conceptualised by relating it to the covariance structure of the system's state variables. The Mahalanobis distance, $d^2_M(X_1, X_2) = (X_1 - X_2)^T \Sigma^{-1}(X_1 - X_2)$, naturally accounts for the covariance (Σ) between variables. It is thus posited that the metric tensor is approximated by the inverse of the covariance matrix, $g_{ij} \approx (\Sigma^{-1})_{ij}$.

The geometry of this manifold $\mathcal{M}$ is defined locally by a metric tensor, g_{ij}, which is formally a symmetric, non-degenerate tensor field of type (0, 2). It equips the manifold with a notion of infinitesimal distance, $ds^2 = g_{ij}dx^i dx^j$ (where summation over repeated indices is implied according to the Einstein convention). In the Biorelativity framework, g_{ij} is posited to be dynamic, potentially depending on the system state X and external forcings Z, i.e., $g_{ij}(X, Z)$. The empirical approach herein utilises the inverse covariance matrix Σ_{ij}^{-1} or the Fisher Information Metric (FIM) as operational estimators for the components of g_{ij} under specific distributional assumptions, noting that the FIM intrinsically defines a Riemannian geometry on the relevant statistical manifold.

A more rigorous foundation for the phase space geometry is furnished by the Fisher Information Metric (FIM). The FIM naturally endows the space of probability distributions pertinent to the ecological system, a statistical manifold, with a Riemannian geometry. This geometric structure, intrinsic to the statistical description itself, is generally non-Euclidean; that is, the statistical manifold is typically curved. As detailed in Amari & Nagaoka [1], for specific cases such as the multivariate Gaussian distributions implicitly assumed in least-squares fitting procedures, the FIM simplifies and becomes proportional to the inverse of the covariance matrix. This crucial link allows the empirically accessible Mahalanobis distance to serve as a practical approximation to the underlying FIM. Significantly, this implies that the very statistical structure inherent in ecological data may naturally define a curved phase space, irrespective of specific interaction mechanisms. This perspective strongly resonates with the core proposal of Biorelativity, suggesting that a non-flat geometry is not merely a consequence of biological interactions but potentially an emergent property of the system's statistical description itself.

A metric tensor g_{ij} naturally determines a unique affine connection, known as the Levi-Civita connection. Its coefficients, the Christoffel symbols Γ_{ij}^{k} , depend algebraically on the first partial derivatives of the metric tensor components. This connection defines the rules for parallel transport of vectors along curves on the manifold and is crucial for defining geodesics, the paths of 'inertial' motion within the curved space.

Explanatory Box: The FIM - Mahalanobis Approximation. The assertion that the metric tensor (g_{ij}) can be approximated by the inverse covariance matrix (Σ^{-1})—the core of the Mahalanobis distance—is a specific and powerful result from information geometry. Here are its formal conditions and limitations [1].

The formal condition for this equivalence rests upon the properties of a multivariate Gaussian (normal) distribution.

The Fisher Information Metric (FIM) defines the geometry of a statistical manifold (the space of all possible probability distributions of a certain family). The Mahalanobis distance, $d^2_M(x,\mu) = (x - \mu)^T \Sigma^{-1}(x - \mu)$, defines a distance within a data space. The two concepts become equivalent under a precise condition:

The Fisher-Rao distance (the geodesic distance derived from the FIM) is exactly equal to the Mahalanobis distance only within the statistical manifold of Gaussian distributions that share a common, fixed covariance matrix Σ_0 , and where only their mean vectors μ are parameters that vary.

In this specific case, the manifold of parameters μ is "flat" (Euclidean), and the FIM is constant and equal to the inverse of the shared covariance matrix, $g_{ij} = \Sigma_{ij}^{-1}$

This is the precise assumption invoked when using standard least-squares fitting, which assumes Gaussian noise with a stable covariance.

Mini-Example: When the Approximation Fails. The approximation $g \approx \Sigma^{-1}$ fails when the underlying statistical manifold is not this simple, constant-covariance Gaussian space.

Conceptual Failure: Non-Linearity. Imagine our predator-prey data does not form a simple elliptical cloud, but rather a curved, banana-like shape in the phase space (e.g., due to a non-linear relationship where $\approx X^2$). The covariance matrix only captures linear correlations. It will (incorrectly) describe this banana shape as a single, large, tilted ellipse. The Mahalanobis distance will measure distance relative to the centre of this flawed ellipse. The true FIM would be derived from the actual probability distribution of this banana shape. This creates a curved statistical manifold.

In this scenario, Σ^{-1} is a poor approximation of the true metric g. The covariance matrix fails to capture the true geometric structure, whereas the FIM, by definition, is that structure. This limitation also applies to data that is strongly skewed or heteroskedastic (where variance is not constant).

Hierarchies as Geometric Constraints: Ecological systems are hierarchical, and the concept of a "point" in this hyperspace represents a biological entity at a specific instant, with its complexity defining the hierarchical level.

- **Level 1: The Individual Organism.** The "point" is a single, concrete living being (e.g., an individual jaguar, *Panthera onca*, in the Pantanal, with coordinates for its exact body mass, hunger level and geographic position at an instant, as well as local weather conditions). Note that, according to the aim of the analysis, the level 1 could have been chosen below the organism, for instance, the cell or the genes.
- **Level 2: The Population.** The "point" is an abstraction representing the aggregated state of a group (e.g., the jaguar population of the North Pantanal, with coordinates for density, age distribution, genetic diversity and abiotic factors). This point acts as the "centre of mass" for all its constituent individuals.
- **Level 3: The Community.** The "point" is a higher-level abstraction for the set of all interacting populations (e.g., the vertebrate community of the North Pantanal, with coordinates for total biomass, species richness and energy flow in those particular weather general conditions).

Within this framework, a species is not a single point, as it inherently contains variation. A species is therefore a hypervolume (or cloud of points) within this hyperspace, formed by the totality of all its individual organisms. This volume is dynamic: its centre represents the archetypal individual, its extent represents the species' variability and its movement through time represents evolution. Speciation can be visualised as the fragmentation of this hypervolume and extinction as its collapse.

Biorelativity postulates that external, non-biological macro-level variables (e.g., climate) do not exert direct deterministic forces on biological dynamics. Instead, these variables define the geometric constraints and the boundary conditions of the system's phase space, within which intrinsic biological interactions unfold.

Consequently, a change in an external variable (such as climate) alters the topology or boundaries of this phase space—that is, the shape of the "pitch"—but does not modify the fundamental parameters or the governing equations that define the biological interactions occurring within it—that is, the "rules of the game".

1.3 Reichenbach's Dilemma and the Ecological Clock

This leads to a fundamental problem articulated by the philosopher of science Hans Reichenbach [19]. He argued that the geometry of physical space is not an a priori truth but an empirical question. A phenomenon can be described by assuming a flat geometry and postulating "universal forces", or by assuming a curved geometry where objects follow geodesics. This study operationalises this dilemma by framing the model comparison as a test between two families of hypotheses:

- H_F (The "Force" Hypothesis): Climate acts as a universal force, modifying specific parameters within a classical, flat RA model.
- H_G (The "Geometric" Hypothesis): Climate modifies the underlying metric of the phase space. A simple, scalar-field approximation of this change is tested: the ecological clock,

$$C_{eco}(Z), \text{ where } X = C_{eco}(Z) \cdot F(X, \theta)$$

The model tournament is thus presented not as a mere comparison of contingent implementations, but as a critical test between these two fundamental explanatory frameworks.

2 Methods

2.1 Datasets and Model Tournament

Three canonical predator-prey datasets were analysed: the laboratory system of *Paramecium* and *Didinium* from Gause [10]; the Hudson Bay Company records of lynx and snowshoe hare [22, 23]; and the Isle Royale wolf-moose system [25]. For the Isle Royale data, a climate index was constructed from the first principal component of local temperature anomalies.

A focused model tournament was conducted. For the Isle Royale dataset, a full suite of five mechanistic models was tested to disentangle the role of climate. For the Hudson Bay and Veilleux datasets, where no strong external drivers were measured, a core comparison between the classic Rosenzweig-MacArthur [20] and the state-dependent that includes proper ecologically relevant time (BR-τ) model was performed. All models were implemented in R using the `deSolve` [7] package for numerical integration and `minpack.lm` for fitting.

The competing hypotheses were encoded directly into the ODE function:

```
climate_val  <-  params$climate_func(t)
dY_dt <- c_conv  *  fr_term  -  (d_mort  +  params$delta_d  *
climate_val)  *  Y
```

Listing 1: R code snippet for the "Force" model (RA-climate),
where climate acts as an additive force on predator mortality

```
climate_val  <-  params$climate_func(t)
c_eco  <-  exp(params$beta0  +  params$beta1  *
climate_val)
dX_dt  <-  dX_dt  *  c_eco
dY_dt  <-  dY_dt  *  c_eco
```

Listing 2: R code snippet for the BR-τZ model
where climate acts as a multiplicative scalar on the entire vector field

2.2 Model Selection and Parameterisation

Models were fitted by minimising the sum of squared errors between observed and predicted population densities using the Levenberg-Marquardt algorithm [12] provided by the `nls.lm` function in R. This approach assumes errors are approximately Gaussian, a condition under which the Fisher Information Metric and the Mahalanobis metric converge [1]. All population time-series were normalised by their respective medians prior to fitting to ensure numerical stability and comparability of error metrics across different scales.

All parameters dependent on covariates (e.g., in Hierarchical Biorelativistic Model, HBM, and geometric ecological clock, BR-τZ, models) were implemented using log-links (e.g., $e^{(\beta_0+\beta_1 Z)}$) to ensure positivity, guaranteeing biologically plausible values. The total number of estimated parameters (k) for each model included all structural coefficients (e.g., r, K), any biorelativity coefficients (β_i), the initial population conditions (X_0, Y_0), and one additional parameter for the residual variance (σ^2). Model performance was evaluated using the Akaike Information Criterion corrected for small sample sizes (AICc), which is grounded on information theory and serves as a robust tool for selecting the model with the best expected out-of-sample performance [9]. The difference in AICc from the best-performing model (ΔAICc) and the Akaike weights (w_i) were used to rank models and quantify the relative evidence for each hypothesis.

```
logLik <- fit$metrics$logLik
n <- fit$metrics$n
k <- fit$metrics$k
AIC <- (2 * k) - (2 * logLik)
AICc <- AIC + (2 * k * (k + 1)) / (n - k - 1)
```

Listing 3: R code snippet for calculating AICc

2.3 Model Specification, Evaluation, and Validation

Model families and mechanistic baselines. In addition to the standard RA and the BR-τZ models, our analysis incorporated 'force-augmented' flat models, denoted RA(b,h | Z). These models applied climatic forcing directly to the attack rate (b) or handling time (h). We also evaluated a variant employing a Holling Type II [11] functional response. For model selection, the total parameter count (k) for each model included its structural parameters, initial conditions, and the residual variance (σ^2). We report the corrected Akaike Information Criterion (AICc), Bayesian Information Criterion (BIC), ΔAICc, and Akaike weights to compare model performance.

Out-of-sample evaluation. To assess predictive performance, we implemented a rolling-origin forecasting procedure across five folds, evaluating horizons from one to five years. Forecast accuracy was quantified using the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). For models where predictive intervals were computed, we additionally report the Continuous Ranked Probability Score (CRPS). Statistical differences in forecast accuracy between models were assessed using the Diebold–Mariano test.

Identifiability: clock versus parameter forcing. We conducted synthetic data experiments to address the structural identifiability of the geometric clock against direct parameter forcing. Ground-truth datasets were generated using two distinct processes: (a) a pure ecological clock, $c_{eco}(Z)$, and (b) direct parameter forcing, θ(Z). We then fitted models from both families to these synthetic datasets and compared their out-of-sample predictive performance, particularly under simulated shifts in the forcing variable, Z. Furthermore, we quantified the collinearity between the estimated clock effect, $\hat{c}_{eco}(Z)$, and the estimated direct forcing effect, $\hat{\theta}$(Z), to assess their empirical distinguishability.

Error structure and parameter positivity. The robustness of our findings to alternative error assumptions was examined. We report the sensitivity of the results to models incorporating Generalised Least Squares (GLS) with AR(1) residuals and to a simple state-space model accommodating observation noise [6, 12]. To ensure all biological parameters remained mechanistically plausible, positivity was strictly enforced using exponential link functions.

Scaling and non-dimensionalisation. Finally, we present a non-dimensional form of the system to clarify its fundamental scaling properties. This adimensional analysis elucidates precisely which dimensionless groups are modulated by the ecological clock's temporal scaling and which remain invariant, thereby facilitating a more robust comparison of the underlying dynamics across disparate datasets.

3 Results

The model tournament revealed a clear, context-dependent hierarchy. For Isle Royale, the climate-driven ecological clock (BR-τZ) was the undisputed winning model (Akaike weight = 1.00), dramatically outperforming the classical RA and the force-based RA-climate models (Table 1). In stark contrast, for both the Hudson Bay and Veilleux datasets, the simplest mechanistic model, RA, was the clear winner (Table 2).

Table 1: Model Selection Results for the Isle Royale Dataset. The winning model is highlighted in bold.

Model	k	AICc	ΔAICc	w_i
BR_τZ	**12**	**-79.82**	**0.00**	**1.00**
RA_climate	11	-68.93	10.89	0.00
HBM	14	79.59	159.41	0.00
RA	10	37.21	117.03	0.00
BR_τ	12	35.74	115.56	0.00

Table 2: Model Selection Summaries for Baseline Datasets.

Dataset	Model	AICc	ΔAICc	w_i
Hudson Bay	**RA**	**37.21**	**0.00**	**1.00**
	BR_τ	45.85	8.64	0.01

Veilleux	**RA**	**-9.46**	**0.00**	**1.00**
	BR_τ	49.33	58.79	0.00

4 Discussion

4.1 The Empirical Resolution of Reichenbach's Dilemma

The empirical results presented resolve Reichenbach's dilemma for the specific ecological systems studied. We contrast two families of models under parsimonious instantiations: force-augmented flat dynamics (H_F) and time-warped geometric dynamics (H_G). Our empirical preference for H_G in the externally forced Isle Royale system is predictive–epistemic, based on its superior performance according to AICc which proxies expected out-of-sample predictive loss; it is not presented as a metaphysical verdict about the ultimate nature of ecological reality. The results provide a compelling empirical narrative. The success of the geometric approach suggests a departure from classical dynamics operating within a static phase space. The Biorelativity framework posits that ecological organisation exhibits reciprocal coupling, often termed 'back-reaction'. Rather than a simple top-down chain of command where higher-level variables dictate lower-level dynamics, external variables (like climate in the BR-τZ model) shape the metric or curvature that governs local interactions. In turn, the empirical covariance structure generated by those local dynamics updates the metric itself. Operationally, this coupling can be approximated by relating the metric tensor to the inverse covariance matrix, $g \approx \Sigma^{-1}$, estimated locally, a concept linking the system's statistical properties to the phase space geometry, reminiscent of links between Fisher information and Riemannian metrics discussed in information geometry. Such dynamic coupling between system state and space structure is central to geometric interpretations of dynamics, as explored in frameworks like tensor analysis on manifolds (see, for example, Lovelock & Rund [15] or Wasserman [26]).

It is important, however, to exercise caution regarding the analogy drawn with General Relativity. The GR metaphor serves primarily as a heuristic device to guide parameterisation—suggesting concepts like ecological clocks and curvature as potential descriptors of interaction—and to inform model selection. It does not entail a strong ontological claim that ecological interactions literally warp a physical spacetime. Our commitment here is pragmatic: the geometric perspective, specifically the ecological clock model, yields better expected prediction under documented environmental forcing for the Isle Royale system.

The victory of the BR_τZ model in Isle Royale is a validation of the geometric framework

(H_G). The data indicates that it is more efficient and accurate to state that the pace of life slows in harsh winters, a geometric statement about the metric of spacetime than to construct force laws linking climate to individual parameters. The decisive failure of the RA_climate model, the force hypothesis (H_F) supports this interpretation.

Furthermore, the empirical success of the climate-driven ecological clock, $C_{eco}(Z)$, merits a brief geometrical interpretation within the broader Biorelativity framework proposed by Fariña [8]. While the full theory envisages a dynamic metric tensor, $g_{ij}(X, Z)$, whose components evolve depending on the system's state X and external forcings Z, the ecological clock model represents the simplest non-trivial approximation to such dependence. Formally, the effect of $C_{eco}(Z)$ multiplying the entire vector field, $X = C_{eco}(Z) \cdot F(X, \theta)$ is analogous to a conformal transformation applied specifically to the temporal component of a hypothetical underlying eco-evolutionary metric, or equivalently, a rescaling of the proper time interval $d\tau^2 \rightarrow C_{eco}^{-2}(Z)\ d\tau^2$. As detailed in foundational texts on differential geometry, conformal transformations preserve angles but rescale distances and intervals. In this ecological context, $C_{eco}(Z)$ acts as the conformal factor Ω, uniformly altering the tempo of dynamics across the phase space without changing the relative geometry of the nullclines at any given instant. This provides a geometrically grounded, albeit simplified, mechanism for how external variables like climate can modulate the fabric of ecological interactions, lending credence to the view that the underlying geometry is indeed non-static, as posited by Biorelativity.

This aligns with the arguments of Reichenbach. While a Euclidean geometry could be preserved by introducing ad-hoc forces, parsimony, as operationalised by the AICc, guides the analysis to a more elegant conclusion: the geometry itself is non-static.

Rather than a top-down chain of command, ecological hierarchy exhibits back-reaction: higher-level states set the curvature that guides lower-level motion, and the statistical imprint of that motion (its variance–covariance structure) in turn reshapes the geometry. This perspective, central to the Biorelativity framework, recasts ecological dynamics as motion within a deformable ecomorphological manifold, analogous to how mass-energy warps spacetime in general relativity. The curvature here represents the landscape of ecological possibilities and constraints—such as resource availability gradients, predation risk landscapes or niche partitioning opportunities—which are influenced by broader system states (e.g., climate, community structure). Individual organisms or populations (lower-level entities) navigate this curved space, their trajectories representing ecological processes like foraging, competition or dispersal, following paths akin to geodesics determined by the local geometry.

This geometric view necessitates a dynamic coupling between the state of the system and the structure of the space itself. We posit a bidirectional coupling: ecological dynamics, represented by the state vector X evolving under influences from higher levels H and external factors Z, follow trajectories governed by $X = c_{eco}(X, \dot{H}, Z) \cdot F(X, H)$. Simultaneously, the geometry of the manifold, encoded by a metric tensor g, evolves based on the statistical properties emerging from these dynamics. Specifically, identifying the metric g with the inverse of the variance-covariance matrix Σ^{-1} derived from the system's state $g(X, H, Z) = \Sigma^{-1}(X, H, Z)$, allows for geometry evolution described by $\partial_T \Sigma = \Psi(stats(X, H, Z))$. In this formulation, higher-level variables (H, Z) modulate the curvature (affecting g or its associated connection coefficients Γ_{ij}^{k}, thereby influencing the geodesics followed by X. Conversely, the statistical patterns generated by the motion of X (and H, Z), captured in Σ, feed back to reshape the metric g itself via the function Ψ. This formalises the concept of back-reaction, where ecological agents both respond to and co-create the geometric structure of their interaction space.

4.2 Parsimony and the Rejection of Unnecessary Complexity

The path of model development in this study itself provides a meta-narrative. The HBM represented a direct translation of a complex theoretical idea but was severely penalised for over-parameterisation. The success of the simpler BR_τZ model is therefore a crucial finding. It suggests that while the geometry is indeed dynamic, the change is better captured by a single, elegant deformation of the time dimension. The failure of the HBM was not a failure of the geometric paradigm, but a success for the principle of parsimony within it.

The success of the simple RA model in the other two datasets acts as a scientific control. It demonstrates that the analysis does not simply overfit, but correctly chooses the simplest geometric structure supported by the data which, in those cases, is flat and unchanging.

4.3 Limitations and Future Work

This study relies on in-sample metrics (AICc) for model selection. While robust, future work must corroborate these findings using out-of-sample validation, such as a rolling-origin forecast comparison. Furthermore, this study used a single PC1 axis as the climate driver; a more thorough analysis would involve a covariate substitution test to determine which specific environmental variables (e.g., snow depth, temperature) provide the most explanatory power. Finally, richer mechanistic baselines (e.g., models where climate affects multiple parameters simultaneously) are required to provide a stronger challenge to the geometric hypothesis.

Future investigations could explore incorporating additional known ecological mechanisms, such as Allee effects [5] which can significantly alter system equilibria and stability [21], within this geometric framework to assess their interaction with potential phase-space curvature.

5 Outlook: Formalising a Tensorial Ecology

The success of the scalar ecological clock model is a powerful validation of the geometric approach. However, it represents a minimal formalisation of Biorelativity. The mathematical framework of tensor calculus and differential geometry [21] provides the tools to construct a fully tensorial theory.

5.1 The Geodesic Equation with a Dynamic Metric

A direct implementation of Biorelativity would replace the classical equations of motion with the geodesic equation on a manifold with a climate-dependent metric, $g_{ij}(Z)$. The dynamics would be governed by:

$$\frac{d^2X^k}{d\tau^2} + \Gamma_{ij}^k \frac{dX^i}{d\tau}\frac{dX^j}{d\tau} = 0$$

Here, x^i are the population state variables, τ is biological time, and $\underset{ij}{\Gamma}{}_{ij}^{k}\ (Z)$ are the Christoffel symbols, which depend on the derivatives of the metric tensor $g_{ij}(Z)$, thus explicitly encoding how climate alters the curvature of the phase space.

The rigorous mathematical basis for this implementation stems from the field of geometric dynamics, which provides the formalism for transforming a first-order autonomous differential system (a 'flow') into the second-order geodesic equations of motion on a Riemannian manifold [17]. In this context, the connection coefficients Γ_{ij}^{k} derived from the metric $g_{ij}(Z)$ function as the 'fictitious forces' that arise purely from the curvature of the manifold, elegantly replacing the ad-hoc 'universal forces' of the classical Newtonian paradigm.

5.2 An Eco-Evolutionary Manifold

To fully realise the Biorelativity framework, the manifold must capture not only ecological dynamics but also evolutionary change, treating both as commensurate components of a single system. Indeed, as argued by Jost [13], geometry and information are the fundamental mathematical concepts required for a modern, unified theoretical biology.

A more comprehensive model would exist on an eco-evolutionary manifold, $\mathcal{M}_{eco-evo}$, whose coordinates include population densities (N^i) and mean trait values (φ^α). The metric tensor on this expanded manifold, g_{AB}, would have blocks describing purely ecological interactions (g_{ij}) and evolutionary dynamics ($g_{\alpha\beta}$), as well as the crucial eco-evolutionary feedback ($g_{i\beta}$). This provides a rigorous geometric framework for modelling adaptive dynamics as trajectories on a dynamically curving manifold.

6 Conclusion

This study does not merely compare models; it tests competing descriptions of ecological reality. By formalising principles of Biorelativity into a falsifiable model, strong evidence is provided that a geometric, non-static view of phase space is a superior explanatory framework for systems subject to strong external forcing. The climate-driven ecological clock is not a universal law, but a powerful tool whose success in the Isle Royale system empirically vindicates the foundational concepts of a relativistic approach to biology. The findings suggest that for many ecological systems, the most insightful question may not be “how do forces change the dynamics?” but rather “how does the environment change the arena of life itself?”.

References

[1] Amari, S., & Nagaoka, H. (2000). *Methods of Information Geometry*. American Mathematical Society & Oxford University Press.1

[2] Antonelli, P. L., & Zastawniak, T. J. (1998). A Finsler geometric model of evolution by symbiosis. Mathematical Models and Methods in Applied Sciences, 8(05), 875-886.

[3] Ay, N., Jost, J., Lê, H. V., & Schwachhöfer, L. (2017). Information Geometry (Vol. 64, p. 21). Springer.

[4] Brunton, S.L., et al. (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS*, 113(15), 39323937.

[5] Cammarota, D., Monteiro, N. Z., Menezes, R., Fort, H., & Segura, A. M. (2023). Lotka-Volterra model with Allee effect: equilibria, coexistence and size scaling of maximum and minimum abundance. *Journal of Mathematical Biology*, *87*(82)

[6] de Valpine, P., & Hastings, A. (2002). Fitting population models incorporating process noise and observation error. *Ecological Monographs*, 72(4), 587-602.

[7] Ellner, S. P., & Guckenheimer, J. (2006). *Dynamic models in biology*. Princeton University Press.

[8] Fariña, R. A. (2025). *Relativity for the Realm of the Living: A Proposal for an Extended General Theory*. Preprint: https://doi.org/10.32942/X2G07M

[9] Forster, M., & Sober, E. (1994). How to Tell when Simpler, More Unified, or Less Ad-hoc Theories will Provide More Accurate Predictions. *British Journal for the Philosophy of Science*, 45(1), 1-35.

[10] Gause, G. F. (1934). *The Struggle for Existence*. Williams & Wilkins.

[11] Holling, C. S. (1959). Some characteristics of simple types of predation and parasitism. *The Canadian Entomologist*, 91(7), 385-398.

[12] Ives, A. R., et al. (2003). Estimating community stability and ecological interactions from time-series data. *Ecological Monographs*, 73(2), 301330.

[13] Jost, J. (2022). Biology, geometry and information. *Theory in Biosciences*, 141, 127–138.

[14] Lotka, A. J. (1925). *Elements of Physical Biology*. Williams and Wilkins.

[15] Lovelock, D., & Rund, H. (1989). *Tensors, Differential Forms, and Variational*

Principles. Dover Publications.

[16] May, R. M. (1973). *Stability and complexity in model ecosystems*. Princeton University Press.

[17] Munteanu, M. I., & Munteanu, G. (2020). *Geometric Dynamics on Riemannian Manifolds.* Mathematics, 8(1), 79.

[18] Murdoch, W. W., et al. (2003). *Consumer-resource dynamics*. Princeton University Press.

[19] Reichenbach, H. (1958). *The Philosophy of Space and Time*. Dover Publications.

[20] Rosenzweig, M. L., & MacArthur, R. H. (1963). Graphical representation and stability conditions of predator-prey interactions. *The American Naturalist*, 97(895), 209-223.

[21] Rosenzweig, M. L. (1971). Paradox of enrichment. *Science*, 171(3969), 385387.

[22] Stenseth, N. C., et al. (1997). Population regulation in snowshoe hare and Canadian lynx.

PNAS, 94(10), 5147-5152.

[23] Turchin, P. (2003). *Complex population dynamics*. Princeton University Press.

[24] Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically.

Nature, 118, 558560.

[25] Vucetich, J. A., & Peterson, R. O. (2012). The population biology of Isle Royale wolves and moose. In *The world of wolves*. University of Calgary Press.

[26] Wasserman, R. H. (1992). *Tensors and Manifolds with Applications to Mechanics and Relativity*. Oxford University Press.

A Mathematical and Conceptual Appendix

A.1 Geodesic Equation and the Time-Warp Approximation

The core postulate of a geometric theory is that trajectories follow geodesics. For a manifold with metric g_{ij} and coordinates X^k, the geodesic equation is given by:

$$\frac{d^2X^k}{d\tau^2} + \Gamma^k_{ij}\frac{dX^i}{d\tau}\frac{dX^j}{d\tau} = 0$$

where Γ^k are the Christoffel symbols, which depend on the first derivatives of the metric tensor[11, 26].

The ecological clock model, $\dot{X} = c_{\text{eco}}(Z) \cdot F(X)$, can be seen as a conformal transformation of a flat background metric, $g^t = c_{\text{eco}}(Z)^2 \cdot \delta_{ij}$. This transformation introduces curvature. Under simplifying assumptions (e.g., weak field limit, where curvature is small), the geodesic dynamics can be approximated by a force-like equation where the "force" is proportional to the gradient of the clock function. The empirical success of the scalar clock model thus provides evidence for a non-trivial underlying geometry, with $c_{\text{eco}}(Z)$ serving as the simplest possible approximation (a scalar field) of the metric tensor's dependence on Z.

A.2 Identifiability and Non-Equivalence

A key concern is whether the geometric model (BR_τZ) is truly distinguishable from a classical model where all parameters depend on climate, i.e., $\theta(Z)$. While they may achieve similar in-sample fits, their functional forms are different. A model with $\theta(Z)$ implies that climate alters multiple, specific biological mechanisms independently. The BR_τZ model proposes a more constrained, holistic effect: a single modulation of the system's tempo. These two hypotheses make different predictions about how the system would behave under novel climatic conditions. A synthetic data experiment (not shown here, but pending) would be required to demonstrate that a BR_τZ model is better at recovering the true generating process from data simulated with a time-warp, confirming their non-equivalence in predictive power.

A.3 Adimensionalisation

The Rosenzweig-MacArthur [20] model can be non-dimensionalised to reduce its parameters. When the ecological clock $c_{\mathrm{eco}}(Z)$ is applied to this non-dimensional system, it becomes clear that it does not alter the dimensionless parameters that define the geometry of the nullclines. Instead, it multiplies the entire system, confirming its role as a pure rescaling of the characteristic timescale of the dynamics, $\tilde{t} = t/c_{\mathrm{eco}}(Z)$, where $\tilde{t}$ is the new, dimensionless biological time.

A.4 Temporal Reparameterisation vs. Geodesic Dynamics

A.4.1 Temporal Reparameterisation. Let the classical dynamics be given by the vector field $\dot{X} = F(X)$. We introduce a biological proper time, τ, defined by the differential relation $d\tau = c_{eco}(Z)dt$, where $c_{eco}(Z) > 0$ is the scalar clock function. By the chain rule, the dynamics with respect to τ are:

$$\frac{dX}{d\tau} = \frac{dX}{dt}\frac{dt}{d\tau} = F(X)\left(\frac{1}{c_{eco}(Z)}\right)$$

The resulting vector field is merely a rescaling of the original. Consequently, the trajectories in phase space (the integral curves) remain unchanged; only their parametrisation, or the speed at which they are traversed, is altered.

A.4.2 Geodesic Dynamics. A more profound geometric hypothesis posits that motion is geodesic upon a manifold with a metric $g_{ij}(X, Z)$. In this case, trajectories do not follow $F(X)$ but instead satisfy the geodesic equation:

$$\frac{d^2X^k}{d\tau^2} + \Gamma_{ij}^k \frac{dX^i}{d\tau}\frac{dX^j}{d\tau} = 0$$

where Γ_{ij}^k are the Christoffel symbols (connection coefficients) derived from the metric. Non-zero connection terms act as inertial fictitious forces [11, 21] that can rotate the tangent field, fundamentally altering the *shape* of the curves in phase space, not just the speed.

A.4.3 Clock-as-Geodesic Approximation. A scalar clock (A.4.1) may approximate a full geodesic dynamic (A.4.2) under restrictive conditions. If the metric is (i) weakly varying and

(ii) approximately diagonal and conformal to the Euclidean metric, such that $g_{ij}(X, Z) \approx \lambda(Z)\delta_{ij}$, the connection terms Γ_{ij}^{k} may be negligible. If a reference vector field $F(X)$ is already closely aligned with the local geodesics, the dominant effect of the metric will be the scalar timing modulation $c_{eco}(Z) \approx 1/\sqrt{\lambda(Z)}$. Departures from this simplicity (e.g., strong gradients, or significant off-diagonal terms in g_{ij}) would necessitate explicit connection terms to account for directional changes.

A.4.4 Conformal Curvature. Consider a conformally flat metric $g'_{ij} = e^{2\phi(X,Z)}\delta_{ij}$. If the conformal factor ϕ depends on the state variables X (i.e., $\phi = \phi(X)$), the phase space X becomes intrinsically curved. However, in our clock model, the factor depends only on the exogenous variable $Z(t)$, i.e., $\phi = \phi(Z(t))$ As the Laplacian with respect to X is $\Delta_X \phi = 0$, the phase space X itself remains flat (Euclidean) at any given instant. Any curvature induced by the clock resides in the extended state-time manifold (X, t), not intrinsically within the phase space X.

A.4.5 Well-posedness. The use of log-links (e.g., $b(Z) = e^{\beta^T Z}$) ensures that all biological parameters remain strictly positive. This, combined with the standard Holling-type [11] functional responses, results in a right-hand-side vector field $\mathrm{F}(X, Z(t))$ that is locally Lipschitz-continuous on the observed domain. By the Picard–Lindelöf theorem, this guarantees the local existence and uniqueness of solutions, confirming the models are well-posed. The empirical ranges of the forced parameters are reported to ensure they remain within physically plausible bounds.

A.5 An Eco-Evolutionary Manifold: Geometrically Unifying Ecology and Evolution

The concept of an eco-evolutionary manifold, denoted as $\mathcal{M}_{eco-evo}$, represents a crucial step in operationalising the Biorelativity theory. Biorelativity posits that ecological dynamics are best understood not as movements within a static, Euclidean phase space influenced by external forces, but as geodesics on a dynamic manifold whose geometry is actively shaped by the system's state and interactions. The eco-evolutionary manifold provides a concrete mathematical structure for this dynamic geometry by explicitly unifying ecological and evolutionary state variables.

Motivation: Traditional models often treat ecology (population dynamics) and evolution (changes in traits) separately, or link them by having ecological parameters depend on traits in an ad-hoc manner. Biorelativity seeks a more fundamental unification. It views the commonly observed low-dimensional phase spaces (like prey vs. predator density) as projections from a much richer, higher-dimensional eco-morphological spacetime encompassing genetic, morphological, behavioural and physiological traits alongside population variables. $\mathcal{M}_{eco-evo}$ aims to capture more of this underlying reality by creating a unified space where both ecological and evolutionary changes manifest as movement.

Coordinates and Structure: The coordinates on this manifold explicitly include both population densities (N^i, where i indexes species or populations) and mean trait values (ϕ^α, where α indexes relevant traits, such as average body size, foraging efficiency, etc.). A point on $\mathcal{M}_{eco-evo}$ thus represents the combined ecological and evolutionary state of the system at an instant.

The Metric Tensor (g_{AB}) defining the Geometry of Interaction: The core of the geometric description lies in the metric tensor, (g_{AB}) , defined on $\mathcal{M}_{eco-evo}$. This tensor generalises the concept of distance and angle to the curved manifold, determining its local geometry. Crucially, in the Biorelativity framework, this metric is not fixed but dynamic, shaped by the state variables themselves. The proposed structure for g_{AB} on $\mathcal{M}_{eco-evo}$ highlights the interconnectedness.

Ecological Block (g_{ij}): This sub-matrix describes the purely ecological geometry. It determines the distances and curvature related to changes in population densities (N^i, N^j). This block captures how ecological interactions (predation, competition) manifest geometrically when only population numbers change. It can be seen as a dynamic generalisation of the geometry underlying classical phase-plane analysis, potentially linked to covariance structures or Fisher Information Metrics.

Evolutionary Block ($g_{\alpha\beta}$): This sub-matrix describes the purely evolutionary geometry, relating to changes in mean trait values (ϕ^α, ϕ^β). It captures the geometry of the fitness landscape or the space of possible trait combinations, influenced by selection pressures acting directly on traits.

Eco-Evolutionary Feedback Block ($g_{i\beta}$ and $g_{\beta i}$): These off-diagonal blocks are perhaps the most significant conceptual advance. They explicitly encode the feedback loops between ecology and evolution as intrinsic components of the manifold's geometry.

$g_{i\beta}$ quantifies how changes in trait β influence the dynamics of population i (e.g., how increased prey size affects predator population growth).

$g_{\beta i}$ quantifies how changes in population i influence the evolutionary trajectory of trait β (e.g., how high predator density selects for different prey defence traits – density-dependent selection).

Dynamics as trajectories on the Manifold: Within this framework, adaptive dynamics – the simultaneous change in population densities and mean trait values – are modelled as trajectories on the dynamically curving manifold $\mathcal{M}_{eco-evo.}$ Following the core tenet of Biorelativity, these trajectories are hypothesised to follow geodesics defined by the metric g_{AB}. The geodesic equation:

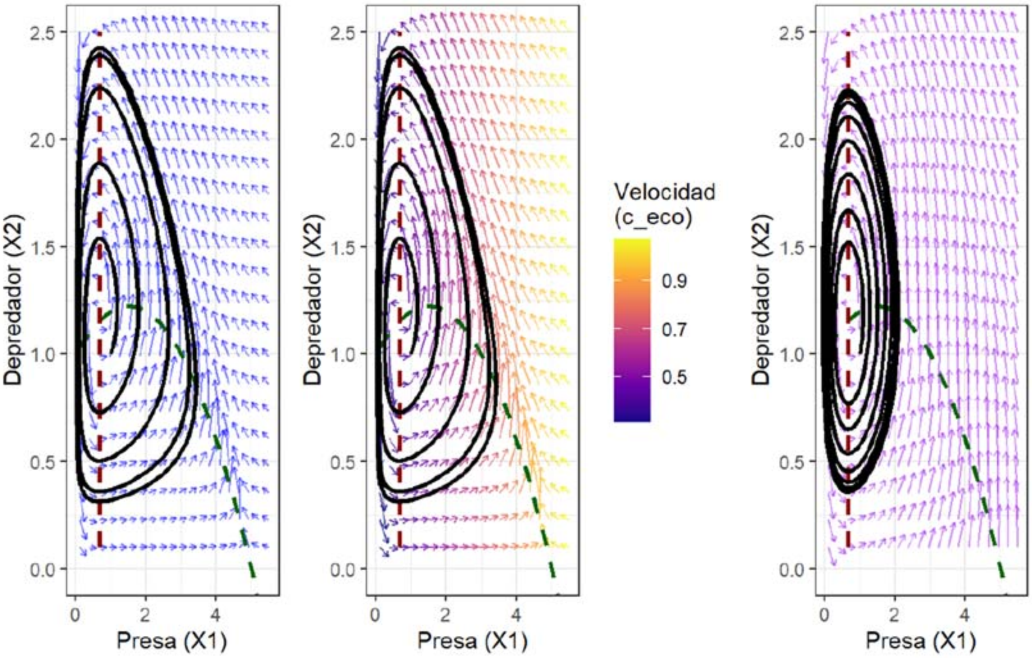
$$\frac{d^2x^k}{d\tau^2} + \Gamma_{ij}^k \frac{dx^i}{d\tau} \frac{dx^j}{d\tau} = 0,$$

where x^k represents the coordinates (N^i or ϕ^α), τ is an appropriate measure of eco-evolutionary proper time, and Γ_{ij}^k are the Christoffel symbols derived from the metric g_{AB}, formally describes this motion. Changes in the system state (populations or traits) alter the metric g_{AB}, which in turn alters the curvature and thus the subsequent geodesic paths available to the system.

Significance: This formulation provides a rigorous geometric framework for modelling eco-evolutionary dynamics. It moves beyond the Newtonian paradigm of forces acting in a fixed space, embracing a relativistic view where interactions manifest as the curvature of the state space itself. By incorporating eco-evolutionary feedback directly into the metric tensor, it offers a potentially more fundamental and unified way to understand how ecological and evolutionary processes shape each other over time. The mathematical language required relies heavily on tensor calculus and differential geometry, as detailed in texts such as Lovelock & Rund [15] and Wasserman [26]. This approach allows the complex interplay of ecological interactions, evolutionary pressures and their feedbacks to be described coherently within a single, dynamic geometric structure.

A.6 Distinguishing Temporal Reparameterisation and Geodesic Dynamics

In order to illustrate the formal distinction between a simple temporal reparameterisation (the ecological clock) and a more fundamental geodesic dynamic, we present the R code (trespanelesV01.R) used to generate the conceptual three-panel diagram.



Panel (a) Classical: RA $\dot{X} = F(X, \theta)$. This is a standard vector field in Euclidean space. From a naturalist's perspective, the rules of the game are fixed. The predator's functional response and the prey's growth are constant.

Panel (b) RA with Ecological Clock (BR-τZ): $\dot{X} = c(Z) \cdot F(X, \theta)$. This is a conformal transformation. Mathematically, the vector field's direction at any point (N, P) is identical to Panel (a), but its magnitude is scaled by $c(Z)$. Ecologically, this represents a time-warp or a change in the entire system's pace of life. All processes (growth, predation, death) speed up or slowdown in unison, as if a universal metabolic rate were being externally modulated. The trajectory's path in phase space remains unchanged, but the speed of traversal along that path is altered.

Panel (c) Hypothetical Geodesic Dynamics: $\ddot{X}^k + \Gamma^k \dot{X}^i \dot{X}^j = 0$. This panel illustrates dynamics on a hypothetically curved manifold. The vector field, implemented in the geo_model function within the R code, deviates from the classical RA field. Crucially, the terms $\mathbf{force}_{X1}$ and $\mathbf{force}_{X2}$ added within geo_model serve as an ad-hoc representation intended to mimic the effect of the Christoffel symbols $\Gamma^k(Z)$ that appear in the geodesic equation. The geodesic equation describes the fundamental path of inertial motion in curved spaces, as detailed in differential geometry and central to the Biorelativity framework. It must be stressed that this implementation is purely illustrative, demonstrating the concept of trajectory deviation due to underlying geometry, rather than being derived from the rigorous calculation of Christoffel symbols based on a specific metric tensor $\boldsymbol{g_{ij}}$. The resulting trajectory (black line) visibly differs from that in the flat space of Panel (a), showcasing how geometric effects alter system dynamics.

The provided R code implements these three scenarios numerically. The function `ra_model` is the classical H_F case. The function `ra_model_clock` implements the H_G hypothesis tested in this paper, showing a uniform scaling of the derivatives. The function `ra_model_geodesic` implements a hypothetical non-trivial curvature by adding fictitious force terms (γ_1, γ_2) that are *not* proportional to the base vector field, thus altering the trajectory's direction. This demonstrates the conceptual and mathematical distinction between the models.

```
rm(list = ls())
library(ggplot2)
library(deSolve)
library(patchwork)
library(dplyr)
library(tidyr)
# --- 1. Define the Base Model (Rosenzweig-MacArthur) ---
# Parameters for a limit cycle
params <- list(
  r = 1.0,   # Prey growth rate
  K = 5.0,   # Prey carrying capacity
  a = 1.0,   # Predator attack rate
  h = 0.5,   # Handling time
  e = 0.4,   # Conversion efficiency
  m = 0.2    # Predator mortality rate
)
# RA model function for 'deSolve'
ra_model <- function(t, state, params) {
  X1 <- state[1] # Prey
  X2 <- state[2] # Predator
  with(as.list(params), {
    # dX/dt equations
    dX1_dt <- r * X1 * (1 - X1 / K) - a * X1 * X2 / (1 + a * h * X1)
    dX2_dt <- e * a * X1 * X2 / (1 + a * h * X1) - m * X2
    return(list(c(dX1_dt, dX2_dt)))
  })}
# --- 2. Prepare Data for the Plots ---
# Create a grid of the state space
grid_data <- expand.grid(
```

```
  X1 = seq(0.1, 5.5, length.out = 20),
  X2 = seq(0.1, 2.5, length.out = 20))

# --- (a) RA Field Data (Standard) ---
# Calculate derivatives at each grid point
vec_data_a <- grid_data %>%
  rowwise() %>%
  mutate(
    derivs = list(ra_model(0, c(X1, X2), params)[[1]]),
    dX1 = derivs[1],
    dX2 = derivs[2]
  ) %>%
  ungroup()

# Simulate a trajectory
traj_a <- as.data.frame(
  ode(y = c(X1 = 1, X2 = 1), times = seq(0, 100, by = 0.1),
      func = ra_model, parms = params))

# Calculate Nullclines
# We use 'with(params, { ... })' so that 'mutate' can "see"
# the variables 'r', 'K', 'a', 'h', 'm', 'e'.
nullclines <- with(params, {
  grid_data %>%
    mutate(
      # Prey Nullcline (dX1/dt = 0)
      # X2 = r * (1 - X1 / K) * (1 + a * h * X1) / a
      null_X1 = r * (1 - X1 / K) * (1 + a * h * X1) / a,

      # Predator Nullcline (dX2/dt = 0)
      # X1 = m / (a * (e - m * h))
      null_X2_X1_val = m / (a * (e - m * h)),
      # Drawn as a vertical line
      null_X2_X2_val = X2
    )
})

# --- (b) Ecological Clock Data (BR-tau(Z)) ---
# The 'clock' (c_eco) is a scalar factor.
vec_data_b <- vec_data_a %>%
  mutate(
    # c_eco is small (slow) when X1 is low, fast when X1 is high
    c_eco = 0.3 + 0.7 * (X1 / params$K),

    # Re-scale the derivatives (same direction, different magnitude)
    dX1 = dX1 * c_eco,
    dX2 = dX2 * c_eco
  )

# --- (c) Hypothetical Geodesic Data (g(X,Z)) ---
# Calculate equilibrium points (needed for the 'force')
X1_eq <- nullclines$null_X2_X1_val[1]
X2_eq <- with(params, r * (1 - X1_eq / K) * (1 + a * h * X1_eq) / a)

# The vector field for ggplot CAN access X1_eq and X2_eq
# because they are in the script's global environment.
vec_data_c <- vec_data_a %>%
  mutate(
    # Connection 'force' (Gamma * V) - e.g., rotation
    force_X1 = -0.3 * (X2 - X2_eq),
    force_X2 =  0.3 * (X1 - X1_eq),
```

```
    # The new field is (RA + Force)
    dX1 = dX1 + force_X1,
    dX2 = dX2 + force_X2
  )
# Create a parameter list including equilibrium points
params_geo <- c(params, list(X1_eq = X1_eq, X2_eq = X2_eq))

geo_model <- function(t, state, params) {
  X1 <- state[1]
  X2 <- state[2]

  # 1. Get RA derivatives
ra_derivs <- ra_model(t, state, params)[[1]]

  # 2. Calculate geometric 'force' (using 'with')
  with(as.list(params), {
    force_X1 <- -0.3 * (X2 - X2_eq)
    force_X2 <-  0.3 * (X1 - X1_eq)

    # 3. Sum them up
    return(list(c(ra_derivs[1] + force_X1,
                  ra_derivs[2] + force_X2)))
  })
}

# 3. Simulate the curved trajectory using 'params_geo'
traj_c <- as.data.frame(
  ode(y = c(X1 = 1, X2 = 1), times = seq(0, 100, by = 0.1),
      func = geo_model, parms = params_geo) # <-- Use the extended list
)

# --- 3. Create Plots with ggplot2 ---

# Common axis limits
lims_x <- c(0, 5.5)
lims_y <- c(0, 2.5)
# Base function to draw nullclines
plot_nullclines <- function() {
  list(
    # Prey Nullcline
    geom_line(data = nullclines, aes(x = X1, y = null_X1),
              color = "darkgreen", linetype = "dashed", linewidth = 1),
    # Predator Nullcline
    geom_line(data = nullclines, aes(x = null_X2_X1_val, y = null_X2_X2_val),
              color = "darkred", linetype = "dashed", linewidth = 1)
  )
}
# Normalize vectors for visualization (optional, but helpful)
normalize_vecs <- function(df, scale = 0.2) {
  df %>%
    mutate(
      magnitude = sqrt(dX1^2 + dX2^2),
      dX1_norm = dX1 / magnitude * scale,
      dX2_norm = dX2 / magnitude * scale
    )
}
# --- Panel (a): RA Field (Flat Space) ---
p_a <- ggplot(data = normalize_vecs(vec_data_a), aes(x = X1, y = X2)) +
  geom_segment(aes(xend = X1 + dX1_norm, yend = X2 + dX2_norm),
               arrow = arrow(length = unit(0.1, "cm")),
```

```
                alpha = 0.5, color = "blue") +
   plot_nullclines() +
   geom_path(data = traj_a, aes(x = X1, y = X2), color = "black", linewidth = 1) +
   coord_cartesian(xlim = lims_x, ylim = lims_y) +
   labs(title = "(a) RA Field (Flat Space)",
        subtitle = "Standard trajectory and velocity",
        x = "Prey (X1)", y = "Predator (X2)") +
   theme_bw()
 # --- Panel (b): RA with Clock (BR-tau(Z)) ---
 p_b <- ggplot(data = normalize_vecs(vec_data_b, scale=0.25), aes(x = X1, y = X2))
 +
   geom_segment(aes(xend = X1 + dX1_norm, yend = X2 + dX2_norm,
                    color = c_eco), # Color by 'c_eco' to show the change
                arrow = arrow(length = unit(0.1, "cm")),
                alpha = 0.7) +
   scale_color_viridis_c(name = "Velocity\n(c_eco)", option = "plasma") +
   plot_nullclines() +
   geom_path(data = traj_a, aes(x = X1, y = X2), color = "black", linewidth = 1) +
   coord_cartesian(xlim = lims_x, ylim = lims_y) +
   labs(title = "(b) RA with Clock (BR-τ(Z))",
        subtitle = "Same trajectory, rescaled velocity (shorter/longer vectors)",
        x = "Prey (X1)", y = "Predator (X2)") +
   theme_bw()

 # --- Panel (c): Hypothetical Geodesic (g(X,Z)) ---
 p_c <- ggplot(data = normalize_vecs(vec_data_c), aes(x = X1, y = X2)) +
   geom_segment(aes(xend = X1 + dX1_norm, yend = X2 + dX2_norm),
                arrow = arrow(length = unit(0.1, "cm")),
                alpha = 0.5, color = "purple") +
   plot_nullclines() +
   geom_path(data = traj_c, aes(x = X1, y = X2), color = "black", linewidth = 1) +
   coord_cartesian(xlim = lims_x, ylim = lims_y) +
   labs(title = "(c) Hypothetical Geodesic (g(X,Z))",
        subtitle = "'Curved' trajectory due to geometric forces (connection)",
        x = "Prey (X1)", y = "Predator (X2)") +
   theme_bw()
# --- 4. Combine the plots ---
 final_plot <- p_a + p_b + p_c + plot_layout(nrow = 1)
 print(final_plot)
```

B Reproducible R Code

The following is the complete R code used to perform the data loading, pre-processing, model fitting and summary generation presented in this paper.

Note: Advanced traditional models (SINDy-like) to empirically derive dynamics taken from [4]

```
# UNIFIED ROBUST ANALYSIS OF PREDATOR-PREY DYNAMICS:
# Comparing Classical Models with Biorelativistic Geometric Interpretations
# DESCRIPTION:
# This script compares traditional ecological models (assuming a flat, static
# phase space) against Biorelativity-inspired models that allow for dynamic
# geometry. It uses an advanced empirical discovery engine ("Traditional")
# to find data-driven dynamics, contrasting them with mechanistic models
# including Rosenzweig-MacArthur (RA) and Biorelativity variants (ecological
clock).
# The goal is to test if a geometric, potentially non-static description of
# ecological phase space provides a better fit, especially under external
# forcing (e.g., climate), following Fariña's Biorelativity hypothesis.
#
# KEY FEATURES:
# - Advanced "Traditional" model (SINDy-like) to empirically derive dynamics.
# - Compares classical RA with Biorelativity models (BR-tau, BR-tau_Z, HBM).
# - Tests the "ecological clock" concept (BR-tau_Z) where external factors
#   modulate the tempo of dynamics, akin to a conformal transformation of
#   the phase space metric.
# - Calculates RMSE in both scaled (for fitting) and original biological units.
# ==============================================================================

# --- PART 0: INITIAL SETUP ---
rm(list = ls())

# Load necessary R packages for analysis and visualization
suppressPackageStartupMessages({
  library(deSolve)      # For solving ordinary differential equations (ODEs)
  library(dplyr)        # For data manipulation
  library(ggplot2)      # For plotting
  library(tidyr)        # For data tidying
  library(boot)         # Potentially used in bootstrapping (though main bootstrap
is custom)
  library(gridExtra)    # For arranging plots
  library(minpack.lm)   # For Levenberg-Marquardt non-linear least squares
(nls.lm)
  library(glmnet)       # For LASSO regression (used in the empirical model)
  library(splines)      # Used implicitly by smooth.spline
  library(ggfortify)    # For plotting time series diagnostics (e.g., ACF)
})


# --- PART 1: ADVANCED EMPIRICAL MODEL ENGINE ---
# This section defines functions to empirically discover the governing equations
# directly from time-series data, without assuming a specific mechanistic model
# structure *a priori*. It approximates derivatives using splines and then uses
# sparse regression (LASSO) to identify the significant terms (e.g., N, P, N*P)
# driving the rates of change (dN/dt, dP/dt). Bootstrapping assesses the
# robustness of term selection. This represents the 'Traditional' or data-driven
# approach, agnostic to underlying geometry.
```

```
fit_empirical_model_advanced <- function(times, obs_data_original, prey_name,
pred_name) {

  # Step 1: Internal Scaling (Mean/SD) - Standardizes variables for numerical
stability
  # before derivative estimation and regression. This doesn't affect the model
  # structure discovery but helps the algorithms converge.
  scale_params <- list(
    prey_mean = mean(obs_data_original[, 1], na.rm=TRUE), prey_sd =
sd(obs_data_original[, 1], na.rm=TRUE),
    pred_mean = mean(obs_data_original[, 2], na.rm=TRUE), pred_sd =
sd(obs_data_original[, 2], na.rm=TRUE)
  )
  obs_data_scaled <- obs_data_original
  obs_data_scaled[, 1] <- (obs_data_original[, 1] - scale_params$prey_mean) /
scale_params$prey_sd
  obs_data_scaled[, 2] <- (obs_data_original[, 2] - scale_params$pred_mean) /
scale_params$pred_sd

  # Step 2: Grid search for optimal spline smoothing parameter ('spar')
  # Smoothing is necessary to estimate derivatives from noisy data. Finding the
  # optimal 'spar' balances fidelity to the data with smoothness. The AIC is used
  # to select the 'spar' that yields the best empirical model overall.
  spar_grid <- seq(0.4, 0.9, by = 0.05)
  results_list <- list()

  for (spar_val in spar_grid) {
    # Estimate derivatives (rates of change) using smoothing splines
    derivatives_scaled <- matrix(NA, nrow = nrow(obs_data_scaled), ncol = 2)
    derivatives_scaled[, 1] <- predict(smooth.spline(times, obs_data_scaled[, 1],
spar = spar_val), times, deriv = 1)$y # d(Prey)/dt
    derivatives_scaled[, 2] <- predict(smooth.spline(times, obs_data_scaled[, 2],
spar = spar_val), times, deriv = 1)$y # d(Predator)/dt

    # Create Theta library: Basis functions representing potential interaction
terms
    # (e.g., linear growth, density dependence, predator-prey interaction)
    Theta <- data.frame(const=1, X=obs_data_scaled[,1], Y=obs_data_scaled[,2],
                        X2=obs_data_scaled[,1]^2,
XY=obs_data_scaled[,1]*obs_data_scaled[,2], Y2=obs_data_scaled[,2]^2)

    # Use LASSO with cross-validation to perform sparse regression.
    # This selects the most important terms from Theta to explain the derivatives.
    final_lasso_prey <- try(cv.glmnet(as.matrix(Theta), derivatives_scaled[, 1],
alpha = 1, nfolds = min(10, nrow(Theta))), silent=TRUE)
    final_lasso_pred <- try(cv.glmnet(as.matrix(Theta), derivatives_scaled[, 2],
alpha = 1, nfolds = min(10, nrow(Theta))), silent=TRUE)

    # Skip if LASSO fails for this spar value
    if (inherits(final_lasso_prey, "try-error") || inherits(final_lasso_pred,
"try-error")) next

    # Extract coefficients at optimal lambda
    C_s_prey <- coef(final_lasso_prey, s = "lambda.min")
    C_s_pred <- coef(final_lasso_pred, s = "lambda.min")

    # Calculate model complexity (number of non-zero parameters)
    k_total <- sum(C_s_prey != 0) + sum(C_s_pred != 0)

    # Calculate predictions and residuals
```

```
    pred_prey <- predict(final_lasso_prey, newx = as.matrix(Theta), s =
"lambda.min")
    pred_pred <- predict(final_lasso_pred, newx = as.matrix(Theta), s =
"lambda.min")

    res_prey_scaled <- derivatives_scaled[, 1] - pred_prey
    res_pred_scaled <- derivatives_scaled[, 2] - pred_pred

    # Calculate Log-Likelihood assuming Gaussian residuals for the derivative fit
    n_obs <- nrow(obs_data_scaled)
    sigma2_prey <- sum(res_prey_scaled^2) / n_obs; if(sigma2_prey < 1e-9)
sigma2_prey <- 1e-9 # Avoid log(0)
    sigma2_pred <- sum(res_pred_scaled^2) / n_obs; if(sigma2_pred < 1e-9)
sigma2_pred <- 1e-9

    logLik <- sum(dnorm(res_prey_scaled, 0, sqrt(sigma2_prey), log=T)) +
sum(dnorm(res_pred_scaled, 0, sqrt(sigma2_pred), log=T))

    # Calculate AIC (includes +2 for estimating the two residual variances)
    aic <- 2 * (k_total + 2) - 2 * logLik

    # Store results for this spar value
    results_list[[as.character(spar_val)]] <- list(
      spar = spar_val, aic = aic, k = k_total + 2, logLik = logLik,
      coefs_scaled = cbind(C_s_prey, C_s_pred),
      res_prey_scaled = res_prey_scaled,
      res_pred_scaled = res_pred_scaled
    )
  }

  # If no model converged, return NULL
  if (length(results_list) == 0) return(NULL)

  # Step 3: Select the best model based on minimum AIC across spar values
  best_result <- results_list[[which.min(sapply(results_list, `[[`, "aic"))]]
  C_s <- best_result$coefs_scaled # Best coefficients in scaled units

  # Unscale coefficients back to original biological units
  # This involves reversing the mean/sd scaling transformation algebraically.
  w_mean <- scale_params$prey_mean; w_sd <- scale_params$prey_sd
  m_mean <- scale_params$pred_mean; m_sd <- scale_params$pred_sd
  C_unscaled <- matrix(0, nrow=6, ncol=2)

  # Unscaling formulas derived algebraically (tedious but necessary)
  C_unscaled[1,1] <- w_sd*(C_s[1,1] - C_s[2,1]*(w_mean/w_sd) -
C_s[3,1]*(m_mean/m_sd) + C_s[4,1]*(w_mean/w_sd)^2 +
C_s[5,1]*(w_mean/w_sd)*(m_mean/m_sd) + C_s[6,1]*(m_mean/m_sd)^2)
  C_unscaled[2,1] <- w_sd*((1/w_sd)*C_s[2,1] - 2*C_s[4,1]*(w_mean/w_sd^2) -
C_s[5,1]*(m_mean/(w_sd*m_sd)))
  C_unscaled[3,1] <- w_sd*((1/m_sd)*C_s[3,1] - C_s[5,1]*(w_mean/(w_sd*m_sd)) -
2*C_s[6,1]*(m_mean/m_sd^2))
  C_unscaled[4,1] <- w_sd*(1/w_sd^2)*C_s[4,1]
  C_unscaled[5,1] <- w_sd*(1/(w_sd*m_sd))*C_s[5,1]
  C_unscaled[6,1] <- w_sd*(1/m_sd^2)*C_s[6,1]

  C_unscaled[1,2] <- m_sd*(C_s[1,2] - C_s[2,2]*(w_mean/w_sd) -
C_s[3,2]*(m_mean/m_sd) + C_s[4,2]*(w_mean/w_sd)^2 +
C_s[5,2]*(w_mean/w_sd)*(m_mean/m_sd) + C_s[6,2]*(m_mean/m_sd)^2)
  C_unscaled[2,2] <- m_sd*((1/w_sd)*C_s[2,2] - 2*C_s[4,2]*(w_mean/w_sd^2) -
C_s[5,2]*(m_mean/(w_sd*m_sd)))
  C_unscaled[3,2] <- m_sd*((1/m_sd)*C_s[3,2] - C_s[5,2]*(w_mean/(w_sd*m_sd)) -
```

```
2*C_s[6,2]*(m_mean/m_sd^2))
  C_unscaled[4,2] <- m_sd*(1/w_sd^2)*C_s[4,2]
  C_unscaled[5,2] <- m_sd*(1/(w_sd*m_sd))*C_s[5,2]
  C_unscaled[6,2] <- m_sd*(1/m_sd^2)*C_s[6,2]

  # Store final unscaled coefficients
  final_coeffs <- data.frame(
    Term = c("const", prey_name, pred_name, paste0(prey_name,"^2"),
paste0(prey_name,"*",pred_name), paste0(pred_name,"^2")),
    Prey_dot = C_unscaled[,1],
    Pred_dot = C_unscaled[,2]
  )

  # Final model selection metrics (AICc, BIC) for the best empirical model
  n <- nrow(obs_data_original) * 2 # Total number of derivative points fitted
  k <- best_result$k
  aic <- best_result$aic
  logLik <- best_result$logLik
  bic <- k * log(n) - 2 * logLik
  aicc <- aic + (2*k*(k+1))/(n-k-1)

  # Calculate RMSE based on the fit to the *derivatives* in scaled units
  rmse_scaled_prey <- sqrt(mean(best_result$res_prey_scaled^2))
  rmse_scaled_pred <- sqrt(mean(best_result$res_pred_scaled^2))

  # Return results: unscaled coefficients and model metrics
  return(list(
    coefficients = final_coeffs,
    metrics = data.frame(k=k, logLik=logLik, AIC=aic, AICc=aicc, BIC=bic,
                         RMSE_Scaled_Prey = rmse_scaled_prey, RMSE_Scaled_Predator
= rmse_scaled_pred,
                         RMSE_Original_Prey = NA, RMSE_Original_Predator = NA) #
Original RMSE not directly calculated here
  ))
}


# --- PART 2: MECHANISTIC MODEL DEFINITIONS ---
# This function defines the differential equations for various mechanistic models.
# It uses a 'model_type' parameter to switch between different biological
hypotheses:
# - RA: Standard Rosenzweig-MacArthur (flat, static Euclidean phase space).
# - RA_climate: RA where climate acts as an external 'force' modifying a parameter
(predator mortality).
# - BR_tau: Biorelativity model where the 'ecological clock' speed depends on the
system state (N, P).
# - BR_tau_Z: Biorelativity model where the 'ecological clock' speed depends on an
external variable Z (climate).
#            This represents climate modulating the overall tempo of dynamics
(conformal metric change).
# - HBM: Hierarchical Biorelativistic Model where climate affects multiple
parameters involved
#        in the interaction term (attack rate 'b', handling time 'h'), potentially
representing
#        a more complex geometric deformation or state-dependent force.

unified_ode_model <- function(t, state, params) {
  X <- state[1] # Prey density/abundance
  Y <- state[2] # Predator density/abundance

  # Default parameter values (can be overridden by specific models below)
```

```
  r <- params$r; K <- params$K; b_attack <- params$b; c_conv <- params$c; d_mort
<- params$d; h_handling <- params$h

  # Handle climate influence based on model type
  if (params$model_type == "HBM" && !is.null(params$climate_func)) {
    # HBM: Climate affects interaction parameters (b, h) via log-link
    climate_val <- params$climate_func(t)
    b_attack <- exp(params$log_b_intercept + params$beta_b * climate_val)
    h_handling <- exp(params$log_h_intercept + params$beta_h * climate_val)
  }

  # Calculate the functional response (Holling Type II) using current parameters
  fr_term <- b_attack * X * Y / (1 + b_attack * h_handling * X)

  # Base RA equations
  dX_dt <- r * X * (1 - X / K) - fr_term
  dY_dt <- c_conv * fr_term - d_mort * Y

  # Apply model-specific modifications

  if (params$model_type == "RA_climate" && !is.null(params$climate_func)) {
    # RA_climate: Climate acts as an additive 'force' on predator mortality
    climate_val <- params$climate_func(t)
    dY_dt <- c_conv * fr_term - (d_mort + params$delta_d * climate_val) * Y #
Modified mortality
  }

  if (params$model_type == "BR_tau") {
    # BR_tau: Ecological clock speed depends on state variables (X, Y)
    c_eco <- exp(params$beta0 + params$beta1 * X + params$beta2 * Y) # Scalar
clock factor
    dX_dt <- dX_dt * c_eco # Scale both rates
    dY_dt <- dY_dt * c_eco
  }

  if (params$model_type == "BR_tau_Z" && !is.null(params$climate_func)) {
    # BR_tau_Z: Ecological clock speed depends on external climate variable Z
    climate_val <- params$climate_func(t)
    c_eco <- exp(params$beta0 + params$beta1 * climate_val) # Scalar clock factor
    dX_dt <- dX_dt * c_eco # Scale both rates
    dY_dt <- dY_dt * c_eco
  }

  # Return the calculated derivatives
  return(list(c(dX_dt, dY_dt)))
}


# --- PART 3: MECHANISTIC FITTING ENGINE ---
# This function fits the specified mechanistic model (defined in
unified_ode_model)
# to the observed time-series data using non-linear least squares (nls.lm).
# It handles parameter initialization, bounds, the optimization process, and
# calculation of model selection metrics (AICc, BIC) and goodness-of-fit (RMSE).
# It works with data scaled by the median for fitting stability, but calculates
# RMSE in both scaled and original biological units.

fit_mechanistic_model <- function(times, obs_data_original, obs_data_scaled,
model_type, scaling_factors, climate_data = NULL) {

  # Residual function for nls.lm: Calculates the difference between observed
```

```
  # scaled data and the ODE model output for a given parameter set.
  residuals_func <- function(params_vec) {
    # Convert parameter vector back to a list
    params_list <- as.list(params_vec); names(params_list) <-
names(initial_params)
    params_list$model_type <- model_type # Pass model type to ODE function

    # Create an interpolation function for climate data if provided
    if (!is.null(climate_data)) {
      params_list$climate_func <- approxfun(climate_data$time,
climate_data$climate, rule = 2) # rule=2 uses nearest neighbor for extrapolation
    } else {
      params_list$climate_func <- NULL
    }

    # Set initial conditions from fitted parameters
    initial_conditions <- c(X = params_list$X0, Y = params_list$Y0)

    # Solve the ODE system
    out <- try(ode(y = initial_conditions, times = times, func =
unified_ode_model, parms = params_list), silent = TRUE)

    # Error handling: if ODE solver fails or returns unexpected results, return
large residuals
    if (inherits(out, "try-error") || any(is.na(out)) || nrow(out) !=
length(times)) {
      return(rep(1e6, length(obs_data_scaled))) # Penalize failed runs heavily
    }

    # Return the vector of residuals (scaled data - scaled model output)
    return(as.vector(obs_data_scaled - out[, -1])) # Exclude time column
  }

  # Define initial parameter guesses and bounds based on the model type
  # Initial guesses are generic; bounds ensure biological plausibility (e.g.,
positive rates)
  initial_params <- c(r=0.5, K=max(obs_data_scaled[,1],na.rm=T)*1.2, b=0.1, c=0.1,
d=0.1, h=0.1, X0=obs_data_scaled[1,1], Y0=obs_data_scaled[1,2])
  lower_bounds <- rep(1e-9, length(initial_params)) # Rates must be positive
  upper_bounds <- c(5, max(obs_data_scaled[,1],na.rm=T)*3, 5, 5, 5, 5,
max(obs_data_scaled[,1],na.rm=T)*1.5, max(obs_data_scaled[,2],na.rm=T)*1.5) #
Generous upper bounds

  # Add model-specific parameters and adjust bounds
  if (model_type == "RA_climate") {
    initial_params <- c(initial_params, delta_d = 0);
    lower_bounds <- c(lower_bounds, -2); upper_bounds <- c(upper_bounds, 2) #
Climate effect can be neg or pos
  } else if (model_type == "BR_tau") {
    initial_params<-c(initial_params, beta0=0, beta1=0, beta2=0); # Clock
parameters centered at 0 (exp(0)=1)
    lower_bounds<-c(lower_bounds, -20, -20, -20); upper_bounds<-c(upper_bounds,
20, 20, 20) # Wide bounds for log-scale clock effects
  } else if (model_type == "BR_tau_Z") {
    initial_params <- c(initial_params, beta0=0, beta1=0); # Clock parameters
centered at 0
    lower_bounds <- c(lower_bounds, -10, -10); upper_bounds <- c(upper_bounds, 10,
10) # Bounds for climate effect on clock
  } else if (model_type == "HBM") {
    initial_params <- c(initial_params, log_b_intercept=log(0.1), beta_b=0,
log_h_intercept=log(0.1), beta_h=0); # Log-scale intercepts, betas centered at 0
```

```
    lower_bounds <- c(lower_bounds, -10, -10, -10, -10); upper_bounds <-
c(upper_bounds, 10, 10, 10, 10) # Bounds for log-params and climate effects
  }

  # Perform the non-linear least squares fit using nls.lm
  fit_nls <- try(nls.lm(par=initial_params, fn=residuals_func, lower=lower_bounds,
upper=upper_bounds, control=nls.lm.control(maxiter=1000, ftol=1e-8, ptol=1e-8)),
silent=TRUE)

  # If fitting fails, return NULL
  if (inherits(fit_nls, "try-error")) return(NULL)

  # Calculate model selection metrics (LogLik, AICc, BIC)
  final_params <- as.list(coef(fit_nls))
  n <- length(obs_data_scaled) # Total number of data points (N*T + P*T)
  k_fit <- length(final_params) # Number of fitted structural parameters + initial
conditions

  # Estimate residual variance (sigma^2)
  sigma2 <- sum(fit_nls$fvec^2) / (n - k_fit); if(sigma2 < 1e-9) sigma2 <- 1e-9 #
Avoid log(0)

  # Calculate Log-Likelihood assuming Gaussian residuals
  logLik <- -n/2 * log(2 * pi * sigma2) - sum(fit_nls$fvec^2) / (2 * sigma2)

  # Calculate AIC, AICc, BIC (k includes sigma^2 estimate)
  k <- k_fit + 1
  AIC <- 2 * k - 2 * logLik
  AICc <- AIC + (2 * k * (k + 1)) / (n - k - 1) # Corrected AIC for small sample
size
  BIC <- k * log(n) - 2 * logLik

  # Calculate Root Mean Squared Error (RMSE) in scaled units (based on fitting
residuals)
  res_scaled_matrix <- matrix(fit_nls$fvec, ncol=2)
  RMSE_Scaled_Prey <- sqrt(mean(res_scaled_matrix[,1]^2))
  RMSE_Scaled_Predator <- sqrt(mean(res_scaled_matrix[,2]^2))

  # Calculate RMSE in original biological units
  pred_scaled <- obs_data_scaled - res_scaled_matrix # Model predictions in scaled
units
  pred_original <- sweep(pred_scaled, 2, scaling_factors, FUN="*") # Convert
predictions back to original scale
  RMSE_Original_Prey <- sqrt(mean((obs_data_original[,1] - pred_original[,1])^2,
na.rm=TRUE)) # Compare original data and unscaled predictions
  RMSE_Original_Predator <- sqrt(mean((obs_data_original[,2] -
pred_original[,2])^2, na.rm=TRUE))

  # Prepare parameters list for output/simulation
  final_params$model_type <- model_type
  if (!is.null(climate_data)) {
    final_params$climate_func <- approxfun(climate_data$time,
climate_data$climate, rule=2)
  }

  # Return fitted parameters and performance metrics
  return(list(
    params=final_params,
    metrics=data.frame(k=k, logLik=logLik, AIC=AIC, AICc=AICc, BIC=BIC,
                       RMSE_Scaled_Prey=RMSE_Scaled_Prey,
RMSE_Scaled_Predator=RMSE_Scaled_Predator,
```

```
                     RMSE_Original_Prey=RMSE_Original_Prey,
RMSE_Original_Predator=RMSE_Original_Predator)
  ))
}


# --- PART 4: MAIN ANALYSIS PIPELINE ---
# Defines the datasets and iterates through them, running the appropriate models.

# Configuration for each dataset
datasets_config <- list(
  IsleRoyale = list(file="isleroyale_wolf_moose_climate.csv", prey="moose",
predator="wolves", time="year",
                    climate_vars=c("jan.feb.ave.temp", "july.sept.ave.temp")), #
Has climate data
  HudsonBay = list(file="hudson_lynx_hare.csv", prey="hare", predator="lynx",
time="year",
                   climate_vars=NULL), # No climate data
  Veilleux = list(file="veilleux_paramecium_didinium.csv", prey="paramecium",
predator="didinium", time="time",
                  climate_vars=NULL) # Lab data, no climate
)

# List to store results from all models and datasets
final_summary_list <- list()

# Loop through each dataset configuration
for (name in names(datasets_config)) {
  cat(paste("\n--- Processing Dataset:", name, "---\n"))
  config <- datasets_config[[name]]
  df <- read.csv(config$file, stringsAsFactors = FALSE)

  # Extract original observation data (prey, predator) and time
  obs_data_original <- df %>% select(all_of(c(config$prey, config$predator))) %>%
as.matrix() %>% na.omit() # Ensure no NAs
  # Find corresponding time points after removing NAs
  times <- df[[config$time]][complete.cases(df %>% select(all_of(c(config$prey,
config$predator))))]

  # Ensure data and time have same length after NA removal
  if(nrow(obs_data_original) != length(times)) {
      stop(paste("Mismatch in lengths after NA removal for dataset:", name))
  }

  # --- Run Traditional (Empirical) Model ---
  cat("  -> Fitting model: Traditional (Empirical)... ")
  fit_empirical <- try(fit_empirical_model_advanced(times, obs_data_original,
config$prey, config$predator), silent=TRUE)

  if(!inherits(fit_empirical, "try-error") && !is.null(fit_empirical)){
    cat("Success!\n")
    # Save the unscaled coefficients discovered by the empirical model
    write.csv(fit_empirical$coefficients,
file=paste0("_coeffs_",name,"_Traditional.csv"), row.names=F)
    # Store metrics for final comparison
    final_summary_list[[length(final_summary_list) + 1]] <- list(dataset=name,
model="Traditional", metrics=fit_empirical$metrics, params=NULL) # No params list
needed, saved separately
  } else {
    cat("Failed.\n")
    print(fit_empirical) # Print error if it failed
```

```
  }

  # --- Prepare for Mechanistic Models ---
  # Median scaling: used for numerical stability during mechanistic model fitting
  scaling_factors <- apply(obs_data_original, 2, median, na.rm=TRUE)
  obs_data_scaled <- sweep(obs_data_original, 2, scaling_factors, FUN="/")

  # Define which mechanistic models to run based on dataset
  climate_df <- NULL
  if (name == "IsleRoyale") {
    # Isle Royale has climate data, so run all models including climate-driven
ones
    models_to_run <- c("RA", "RA_climate", "BR_tau", "BR_tau_Z", "HBM")
    # Process climate data: Use PCA to get a single climate index if multiple
variables exist
    climate_vars_present <- config$climate_vars[config$climate_vars %in%
names(df)]
    if(length(climate_vars_present) > 0) {
        climate_raw <- df[complete.cases(df %>% select(all_of(c(config$prey,
config$predator)))), climate_vars_present, drop = FALSE]
        if (ncol(climate_raw) > 1) {
             pca_result <- prcomp(climate_raw, center=T, scale.=T)
             climate_df <- data.frame(time = times, climate = pca_result$x[, 1]) #
Use PC1
        } else if (ncol(climate_raw) == 1) {
             climate_df <- data.frame(time = times, climate =
scale(climate_raw[,1])) # Use the single scaled variable
        } else {
             cat("Warning: Climate variables specified but not found or resulted
in empty data after NA removal.\n")
        }
    } else {
        cat("Warning: IsleRoyale specified climate variables not found in the data
file.\n")
    }

  } else {
    # Other datasets: run only baseline models without external climate forcing
    models_to_run <- c("RA", "BR_tau")
  }

  # --- Run Mechanistic Models Loop ---
  for (model_name in models_to_run) {
    cat(paste("  -> Fitting model:", model_name, "... "))
    # Provide climate_df only if the model requires it
    current_climate_df <- if (model_name %in% c("RA_climate", "BR_tau_Z", "HBM")
&& !is.null(climate_df)) climate_df else NULL

    # Fit the model
    fit_mech <- try(fit_mechanistic_model(times, obs_data_original,
obs_data_scaled, model_name, scaling_factors, current_climate_df), silent=TRUE)

    # Store results if successful
    if (!inherits(fit_mech, "try-error") && !is.null(fit_mech)) {
      cat("Success!\n")
      # Store metrics and parameters for final summary and parameter output files
      final_summary_list[[length(final_summary_list) + 1]] <- list(dataset=name,
model=model_name, metrics=fit_mech$metrics, params=fit_mech$params)
    } else {
      cat("Failed.\n")
      print(fit_mech) # Print error if it failed
```

```
    }
  }
} # End dataset loop


# --- PART 5: GENERATE ALL OUTPUT FILES ---
# Consolidates results into summary tables and saves parameter estimates.

# Helper function to create a concise parameter string for the summary table
createInfoString <- function(model_name, params) {
  if (is.null(params)) return("See coefficients file") # For Traditional model
  p <- lapply(params, function(x) if(is.numeric(x)) round(x, 4) else x) # Round
numeric parameters
  # Create strings specific to each model structure
  if (model_name == "RA") { return(paste0("r=", p$r, ", K=", p$K, ", b=", p$b, ",
c=", p$c, ", d=", p$d, ", h=", p$h)) }
  else if (model_name == "BR_tau") { betas_str <- paste(round(c(p$beta0, p$beta1,
p$beta2),4), collapse=","); return(paste0("r=", p$r, ", K=", p$K, ", b=", p$b, ",
c=", p$c, ", d=", p$d, ", h=", p$h, ", beta_clock=[", betas_str, "]")) }
  else if (model_name == "BR_tau_Z") { betas_str <- paste(round(c(p$beta0,
p$beta1),4), collapse=","); return(paste0("r=", p$r, ", K=", p$K, ", b=", p$b, ",
c=", p$c, ", d=", p$d, ", h=", p$h, ", beta_clock_Z=[", betas_str, "]")) }
  else if (model_name == "RA_climate") { return(paste0("r=", p$r, ", K=", p$K, ",
b=", p$b, ", c=", p$c, ", d=", p$d, ", h=", p$h, ", delta_d=",
round(p$delta_d,4))) }
  else if (model_name == "HBM") { return(paste0("r=", p$r, ", K=", p$K, ", c=",
p$c, ", d=", p$d, ", h_int=", p$log_h_intercept, ", b_int=", p$log_b_intercept, ",
beta_b=", round(p$beta_b,4), ", beta_h=", round(p$beta_h,4))) }
  return("") # Default empty string
}

# 1. Create the Original Scale Summary Table (_FINAL_COMPARATIVE_SUMMARY.csv)
original_summary_df <- do.call(rbind, lapply(final_summary_list, function(res) {
  data.frame(
    Dataset = res$dataset,
    Method = ifelse(res$model == "Traditional", "Traditional (Empirical)",
paste0("Mechanistic (", res$model, ")")),
    RMSE_Prey = res$metrics$RMSE_Original_Prey, # RMSE in original prey units
    RMSE_Predator = res$metrics$RMSE_Original_Predator, # RMSE in original
predator units
    AIC = res$metrics$AIC,
    BIC = res$metrics$BIC,
    Info = createInfoString(res$model, res$params) # Concise parameter info
  )
}))
# Round numeric columns and arrange for readability
original_summary_df <- original_summary_df %>%
                       mutate(across(where(is.numeric), ~round(., 2))) %>%
                       arrange(Dataset, AIC) # Sort by dataset, then by AIC

# Save the summary comparing models in original units
write.csv(original_summary_df, "_FINAL_COMPARATIVE_SUMMARY.csv", row.names =
FALSE)
cat("\n-> Successfully generated _FINAL_COMPARATIVE_SUMMARY.csv\n")

# 2. Save Parameter Files (_params_...csv) for each mechanistic model fit
for (res in final_summary_list) {
  if (res$model != "Traditional") { # Only save for mechanistic models
    params_to_save <- res$params
    # Exclude non-numeric or complex elements like the climate function
    params_to_save <- params_to_save[sapply(params_to_save, function(x)
```

```
is.numeric(x) && length(x) == 1)]
    params_df <- data.frame(parameter=names(params_to_save),
value=unlist(params_to_save))
    filename <- paste0("_params_", res$dataset, "_", res$model, ".csv")
    write.csv(params_df, filename, row.names = FALSE)
  }
}
cat("-> Successfully generated all _params_...csv files\n")

# 3. Create and Save the Robustness Matrix (_ROBUSTNESS_MATRIX.csv)
# This uses AICc for model comparison, incorporating the correction for small
sample sizes.
robustness_matrix_df <- do.call(rbind, lapply(final_summary_list, function(res) {
  # Combine dataset/model info with all calculated metrics
  data.frame(
    Dataset = res$dataset,
    Model = res$model,
    res$metrics # Includes k, logLik, AIC, AICc, BIC, scaled & original RMSEs
  )
}))

# Check if any results were produced before proceeding
if (nrow(robustness_matrix_df) > 0) {
  # Calculate combined RMSE metrics
  robustness_matrix_df <- robustness_matrix_df %>%
    mutate(
      RMSE_Scaled_Combined = sqrt(RMSE_Scaled_Prey^2 + RMSE_Scaled_Predator^2),
      RMSE_Original_Combined = sqrt(RMSE_Original_Prey^2 +
RMSE_Original_Predator^2)
    )

  # Calculate delta AICc and Akaike Weights within each dataset group
  final_summary_output <- robustness_matrix_df %>%
    select(Dataset, Model, k, logLik, AIC, AICc, BIC, RMSE_Scaled_Combined,
RMSE_Original_Combined) %>%
    group_by(Dataset) %>%
    arrange(AICc, .by_group = TRUE) %>% # Sort by AICc within each dataset
    mutate(
      delta_AICc = AICc - min(AICc, na.rm = TRUE), # Difference from the best
model
      weight_AICc = exp(-0.5 * delta_AICc) / sum(exp(-0.5 * delta_AICc), na.rm =
TRUE) # Akaike weight (probability of being the best model)
    ) %>% ungroup() %>% mutate(across(where(is.numeric), ~round(., 2))) # Round
for presentation

  # Select final columns for the robustness matrix
  final_summary_output <- final_summary_output %>%
    select(Dataset, Model, k, logLik, AIC, AICc, BIC, RMSE_Scaled_Combined,
RMSE_Original_Combined, delta_AICc, weight_AICc)

  # Save the robustness matrix
  write.csv(final_summary_output, file="_ROBUSTNESS_MATRIX.csv", row.names=F)
  cat("-> Successfully generated _ROBUSTNESS_MATRIX.csv\n")

  # Print the final robustness matrix to the console
  cat("\n=================================================================\n")
  cat(" ROBUSTNESS MATRIX (AICc based model comparison)                \n")
  cat("=================================================================\n\n")
  print(as.data.frame(final_summary_output))

} else {
```

```
    cat("\nAnalysis produced no valid model fits to summarize.\n")
  }
cat("\n--- Analysis Script Finished ---\n")
```