

Article

Lipschitz-Based Reinforcement Learning for Response-Time Distributions in Video-Game Design

Ana Coronado-Ferrer  and Enrique A. Sánchez-Pérez * 

Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain; ancofer1@alumni.upv.es

* Correspondence: easancpe@mat.upv.es

Abstract

This study proposes a mathematical framework for predicting complete response-time distributions associated with parametric video-game configurations. Each configuration is encoded as a point in a normalized metric space, and the statistical descriptors of its response-time distribution (median, mean, selected quantiles, interquartile range, and Skewness) are treated as real-valued Lipschitz functions on that space. Predictions for unseen configurations are obtained through McShane–Whitney extension formulas, which provide geometrically controlled upper and lower bounds compatible with the empirical Lipschitz regularity of the observed data. To handle the sequential incorporation of new observations, a regularization mechanism is introduced that replaces raw descriptors violating Lipschitz continuity constraints with a convex combination of the observed value and a weighted historical estimate. In the extended version of the method, the coefficient of this combination is selected through a one-pass online Q-learning-inspired procedure that selects, for each descriptor and instability regime, a data-dependent trade-off between fidelity and geometric regularity. The final output is a continuous Log-Normal density fitted by nonlinear least squares to the predicted descriptors, together with a Wasserstein-type uncertainty band derived from the Lipschitz bounds. The framework is validated on a controlled experiment with 24 participants across 20 game levels. Results show that the predicted distributions shift systematically with the input configuration and that, in the illustrative comparison, the adaptive mechanism produces feature-specific smoothing decisions that differ from those obtained with a fixed coefficient. The method also provides interpretable predictions from small experimental datasets without requiring fully data-driven models.

Keywords: response time; video game; reinforcement learning; lipschitz; metric regression**MSC:** 62P25

Academic Editor: Renato Bruni

Received: 16 June 2026

Revised: 19 July 2026

Accepted: 22 July 2026

Published: 24 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Response time is one of the most direct observable traces of how a player interacts with a game situation. In action-oriented tasks, it summarizes several processes that are relevant for game design: visual detection, attention, motor preparation, decision making, and execution [1]. For this reason, response-time data can be useful not only for describing player performance, but also for designing new game situations whose difficulty is better adjusted to the expected behavior of players. This is especially relevant in simple but frequent interaction patterns, such as detecting a moving object, aiming at it, and producing a fast response.

A common limitation is that response-time data are often reduced to a small number of summary statistics, such as a mean or a median [2,3]. These values are informative, but they do not describe the full shape of the empirical response-time distribution. In many practical cases, two game configurations may have similar average response times but very different dispersion, asymmetry, or tail behavior. This matters because the perceived difficulty of a task is not determined only by the central response time. It is also affected by the variability of responses and by the probability of unusually slow or unusually fast reactions. Moreover, extreme response-time observations and the procedures used to remove or correct them can substantially affect the statistical interpretation of experimental outcomes [4]. Previous work on reaction-time modeling has shown that these data are often non-Gaussian and that distributional information is important for interpreting cognitive and behavioral performance [5–8]. Studies comparing gamers and non-gamers also support the idea that video-game practice is related to measurable differences in reaction-time and working-memory tasks [9–11].

This paper starts from that observation and proposes a mathematical framework for predicting complete response-time distributions associated with simple video-game-like configurations. The experimental setting is intentionally controlled. A player reacts to an object that appears at a given screen position and moves under fixed conditions. Each trial configuration is described by a small set of parameters, including variables such as object size, color, initial coordinates, and spawning time. The goal is not to reproduce the full complexity of commercial video games. Rather, the aim is to isolate elementary interaction units that are present in many games and to build a model that can simulate the expected response-time distribution for new, untested configurations.

The problem is related to the broader field of adaptive game design. Dynamic difficulty adjustment (DDA) studies how a game can automatically modify parameters, scenarios, or non-player behavior in response to the player state, with the aim of avoiding boredom and frustration while preserving engagement [12–14]. Recent comparative studies also show that no single DDA strategy is uniformly superior across player-performance, affective, and experience-based criteria, which supports the need for adaptive and interpretable mechanisms [15,16]. Recent work has introduced machine learning and reinforcement learning approaches to DDA, including frameworks in which states, actions, and rewards are explicitly designed for action games [17], personalized systems combining imitation learning and reinforcement learning [18], fuzzy adaptive systems [19], MAPE-K self-adaptive architectures [20], chat-based estimation of perceived difficulty [21], and even large-language-model-based DDA in card games [22]. In parallel, procedural content generation via reinforcement learning has framed level design itself as a sequential decision problem, where an agent modifies game content in order to maximize a reward associated with playability, balance, control, or affective response [23–27]. Recent mixed-initiative frameworks further demonstrate how reinforcement learning agents and human designers can cooperate in the generation of diverse game scenarios [28].

Beyond dynamic difficulty adjustment and content generation, continuous response times and interaction traces may also have potential applications in security-oriented multimedia systems. When modeled at the individual or session level, these traces can provide behavioral signatures for detecting departures from a player's usual interaction pattern. Such deviations could support user-behavior anomaly detection, the identification of scripted or bot-assisted inputs, and anti-cheating mechanisms. In this context, the Lipschitz-regulated framework could provide interpretable expected ranges for behavioral descriptors and flag observations that are geometrically incompatible with the available history. However, the present experiment uses aggregated response-time distributions at the level of game configurations and does not evaluate security or player-authentication

tasks. Extending the model to continuous participant-specific trajectories therefore constitutes a future research direction. More broadly, recent work on the cryptanalysis and improvement of video cryptosystems illustrates the growing importance of mathematically controlled robustness and security analysis in multimedia applications [29].

The contribution of the present study differs from these approaches but remains complementary to them. Instead of adapting a whole game environment online, the present study focuses on the statistical modeling of response times generated by elementary game stimuli. The proposed method combines two components. The first one is a metric regression procedure based on Lipschitz functions. Each game configuration is represented as a point in a normalized metric space, and each statistical descriptor of the response-time distribution is treated as a real-valued function on that space. McShane–Whitney Lipschitz extensions are then used to estimate the descriptors of an unseen configuration from the observed ones [30,31]. This gives a geometrically controlled prediction rule, because each estimated descriptor is constrained by the empirical Lipschitz regularity of the available data. The use of Lipschitz-type extensions is also connected with previous work on machine learning and reinforcement learning in metric or quasi-metric settings [32,33].

The second component is a sequential regularization mechanism. Experimental configurations are incorporated into an active memory one by one. When a new observation produces an excessive increase in the empirical Lipschitz constant of one or more statistical descriptors, the algorithm does not simply accept the raw value. Instead, it replaces it with a convex combination of the observed value and a weighted recent average of previous values. In the basic version of the method, the coefficient of this convex combination is fixed. In the extended version, this coefficient, denoted by α , is selected through a reinforcement learning-inspired tabular mechanism. The agent observes a state that summarizes the current Lipschitz instability, selects a value of α from a finite action set, and receives a reward that penalizes both excessive loss of Lipschitz regularity and excessive distortion of the empirical observation. Thus, the procedure learns how strongly the system should trust new information in each instability regime. This tabular action-value mechanism follows the general principles of Q-learning [34].

After the statistical descriptors have been predicted for a new configuration, the method reconstructs a continuous probability density. In the implementation considered here, a Log-Normal distribution is fitted to the predicted descriptors. The parameters of this distribution are obtained by solving a nonlinear least-squares problem that simultaneously matches the predicted median, mean, and selected quantiles. Therefore, the output of the model is not only a point estimate of the response time, but a complete probability distribution. This objective is related to the general framework of distributional regression, in which the entire conditional distribution, rather than only its expected value, is modeled as a function of explanatory variables [35]. This distribution can be used to compare candidate screen configurations, simulate unobserved experimental situations, and support the design of new interaction patterns before collecting additional data. The use of distributional modeling is consistent with research showing that the complete shape of response-time distributions contains information that is lost when only central measures are considered [2,3], as well as with recent work in which non-centered or asymmetric distributions are used to represent performance data more faithfully than simple averages [36].

The resulting framework is a hybrid method. It combines Lipschitz-regulated regression on a normalized metric space, sequential anomaly control, reinforcement learning-based smoothing, and least-squares reconstruction of probability densities. Its main advantage is interpretability. The geometric part provides explicit upper and lower bounds for each descriptor, while the adaptive part gives a transparent rule for incorporating recent information without allowing isolated observations to destroy the regularity of the

model. This makes the approach suitable for small experimental datasets, where purely data-hungry models may be difficult to train and where it is important to preserve a clear mathematical description of the prediction mechanism.

The novelty of the proposed approach lies in predicting complete response-time distributions for unseen game configurations. The method does not introduce a new parametric distributional family; rather, it combines Lipschitz-regulated prediction of several distributional descriptors with a subsequent Log-Normal reconstruction. Unlike approaches that mainly fit or compare distributions under already observed conditions, the proposed method models several distributional descriptors simultaneously as Lipschitz functions of the game-design variables. It combines geometrically controlled McShane–Whitney bounds, adaptive reinforcement-based smoothing, and multi-feature probability-density reconstruction [30,31,34,35]. This provides an interpretable framework particularly suited to small experimental datasets.

Gaussian-process regression provides a related metric- and kernel-based alternative for prediction at unobserved configurations [37–39]. Both approaches exploit relationships between nearby points in the input space and provide bounds or intervals associated with their predictions. Their interpretation is nevertheless different. A Gaussian process specifies a covariance kernel and produces a posterior predictive mean and variance conditional on the selected kernel, observation noise model, and estimated hyperparameters. Its prediction intervals are therefore probabilistic credible intervals under the assumed stochastic model.

By contrast, the McShane–Whitney construction used here requires only a metric and a feature-specific Lipschitz constant. It produces deterministic lower and upper envelopes containing every extension compatible with the assumed Lipschitz regularity. These envelopes are geometric admissibility bounds rather than Bayesian credible intervals. Gaussian processes may provide smoother probabilistic predictions and flexible kernel-based structure, whereas the proposed framework prioritizes explicit geometric control, feature-wise interpretability, and sequential regulation.

Kernel density estimation provides another relevant comparison [40,41]. A basic kernel density estimator can flexibly approximate the response-time distribution observed at a particular configuration, but its result depends strongly on bandwidth selection, especially in the tails and with only 24 observations per level. Moreover, ordinary kernel density estimation does not predict how the distribution changes across unobserved game configurations. Conditional kernel density estimation can incorporate the configuration variables, but it requires the simultaneous selection of bandwidths in the response and covariate spaces. The effective number of observations available near a specific target configuration may therefore be very small when only 20 configurations are observed.

The proposed method instead reduces each empirical distribution to a small set of interpretable descriptors and transfers these descriptors across the configuration space under explicit Lipschitz constraints. It therefore exchanges some of the flexibility of Gaussian-process and kernel-density models for deterministic geometric control, transparent feature-wise predictions, and a sequential mechanism for limiting the influence of abrupt observations. This comparison should not be interpreted as establishing universal superiority, because the methods rely on different assumptions and provide different forms of uncertainty information.

The paper is organized as follows. Section 2 describes the experimental setting, the recorded variables, and the mathematical background. Section 3 presents the Lipschitz-regulated McShane–Whitney prediction procedure and the case studies based on a fixed smoothing coefficient. Section 4 introduces the adaptive Q-learning-inspired selection of the smoothing coefficient and illustrates the resulting model on two target configurations. Section 5 compares the fixed-coefficient and RL-adaptive models and reports the validation and sensitivity analyses. Section 6 compares the numerical and qualitative findings with

existing response-time modeling approaches and with performance measures used in adaptive video-game systems. Finally, Section 7 summarizes the main conclusions and discusses possible extensions of the approach.

2. Method

The methodology developed in this paper addresses two complementary dimensions of the study. The first concerns the experimental setting, including the design of the benchmark task, the definition of the game configurations, the collection of response-time data and the extraction of statistical descriptors from the empirical distributions. The second concerns the mathematical and computational model used to process these data. This model combines Lipschitz-regulated regression, McShane–Whitney extension formulas and a reinforcement learning-inspired adaptive mechanism for the sequential incorporation of new observations. Together, these two dimensions provide the basis for predicting complete response-time distributions for game configurations that have not been directly observed in the experiment.

2.1. Experimental Benchmark and Distributional Characterization

The basic model is built from the statistical characterization of the response-time distributions associated with each level of the benchmark video game. The experimental setting was designed as a simplified interaction task, intended to isolate elementary actions that are common in many video games: detecting a visual stimulus, tracking its position on the screen, aiming at it, and producing a motor response with the mouse. Although the task is deliberately simple, it provides a controlled framework for studying how changes in visual and spatial parameters affect the distribution of player response times.

The experiment was carried out with 24 participants. Each participant completed a sequence of 20 trials corresponding to different variations of the same basic game situation. The resulting dataset contains 480 valid response-time observations. Each participant contributed one observation to each of the 20 levels; consequently, every level contains exactly $n_i = 24$ response-time observations, with no missing response-time values in the analyzed dataset. In every trial, a visual stimulus appeared on the screen and moved across it. The player had to shoot the stimulus using the mouse as quickly and accurately as possible. The stimulus could appear in one of two possible colors, blue or purple, and its initial position changed from one level to another. Other properties of the stimulus and the scene, such as the object size, the initial coordinates and the spawning time, were also used to define the configuration of each level. The number of participants is not an intrinsic parameter of the proposed method, but a characteristic of the present experimental case study. The framework can be applied to other sample sizes, provided that each game configuration contains enough response-time observations to reliably estimate its distributional descriptors. Larger samples generally provide more stable estimates of the quantiles, Skewness, and empirical Lipschitz constants, whereas very small samples may require robust estimation or resampling procedures. The method can also be transferred to other game designs whenever their relevant configuration variables can be encoded in a suitable normalized metric space.

Each level is therefore represented by a finite vector of experimental descriptors. In the implementation used in this work, the configuration vector contains the object size, the color category, the initial spatial coordinates and the spawning-time parameter. These variables define the input space of the model. More precisely, every level is encoded as a point

$$x_i = (\text{size}_i, \text{color}_i, \text{coord}_{x,i}, \text{coord}_{y,i}, \text{coord}_{z,i}, \text{spawn}_i) \in \mathbb{R}^6.$$

These variables represent three complementary dimensions of video-game design, which are visual properties (object size and color), spatial configuration (the initial coordinates), and temporal configuration (spawning time). The movement conditions of the target and the basic shooting mechanic are kept fixed so that the model isolates the influence of these factors on the response-time distributions. The reinforcement learning component does not directly modify these design variables; rather, it adaptively selects the smoothing coefficient used when incorporating new observations.

Since these coordinates have different physical meanings and different numerical ranges, they are normalized before computing distances. This normalization makes it possible to compare levels using a common metric structure and prevents variables with larger numerical scales from dominating the geometry of the space.

The color variable has only two categories in the present experiment and is encoded as $c_i \in \{0, 1\}$. Consequently, its contribution to the squared Euclidean distance is

$$|c_i - c_j|^2 = \mathbb{I}_{\{c_i \neq c_j\}},$$

so it is equal to zero when two configurations have the same color and to one when their colors differ. Hence, the binary coding acts as a simple matching distance and does not impose an ordinal relation between the two colors. Indeed, exchanging the labels 0 and 1 leaves all pairwise distances unchanged. Since every continuous coordinate is independently normalized to $[0, 1]$, a color mismatch has the same maximum contribution as a full-range variation in any other individual coordinate. This equal-weight Euclidean metric provides a parsimonious representation of the controlled two-color experimental design. For settings involving more than two nominal categories or requiring variable-specific importance, one-hot encoding, Gower-type distances, or weighted mixed-variable metrics would be more appropriate.

For each level, the raw experimental data consist of the response times recorded across participants and trials. These response times are not reduced to a single average value. Instead, the empirical distribution associated with each level is summarized by a vector of statistical descriptors. The descriptors used in the basic model are the median, the 10th percentile, the 90th percentile, the interquartile range, the mean and a Skewness coefficient. Thus, each level x_i is associated with a feature vector:

$$y_i = (\text{Median}_i, Q10_i, Q90_i, \text{IQR}_i, \text{Mean}_i, \text{Skewness}_i).$$

This representation is richer than a standard regression target based only on the mean response time. It preserves information about the central tendency, the dispersion and the asymmetry of the response-time distribution. This is important because two levels may have similar mean response times while producing very different behavioral profiles. For example, one level may generate stable responses concentrated around the median, whereas another may produce a long-tailed distribution with a higher probability of delayed reactions.

2.1.1. Sampling Uncertainty of the Distributional Descriptors

Since each level contains only 24 response-time observations, the empirical quantiles and Skewness may be affected by substantial sampling variability. A non-parametric percentile bootstrap was therefore used to quantify the uncertainty of the level-specific descriptors. For each level, 5000 bootstrap samples of size 24 were generated by sampling the observed response times with replacement. The median, Q10, Q90, IQR, mean, and Skewness were recalculated for every bootstrap sample, and the 2.5th and 97.5th percentiles

of the resulting distributions were used as the limits of the corresponding 95% confidence intervals. The random-number seed was fixed at 20260711 for reproducibility.

Across the 20 levels, the median widths of the bootstrap confidence intervals were 0.1417 s for the median, 0.1142 s for Q10, 0.4092 s for Q90, 0.2437 s for the IQR, 0.1455 s for the mean, and 2.2905 for Skewness. Thus, Q90 and Skewness are substantially less stable than the central and lower-tail descriptors. The particularly wide intervals observed for some levels indicate that abrupt increases in the empirical Lipschitz constant of Q90 may reflect both genuine differences between game configurations and sampling variability in the estimated upper tail. The sequential smoothing mechanism mitigates the effect of these abrupt changes on the active memory, although it does not remove the underlying uncertainty of the empirical descriptor estimates.

Table 1 reports the level-specific results for the three descriptors most directly affected by small-sample uncertainty.

Table 1. Level-specific estimates and percentile-bootstrap 95% confidence intervals based on 5000 re-samples. Q10 and Q90 are expressed in seconds; Skewness is dimensionless.

Level	<i>n</i>	Q10 [95% CI]	Q90 [95% CI]	Skewness [95% CI]
Level 1	24	0.7433 [0.6250, 0.8883]	3.2333 [1.4800, 8.4265]	3.4492 [1.2287, 4.4656]
Level 2	24	0.6317 [0.5883, 0.6883]	0.9117 [0.7900, 1.1883]	1.7023 [0.1004, 2.5885]
Level 3	24	0.6550 [0.6150, 0.7050]	0.8733 [0.8116, 0.9166]	−0.0851 [−0.7586, 0.5141]
Level 4	24	0.7550 [0.7383, 0.8000]	1.1166 [0.9833, 1.2783]	0.9344 [0.1758, 1.6108]
Level 5	24	0.6833 [0.6433, 0.7150]	0.9616 [0.9000, 1.2617]	1.4359 [−0.0130, 2.1400]
Level 6	24	0.7600 [0.6933, 0.8166]	1.2250 [0.9167, 1.5566]	1.9426 [0.6000, 3.0190]
Level 7	24	0.7883 [0.7716, 0.8050]	1.3933 [1.0283, 1.8300]	1.7928 [0.7804, 2.9424]
Level 8	24	0.5867 [0.5433, 0.7050]	0.9950 [0.8233, 1.3116]	1.4975 [−0.0957, 2.3386]
Level 9	24	0.5667 [0.5267, 0.6000]	1.0066 [0.8067, 1.2116]	1.2907 [0.4281, 2.1852]
Level 10	24	0.5916 [0.5383, 0.7717]	1.1533 [0.9500, 1.5683]	1.6421 [−0.2502, 2.4071]
Level 11	24	0.5500 [0.5217, 0.6383]	0.9466 [0.8116, 1.1650]	1.1103 [0.0537, 1.8436]
Level 12	24	0.6866 [0.5916, 0.7833]	1.1166 [0.9667, 1.4516]	1.6263 [−0.2137, 2.4433]
Level 13	24	0.6667 [0.6383, 0.7550]	1.0983 [0.8450, 1.2117]	1.2362 [0.3682, 2.0175]
Level 14	24	0.6716 [0.6383, 0.7500]	0.9850 [0.9233, 1.5283]	2.2956 [−0.0792, 3.2052]
Level 16	24	0.5883 [0.5050, 0.6550]	0.8450 [0.7500, 1.0300]	1.0097 [−0.7183, 1.8230]
Level 15	24	0.6550 [0.6383, 0.7250]	1.0833 [0.9350, 1.6700]	2.3781 [0.0220, 3.1670]
Level 17	24	0.6217 [0.5216, 0.7217]	1.0066 [0.8966, 1.1017]	0.0194 [−0.6331, 0.7114]
Level 18	24	0.5550 [0.5267, 0.6166]	0.8450 [0.7417, 0.8967]	0.6663 [−0.0161, 1.4753]
Level 19	24	0.6217 [0.5767, 0.7433]	1.1400 [0.9566, 1.8433]	2.4205 [−0.1864, 3.1316]
Level 20	24	0.4833 [0.4600, 0.5167]	0.8583 [0.7066, 1.1200]	1.6413 [0.4187, 2.4732]

The first stage of the modeling process can therefore be understood as a distributional categorization of the benchmark levels. Each level is not classified by a categorical label, but by the empirical shape of its response-time distribution. The statistical descriptors act as numerical coordinates of this distributional profile. Once these descriptors have been extracted, the learning problem is reformulated as the estimation of several real-valued functions defined on the normalized metric space of level configurations. For every descriptor k , the model considers a function

$$f_k : X \longrightarrow \mathbb{R},$$

where X is the space of game configurations and $f_k(x_i)$ is the value of the corresponding statistical descriptor for level i .

This formulation provides the basis for the Lipschitz-regulated prediction procedure developed in the following sections. Instead of assuming that the response time itself varies smoothly across levels, the model assumes that the statistical descriptors of the response-time distribution vary in a controlled way with respect to the metric distance between level configurations. Under this assumption, the McShane–Whitney extension formulas can be used to predict the descriptors of a new, untested level from the descriptors observed in the experimental benchmark.

The purpose of this construction is not to reproduce the full complexity of a commercial video-game screen. Rather, the benchmark is designed to capture a minimal but meaningful interaction unit. By controlling the color, size, position and spawning conditions of the target, the experiment produces a structured set of response-time distributions that can be used to test whether the proposed method is able to extrapolate from observed configurations to new ones. In this sense, the benchmark plays a double role: it provides empirical data for fitting the model, and it defines a controlled design space in which new game configurations can be simulated before being tested experimentally.

2.1.2. Goodness-of-Fit Assessment of the Log-Normal Family

The suitability of the Log-Normal family was assessed using the empirical response-time observations from the 20 experimentally observed levels. Since a positive random variable T follows a Log-Normal distribution if and only if $\log(T)$ follows a Normal distribution, the 24 response times available for each level were log-transformed. Normal Q–Q plots were then constructed, and a Shapiro–Wilk test was applied independently to the log-transformed observations of each level.

The parameters of the level-specific Log-Normal distributions were estimated directly from the empirical observations as

$$\hat{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \log(T_{ij}), \quad \hat{\sigma}_i = \sqrt{\frac{1}{n_i} \sum_{j=1}^{n_i} (\log(T_{ij}) - \hat{\mu}_i)^2},$$

where $n_i = 24$ for every level. This assessment evaluates the suitability of the distributional family itself and is independent of the nonlinear least-squares reconstruction used later for the predicted descriptors.

Table 2 reports the Shapiro–Wilk statistics and both the unadjusted and Holm-adjusted p -values. At the unadjusted significance level of 0.05, the null hypothesis of Log-Normality is not rejected for 11 of the 20 levels and is rejected for 9 levels. Since the assessment involves 20 simultaneous tests, a Holm correction was also applied. After this correction, statistically significant departures remain only for level11 and level17.

Table 2. Goodness-of-fit assessment of the Log-Normal family for the 20 observed levels. The Shapiro–Wilk test is applied to the log-transformed response times.

Level	Shapiro–Wilk W	Raw p -Value	Holm-Adjusted p -Value
level1	0.8446	0.0017	0.0331
level2	0.9170	0.0502	0.5800
level3	0.9699	0.6637	1.0000
level4	0.9324	0.1102	0.9914
level5	0.9150	0.0453	0.5800
level6	0.8760	0.0069	0.1243
level7	0.8256	0.0008	0.0159
level8	0.9528	0.3110	1.0000
level9	0.9147	0.0446	0.5800
level10	0.9569	0.3787	1.0000
level11	0.9617	0.4726	1.0000
level12	0.9539	0.3284	1.0000
level13	0.9083	0.0324	0.4863
level14	0.8816	0.0090	0.1525
level16	0.9710	0.6929	1.0000
level15	0.8979	0.0194	0.3103
level17	0.9690	0.6414	1.0000
level18	0.9445	0.2056	1.0000
level19	0.9301	0.0980	0.9803
level20	0.9086	0.0329	0.4863

The corresponding Q–Q plots are provided in the data referenced in the Data Availability Statement. Most levels exhibit an approximately linear central pattern, whereas the largest departures occur in the upper tail, particularly for `leve11` and `leve17`. These findings indicate that the Log-Normal family provides a reasonable parsimonious approximation for many levels, but it should not be interpreted as a universally adequate data-generating model. In configurations presenting pronounced tail departures, the reconstructed density represents a smooth two-parameter summary of the predicted descriptors rather than an exact representation of the empirical distribution.

2.2. Mathematical Tools: Lipschitz Real-Valued Functions and McShane–Whitney Extensions

The prediction method proposed in this paper is based on a simple geometric idea. If two game configurations are close to each other, then the statistical descriptors of their response-time distributions should not change arbitrarily. This assumption can be expressed by means of Lipschitz real-valued functions.

Let (X, d) be a metric space, where each point $x \in X$ represents a game configuration. In the present setting, x encodes variables such as the size of the target, its color, its initial coordinates and its spawning time. A real-valued function

$$f : X \longrightarrow \mathbb{R}$$

is said to be K -Lipschitz if

$$|f(x) - f(y)| \leq Kd(x, y), \quad x, y \in X. \quad (1)$$

The constant $K \geq 0$ controls the maximum admissible variation of the function with respect to the metric distance. Small values of K correspond to regular functions, while large values indicate strong changes between nearby configurations.

In the present framework, the response-time distribution is not observed for every possible game configuration. Experimental data are available only for a finite subset

$$A = \{x_1, \dots, x_N\} \subset X.$$

For each observed level x_i , several statistical descriptors of the empirical response-time distribution are computed, including the median, the mean, the interquartile range and selected quantiles. Each descriptor can be interpreted as a real-valued function defined only on the observed subset A . Therefore, for a fixed descriptor, the analysis starts with a function

$$f : A \longrightarrow \mathbb{R}.$$

The problem is to predict the value of this descriptor at a new configuration $x^* \in X \setminus A$.

Lipschitz extension theory provides a natural tool for this purpose. If f is K -Lipschitz on A , then the classical McShane and Whitney formulas provide extensions from A to the whole metric space X preserving the same Lipschitz constant. The McShane extension is defined as

$$M_f(x) = \inf_{a \in A} \{f(a) + Kd(x, a)\}, \quad x \in X, \quad (2)$$

whereas the Whitney extension is defined as

$$W_f(x) = \sup_{a \in A} \{f(a) - Kd(x, a)\}, \quad x \in X. \quad (3)$$

Both functions extend f , that is,

$$M_f(a) = W_f(a) = f(a), \quad a \in A,$$

and both are K -Lipschitz on the whole space X . Moreover, for every $x \in X$ one has

$$W_f(x) \leq M_f(x). \quad (4)$$

Thus, at a new point x^* , the interval

$$\left[W_f(x^*), M_f(x^*) \right] \quad (5)$$

gives the range of values that are compatible with the Lipschitz regularity imposed by the already observed data.

The interval $[\mathcal{L}^{(k)}(\mathbf{x}^*), \mathcal{U}^{(k)}(\mathbf{x}^*)]$ is a deterministic Lipschitz-admissibility interval and should not be interpreted as a frequentist confidence interval or a Bayesian credible interval.

The notation in Equations (2) and (3) refers to a single generic scalar-valued function f . Accordingly, K is the Lipschitz constant associated with that particular function and is not a common constant shared by all the distributional descriptors. In the proposed model, each descriptor k defines a separate function

$$f_k : A \longrightarrow \mathbb{R},$$

with its own feature-specific empirical Lipschitz constant K_k . Thus, the generic notation (f, K) used in this section becomes (f_k, K_k) when the extension formulas are applied to the individual response-time descriptors.

In the algorithm developed below, these formulas are applied independently to each statistical descriptor of the response-time distribution. For example, the median, the mean and the quantiles are treated as different Lipschitz functions over the same metric space of game configurations. This makes it possible to predict not only a single response-time value, but a full set of distributional features associated with an untested level.

The following elementary result explains the role of the convex smoothing step used later in the sequential algorithm. It shows that convex combinations of two extensions have a Lipschitz constant bounded by the corresponding convex combination of their constants.

Proposition 1. *Let (X, d) be a metric space and let $A \subset X$. Let*

$$f : A \longrightarrow \mathbb{R}$$

be a function defined on the observed subset A . Suppose that $F_0, F_1 : X \longrightarrow \mathbb{R}$ are two extensions of f , that is,

$$F_0(a) = F_1(a) = f(a), \quad a \in A.$$

Assume that F_0 is K_0 -Lipschitz and that F_1 is K_1 -Lipschitz. For $\alpha \in [0, 1]$, define

$$F_\alpha(x) = \alpha F_1(x) + (1 - \alpha) F_0(x), \quad x \in X. \quad (6)$$

Then F_α is also an extension of f and it is K_α -Lipschitz with

$$K_\alpha \leq \alpha K_1 + (1 - \alpha) K_0. \quad (7)$$

Proof. For every $a \in A$,

$$F_\alpha(a) = \alpha F_1(a) + (1 - \alpha)F_0(a) = f(a),$$

so F_α extends f . Moreover, for any $x, y \in X$,

$$\begin{aligned} |F_\alpha(x) - F_\alpha(y)| &\leq \alpha |F_1(x) - F_1(y)| + (1 - \alpha) |F_0(x) - F_0(y)| \\ &\leq (\alpha K_1 + (1 - \alpha)K_0)d(x, y). \end{aligned}$$

Hence, F_α is Lipschitz and $K_\alpha \leq \alpha K_1 + (1 - \alpha)K_0$. $\square$

Proposition 1 has a direct interpretation in the proposed model. Suppose that a new experimental level produces a raw statistical value that extends the previous data but increases the empirical Lipschitz constant too much. This raw extension can be regarded as an extension with a large constant K_1 . On the other hand, the historical smoothed value, or a value selected inside the McShane–Whitney admissible interval, provides a more regular extension with a smaller constant K_0 . The convex correction

$$y_{\text{new}} = \alpha y_{\text{raw}} + (1 - \alpha)y_{\text{smooth}} \quad (8)$$

therefore produces an intermediate value whose Lipschitz behavior is controlled by the same convex principle.

This observation provides the mathematical basis for the smoothing rule used in the following algorithms. The parameter α regulates the compromise between two competing objectives. If α is close to one, the algorithm gives more weight to the newly observed experimental information. If α is close to zero, the algorithm gives more weight to the regularity inherited from the previous memory. The reinforcement learning layer introduced later can then be understood as a mechanism for selecting α sequentially, depending on the observed Lipschitz instability. In this way, the algorithm not only interpolates response-time descriptors but also controls how new information is incorporated without destroying the geometric regularity of the prediction model.

To finish this section, a short description is provided of the 1-Wasserstein metric. Since Lipschitz regression is applied on families of distributions, this metric provides a natural way to compare them. The 1-Wasserstein distance measures the minimum cost of transporting one probability distribution into another. In the one-dimensional case considered here, it can be written as

$$W_1(P, Q) = \int_0^1 |F_P^{-1}(t) - F_Q^{-1}(t)| dt,$$

where F_P^{-1} and F_Q^{-1} are the quantile functions of the two distributions.

3. The Unified Lipschitz-Regulated Reinforcement Learning and McShane–Whitney Prediction Framework

As stated in the Introduction, the proposed framework predicts complete response–time distributions for unobserved configurations. The input space may be high-dimensional, and the observations may arrive sequentially under noisy or non-stationary conditions.

The methodology combines geometric interpolation with an adaptive control layer inspired by reinforcement learning (RL). First, the algorithm extracts scalar distributional descriptors from the observed response times. It then models each descriptor as a Lipschitz function and predicts it using McShane–Whitney extensions.

3.1. Algorithmic Design and Feature Extraction

The algorithm follows a multistage sequential workflow. The initial dataset contains response-time observations indexed by distinct experimental regimes. In the first stage, each regime is represented as a point in the parameter space

$$\mathcal{X} \subset \mathbb{R}^d.$$

The input variables describe characteristics such as entity geometry, spatial position, and temporal triggers. Each numerical variable is normalized by its observed dynamic range. This prevents differences in scale from dominating the Euclidean-distance calculations.

The target space $\mathcal{Y}$ is obtained by summarizing the stochastic observations from each regime through the following vector of scalar distributional features:

$$\mathbf{y}_i = \left[f_{\text{Median}}^{(i)}, f_{Q10}^{(i)}, f_{Q90}^{(i)}, f_{\text{IQR}}^{(i)}, f_{\text{Mean}}^{(i)}, f_{\text{Skewness}}^{(i)} \right]^T \in \mathcal{Y} \quad (9)$$

The levels are processed according to the experimental Index variable stored in the raw dataset, rather than according to the numerical suffix of the level identifier. The resulting ingestion sequence is

$$\text{level11}, \dots, \text{level14}, \text{level16}, \text{level15}, \text{level17}, \dots, \text{level20}.$$

In particular, level16 precedes level15 because their experimental indices are 14 and 15, respectively. This ordering reproduces the sequence encoded by the data-acquisition system and was fixed before running the sequential regulation procedure. Since the active memory and the historical smoothing term depend on previously incorporated levels, the method is not theoretically invariant under arbitrary permutations of this sequence.

Before proceeding, the algorithm checks the variance of each feature across all regimes. If a feature has variance close to zero, it is treated as a static variable and is not included in the adaptive correction step. Features with relevant variation are passed to the active reinforcement learning regulation module. In the present dataset, the variance-screening step does not remove any descriptor, because all six features exhibit non-negligible variation. Consequently, all six descriptors are processed by the adaptive module.

3.2. Sequential Lipschitz Regulation via Partial Convex Smoothing

The consistency of the spatial interpolation depends on controlling the Lipschitz constants of the functions that describe the distributional features. For each descriptor k , the feature-specific empirical Lipschitz constant K_k over the active historical memory $\mathcal{I}$ is defined as the maximum pairwise variation ratio:

$$K_k = \max_{i,j \in \mathcal{I}, i \neq j} \frac{|y_i^{(k)} - y_j^{(k)}|}{\|\mathbf{x}_i - \mathbf{x}_j\|_2} \quad (10)$$

Here, “global” refers to the maximum being taken over all pairs in the current active memory for the fixed descriptor k ; it does not indicate a single Lipschitz constant shared by the six descriptors.

As data streams in sequentially, new candidate regimes may produce artificial spikes in K_k due to sampling noise, measurement errors, or unobserved environmental factors.

To control these artificial increases, each incoming regime n is first processed through a trial expansion. The algorithm computes the empirical Lipschitz constant that would result from adding the candidate observation.

For descriptor k , a correction is activated when

$$K_k^{(n)} - K_k^{(n-1)} > \epsilon_k,$$

where ϵ_k is the corresponding feature-specific tolerance.

The algorithm then constructs a historical reference value $y_{\text{smooth}}^{(k)}$. This reference is an exponentially weighted average of the active observations, with decay factor λ :

$$y_{\text{smooth}}^{(k)} = \sum_{m \in I_n} w_m y_m^{(k)}, \quad w_m = \frac{\lambda^{n-1-m}}{\sum_{\ell \in I_n} \lambda^{n-1-\ell}}, \quad m \in I_n. \quad (11)$$

Here, I_n denotes the set of indices of the active regimes stored in memory before processing the candidate regime n .

This normalization strictly guarantees that $\sum w_m = 1$, preventing scaling biases. Subsequently, a partial convex smoothing operation constructs a stabilized synthetic target value y_{new} ,

$$y_{\text{new}} = \alpha_k y_{\text{real}} + (1 - \alpha_k) y_{\text{smooth}}. \quad (12)$$

The coefficient $\alpha_k \in [0, 1]$ controls the balance between the current observation and the historical reference. Values close to one give greater weight to the new observation, whereas values close to zero produce stronger smoothing.

A separate rejection rule is applied when the instability is multivariate. Since six descriptors are active, the candidate regime is rejected when at least four descriptors simultaneously violate their tolerances. Rejected regimes are not incorporated into the active memory.

Since six descriptors are active in the present implementation, the “more than half” criterion corresponds to rejecting a candidate when at least four descriptors simultaneously violate their respective tolerances. This majority rule is intended to distinguish a genuinely multivariate incompatibility from an isolated descriptor fluctuation. In particular, a large jump affecting only Q90 or Skewness does not by itself cause rejection of the complete level. The threshold is therefore an interpretable supervisory design choice rather than a theoretically optimal constant. Its empirical sensitivity is examined in Section 5.5.

3.3. McShane–Whitney Extrapolation and Coherent Multi-Feature Optimization

Once the normalized parameters $\mathcal{X}_{\text{norm}}$ and the corrected descriptor values $\mathcal{Y}_{\text{mem}}$ have been obtained, the model predicts each descriptor separately at a new point $\mathbf{x}^*$. For descriptor k , let $K_{\text{final}}^{(k)}$ denote the empirical Lipschitz constant computed from the final regulated active memory. Equations (13) and (14) are the feature-wise counterparts of the generic McShane–Whitney formulas in Equations (2) and (3), with

$$f \longrightarrow f_k, \quad K \longrightarrow K_{\text{final}}^{(k)}, \quad A \longrightarrow \{\mathbf{x}_i : i \in \mathcal{I}_{\text{active}}\}.$$

The corresponding upper and lower envelopes are

$$\mathcal{U}^{(k)}(\mathbf{x}^*) = \min_{i \in \mathcal{I}_{\text{active}}} \left(y_i^{(k)} + K_{\text{final}}^{(k)} \|\mathbf{x}^* - \mathbf{x}_i\|_2 \right) \quad (13)$$

$$\mathcal{L}^{(k)}(\mathbf{x}^*) = \max_{i \in \mathcal{I}_{\text{active}}} \left(y_i^{(k)} - K_{\text{final}}^{(k)} \|\mathbf{x}^* - \mathbf{x}_i\|_2 \right) \quad (14)$$

If numerical drift forces an invalid geometric state where $\mathcal{L}^{(k)}(\mathbf{x}^*) > \mathcal{U}^{(k)}(\mathbf{x}^*)$, the interval collapses smoothly to its midpoint,

$$\mathcal{L}^{(k)}(\mathbf{x}^*) = \mathcal{U}^{(k)}(\mathbf{x}^*) = \frac{\mathcal{L}^{(k)}(\mathbf{x}^*) + \mathcal{U}^{(k)}(\mathbf{x}^*)}{2} \quad (15)$$

The geometrical McShane–Whitney point prediction is then established as $\hat{f}^{(k)}(\mathbf{x}^*) = \frac{1}{2}(\mathcal{U}^{(k)} + \mathcal{L}^{(k)})$, while the localized spatial uncertainty is quantified via the absolute Wasserstein-1 distance bound $\mathcal{W}_1^{(k)} \leq K_{\text{final}}^{(k)} \min_i \|\mathbf{x}^* - \mathbf{x}_i\|_2$.

The final stage reconstructs a continuous probability density supported on positive response times. The implementation uses the Log-Normal family assessed in Section 2.1.2. The goodness-of-fit results show that this family provides a parsimonious approximation for many, although not all, observed levels.

The parameters are estimated from several predicted descriptors simultaneously rather than from a single statistic. The McShane–Whitney stage provides predictions for the median, mean, Q10, and Q90. In contrast, a Log-Normal distribution $\text{LN}(\mu, \sigma)$ has only two free parameters.

The four predicted descriptors therefore cannot generally be matched exactly. The final values of μ and σ are selected by normalized nonlinear least squares. This produces the Log-Normal distribution whose theoretical descriptors are closest to the four geometric predictions.

Let the vector of predicted features obtained from the geometric stage be

$$\hat{\mathbf{r}}(\mathbf{x}^*) = \left(\hat{f}^{\text{Median}}(\mathbf{x}^*), \hat{f}^{\text{Mean}}(\mathbf{x}^*), \hat{f}^{\text{Q10}}(\mathbf{x}^*), \hat{f}^{\text{Q90}}(\mathbf{x}^*) \right)^T. \quad (16)$$

For a candidate Log-Normal law $Y \sim \text{LN}(\mu, \sigma)$, the corresponding theoretical feature map is

$$\Phi(\mu, \sigma) = \left(e^\mu, e^{\mu + \frac{\sigma^2}{2}}, F_{\text{LN}}^{-1}(0.10; \mu, \sigma), F_{\text{LN}}^{-1}(0.90; \mu, \sigma) \right)^T, \quad (17)$$

where the quantiles are computed in the script with `qlnorm`. Equivalently, if $z_\tau = \Phi_N^{-1}(\tau)$ denotes the standard normal quantile, then

$$F_{\text{LN}}^{-1}(\tau; \mu, \sigma) = \exp(\mu + \sigma z_\tau), \quad \tau \in \{0.10, 0.90\}. \quad (18)$$

Thus, each pair (μ, σ) gives one fitted median, one fitted mean, and two fitted tail quantiles. The role of the least-squares step is to select the pair that best matches this theoretical vector with the geometric prediction $\hat{\mathbf{r}}(\mathbf{x}^*)$.

The criterion used in the script is a normalized nonlinear least-squares objective. Defining the scale factors

$$s_j = \max(\hat{r}_j(\mathbf{x}^*), 10^{-4}), \quad (19)$$

the loss minimized by the algorithm is

$$J(\mu, \sigma) = \sum_{j=1}^4 \left(\frac{\Phi_j(\mu, \sigma) - \hat{r}_j(\mathbf{x}^*)}{s_j} \right)^2. \quad (20)$$

For the result tables, the variability underlying the aggregate loss is summarized through its four descriptor-wise contributions,

$$c_j = \left(\frac{\Phi_j(\mu, \sigma) - \hat{r}_j(\mathbf{x}^*)}{s_j} \right)^2, \quad j = 1, \dots, 4.$$

Their mean and sample standard deviation are reported as

$$\bar{c} = \frac{1}{4} \sum_{j=1}^4 c_j = \frac{J}{4}, \quad s_c = \sqrt{\frac{1}{3} \sum_{j=1}^4 (c_j - \bar{c})^2}.$$

This statistic measures the dispersion of the reconstruction loss across the four descriptors. It should not be interpreted as a sampling confidence interval, which is assessed separately through the bootstrap analysis of the empirical level descriptors.

This scaling is useful because it turns the four residuals into relative errors. It also prevents larger features, such as the mean or the upper quantile, from dominating the fit. As a result, the fitted PDF is not adjusted only to the median or only to the mean, but to a compromise between central tendency and distributional spread.

The parameter space is also controlled numerically. Since σ is the dispersion parameter of the Log-Normal law, values close to zero generate an almost degenerate density concentrated near a single point. To avoid this collapse, the objective function used by the script includes the following barrier rule:

$$\tilde{J}(\mu, \sigma) = \begin{cases} J(\mu, \sigma), & \sigma > 10^{-4}, \\ 10^{12}, & \sigma \leq 10^{-4}. \end{cases} \quad (21)$$

This penalty does not change the least-squares principle. It only prevents the optimizer from accepting an invalid or numerically unstable solution.

The search starts from the McShane–Whitney median and upper quantile predictions. The script first sets

$$m_0 = \max\left(\hat{f}^{\text{Median}}(\mathbf{x}^*), 0.001\right), \quad q_{0.90}^0 = \max\left(\hat{f}^{\text{Q90}}(\mathbf{x}^*), 1.05m_0\right), \quad (22)$$

and then defines the initial point

$$\mu_{\text{init}} = \log(m_0), \quad \sigma_{\text{init}} = \frac{\log(q_{0.90}^0/m_0)}{z_{0.90}}. \quad (23)$$

The factor $1.05m_0$ ensures that the initial upper quantile is strictly larger than the median. This avoids a zero or negative initial dispersion and gives the optimizer a meaningful starting point.

Starting from $(\mu_{\text{init}}, \sigma_{\text{init}})$, the script calls `optim` to solve

$$(\mu_p, \sigma_p) = \arg \min_{\mu \in \mathbb{R}, \sigma > 10^{-4}} \tilde{J}(\mu, \sigma), \quad (24)$$

using the relative tolerance `relTol` = 10^{-8} . The resulting values, stored as `mu_p` and `sig_p`, define the final predictive distribution

$$f_Y(y; \mu_p, \sigma_p) = \frac{1}{y\sigma_p\sqrt{2\pi}} \exp\left[-\frac{(\log y - \mu_p)^2}{2\sigma_p^2}\right], \quad y > 0. \quad (25)$$

After optimization, the script computes the output percentiles

$$\{P_5, P_{10}, P_{25}, P_{50}, P_{75}, P_{90}, P_{95}\} \quad (26)$$

using the fitted Log-Normal quantile function. It also prints an alignment diagnostic comparing the McShane–Whitney predicted features with the fitted analytical features:

$$\begin{aligned} \hat{f}^{\text{Median}} &\leftrightarrow e^{\mu_p}, & \hat{f}^{\text{Mean}} &\leftrightarrow e^{\mu_p + \sigma_p^2/2}, \\ \hat{f}^{\text{Q10}} &\leftrightarrow F_{\text{LN}}^{-1}(0.10; \mu_p, \sigma_p), & \hat{f}^{\text{Q90}} &\leftrightarrow F_{\text{LN}}^{-1}(0.90; \mu_p, \sigma_p). \end{aligned} \quad (27)$$

This diagnostic makes the final reconstruction easy to interpret. The normalized loss J and its descriptor-wise components measure how closely a single Log-Normal distribution reproduces the four McShane–Whitney predictions. The uncertainty bounds computed earlier are included in the visualization by perturbing the median and the upper quantile, producing a band around the optimized central PDF.

The current implementation reports these reconstruction diagnostics but does not automatically reject the Log-Normal fit when J exceeds a fixed critical value. A universal threshold is not imposed because J is a normalized, target-dependent discrepancy and, for a genuinely unobserved configuration, no empirical response-time sample is available for an independent Q–Q or goodness-of-fit test. Therefore, a large value of J should currently be interpreted as a warning that the four predicted descriptors are not well represented by a single two-parameter Log-Normal law, rather than as a formally calibrated rejection decision.

A future operational implementation will estimate a critical value J_{crit} from the empirical or cross-validated distribution of reconstruction losses. A warning would then be generated when the numerical optimizer fails to converge, when σ approaches its admissible boundary, or when $J > J_{\text{crit}}$. Such cases would trigger comparison with alternative positive-support families. However, multimodality cannot be identified reliably from only the median, mean, Q10, and Q90. A mixture-model extension will therefore require the prediction of a richer grid of quantiles or another higher-dimensional representation of distributional shape before estimating the parameters of Log-Normal or other finite mixtures.

3.4. Case Study: Fixed Smoothing Coefficient

To illustrate the behavior of the prediction framework before the reinforcement learning layer is activated, two target configurations are considered that share the same spatial parameters but differ in their spawning time. Both vectors belong to the same family

$$\vec{x}_{\text{new}} = (\text{size}, \text{color}, \text{coord-x}, \text{coord-y}, \text{coord-z}, \text{spawn}) \in \mathbb{R}^6, \quad (28)$$

and are processed with a fixed convex-smoothing coefficient $\alpha_k = 0.5$ for every statistical feature k . For the fixed observed dataset and the fixed level-ingestion order, the sequential regulation stage is performed only once. Consequently, the regulated active memory, the step-by-step status of the observed levels, and the trajectories of the empirical Lipschitz constants are common to the two target configurations considered below. The target configuration enters only in the subsequent McShane–Whitney evaluation through its distances to the regulated active memory.

3.4.1. Case 1: Spawn Time = 2.5 s

The first target configuration is

$$\vec{x}_{\text{new}} = (1.5, 1, 320.5, -195.2, 0.0, 2.5), \quad (29)$$

where color 1 denotes blue. The numerical implementation uses the feature-specific tolerances

$$\begin{aligned} \epsilon_{\text{Median}} &= 0.35, & \epsilon_{\text{Q10}} &= 0.30, & \epsilon_{\text{Q90}} &= 0.40, \\ \epsilon_{\text{IQR}} &= 0.35, & \epsilon_{\text{Mean}} &= 0.30, & \epsilon_{\text{Skewness}} &= 0.50. \end{aligned}$$

The historical exponential weighting uses $\lambda = 0.50$. These same tolerances are retained in the adaptive RL implementation described in Section 4.

Table 3 summarizes the outcome of the sequential Lipschitz-regulation stage. Steps 2–4, 11, and 15 are fully rejected, corresponding respectively to level2, level3, level4,

level11, and level16. The remaining levels are either *accepted* without modification or *smoothed* via a convex combination of the observed value and a weighted historical estimate, with the features that triggered the smoothing action listed in the Details column.

Table 3. Sequential Lipschitz regulation with fixed $\alpha_k = 0.5$ over the observed 20-level sequence.

Step	Level	Status	Details
Step 2	level2	REJECTED	(6/6 active features forced jumps)
Step 3	level3	REJECTED	(4/6 active features forced jumps)
Step 4	level4	REJECTED	(4/6 active features forced jumps)
Step 5	level5	SMOOTHED	Features: Q90, Mean
Step 6	level6	SMOOTHED	Feature: Q90
Step 7	level7	SMOOTHED	Feature: Q90
Step 8	level8	SMOOTHED	Features: Q90, Skewness
Step 9	level9	SMOOTHED	Features: Q90, Skewness
Step 10	level10	Accepted	–
Step 11	level11	REJECTED	(5/6 active features forced jumps)
Step 12	level12	SMOOTHED	Features: Q10, Skewness
Step 13	level13	Accepted	–
Step 14	level14	Accepted	–
Step 15	level16	REJECTED	(5/6 active features forced jumps)
Step 16	level15	Accepted	–
Step 17	level17	Accepted	–
Step 18	level18	Accepted	–
Step 19	level19	SMOOTHED	Features: Q90, Mean
Step 20	level20	Accepted	–

The evolution of the empirical Lipschitz constants K_k across sequential levels is shown in Figure 1. The Q90 feature exhibits the largest dynamic range, reflecting the sensitivity of the upper tail to changes in the spatial configuration.

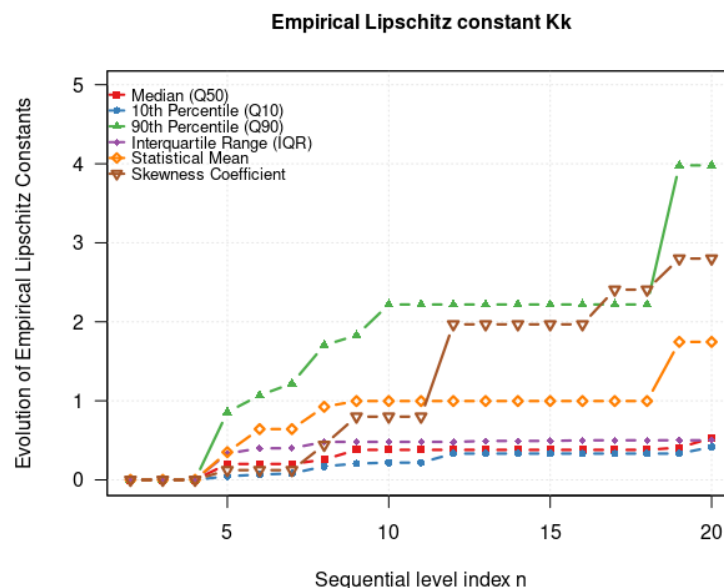


Figure 1. Evolution of the feature-specific empirical Lipschitz constants K_k over the observed level sequence under fixed $\alpha_k = 0.5$. Each curve represents one distributional descriptor and shows how its empirical Lipschitz constant evolves as observed levels are incorporated into the active memory.

Once the regulated memory is built, the McShane–Whitney extension formulas described in Section 2 are applied to predict the distributional descriptors at $\vec{x}_{\text{new}}$, and a Log-Normal distribution is fitted by the least-squares procedure described in Section 3.

Table 4 reports the alignment between the geometric McShane–Whitney predictions and the fitted Log-Normal values. The normalized Log-Normal reconstruction loss is $J = 0.044362$, indicating the degree of simultaneous agreement between the four predicted descriptors and the fitted distribution.

Table 4. Multi-feature optimized model output for spawn time = 2.5 s.

Feature	MW Prediction	Fitted Value
Median	0.7499	0.8768
Mean	0.9899	0.9355
Q10	0.6060	0.5526
Q90	1.4968	1.3910
Log-Normal hyperparameters: $\mu = -0.13151$, $\sigma = 0.36014$		
Normalized Log-Normal reconstruction loss: $J = 0.044362$		
Mean $\pm$ SD of descriptor-wise component losses: $\bar{c} \pm s_c \approx 0.01109 \pm 0.01185$		

The location parameter satisfies $e^\mu = e^{-0.13151} \approx 0.877$, which coincides with the fitted median. The scale parameter $\sigma = 0.36$ indicates moderate dispersion around the central value. The predictive PDF together with its geometric uncertainty band is displayed in Figure 2. The green curve is the fitted Log-Normal density $f(t \mid \mu, \sigma)$; the shaded region corresponds to the Wasserstein-type uncertainty interval $[\text{pdf}_{\text{lo}}, \text{pdf}_{\text{hi}}]$ derived from the Lipschitz bound $\Delta_k = K_k \cdot d(\vec{x}_{\text{new}}, \vec{x}_{NN})$; and the dashed blue curve is the empirical distribution of the nearest historical neighbor $\vec{x}_{NN}$. The overlap of the blue curve with the shaded region provides direct visual evidence that the actual most-similar observation is compatible with the model's geometric uncertainty.

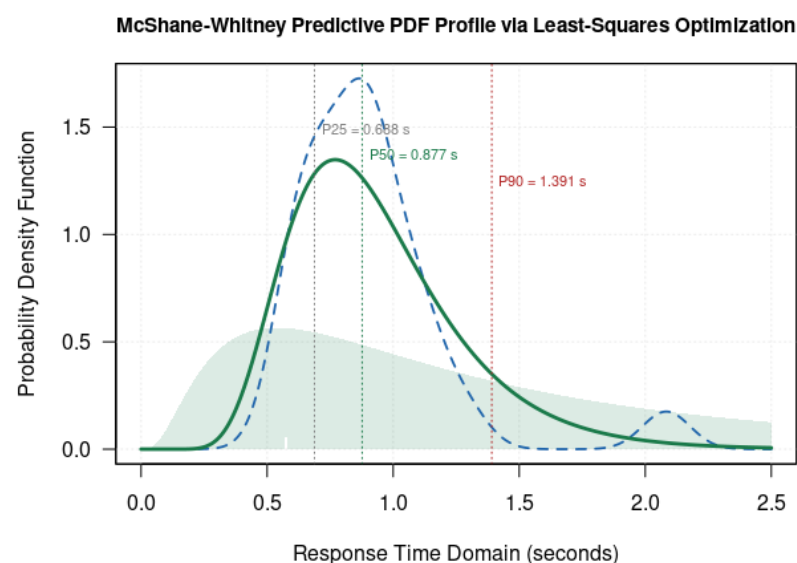


Figure 2. McShane–Whitney predictive PDF for spawn time = 2.5 s (fixed $\alpha_k = 0.5$). Green line: fitted Log-Normal density. Shaded band: Wasserstein-type geometric uncertainty interval. Dashed blue line: empirical distribution of the nearest neighbor.

3.4.2. Case 2: Spawn Time = 1 s

The second configuration shares all spatial parameters but sets the spawning time to 1 s:

$$\vec{x}_{\text{new}} = (1.5, 1, 320.5, -195.2, 0.0, 1.0). \quad (30)$$

The observed dataset, the level-ingestion order, the tolerances ϵ_k , the exponential factor $\lambda = 0.50$, and the coefficient $\alpha_k = 0.50$ are unchanged. Therefore, the regulated memory

and the Lipschitz-constant trajectories are the same as in Case 1. Only the target-specific McShane–Whitney predictions and the subsequent Log-Normal reconstruction change.

The final Log-Normal fit is reported in Table 5. Relative to Case 1, the predicted response times are shorter and less dispersed ($\sigma = 0.208$ vs. 0.360), and the normalized Log-Normal reconstruction loss decreases to $J = 0.002857$, indicating greater simultaneous agreement between the four McShane–Whitney descriptors and the fitted distribution. The corresponding predictive PDF is shown in Figure 3.

Table 5. Multi-feature optimized model output for spawn time = 1 s.

Feature	MW Prediction	Fitted Value
Median	0.7039	0.7333
Mean	0.7474	0.7493
Q10	0.5752	0.5618
Q90	0.9804	0.9571
Log-Normal hyperparameters: $\mu = -0.31024$, $\sigma = 0.20787$		
Normalized Log-Normal reconstruction loss: $J = 0.002857$		
Mean $\pm$ SD of descriptor-wise component losses: $\bar{c} \pm s_c \approx 0.000714 \pm 0.000734$		

The comparison between Cases 1 and 2 illustrates a key feature of the proposed framework: the entire predicted distribution, not only the mean, changes with the input configuration. As the spawn time decreases, both the predicted central tendency and the upper tail shrink, and the fitted distribution becomes more concentrated.

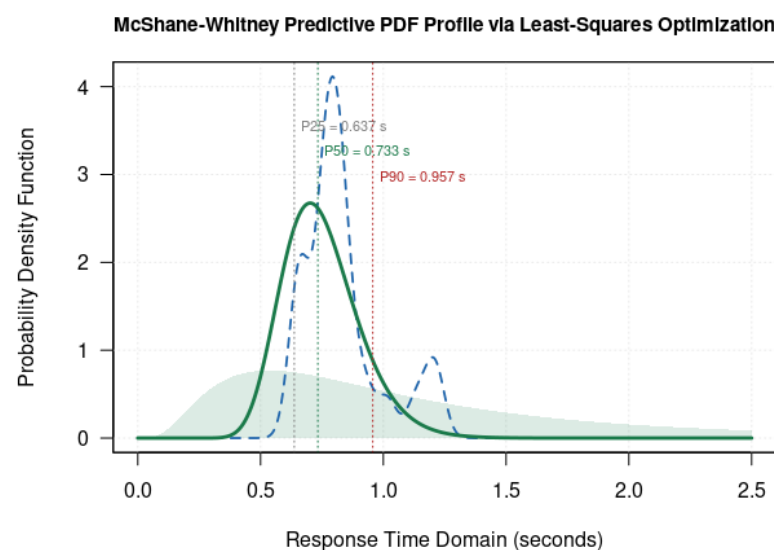


Figure 3. McShane–Whitney predictive PDF for spawn time = 1 s (fixed $\alpha_k = 0.5$). Same legend as Figure 2.

4. Adaptive Reinforcement-Based Selection of the Smoothing Parameter

The examples in Section 3.4 use the fixed coefficient $\alpha_k = 0.5$ for every descriptor and level. This choice provides a transparent illustration of the Lipschitz-regulation mechanism. However, the appropriate degree of smoothing may depend on both the descriptor and the observed instability. The adaptive extension considered in this section therefore selects α_k sequentially through a tabular reinforcement-learning mechanism.

At each incoming regime n , the algorithm evaluates the relative Lipschitz instability produced by the candidate observation via the ratio (31):

$$\rho_{n,k} = \frac{\Delta K_k}{\epsilon_k}, \quad \Delta K_k = K_k^{(n)} - K_k^{(n-1)}, \quad (31)$$

where ϵ_k is the admissible tolerance for feature k . The ratio $\rho_{n,k}$ is discretized into four qualitative states according to

$$s_{n,k} = \begin{cases} \text{stable,} & \rho_{n,k} \leq 1, \\ \text{mild jump,} & 1 < \rho_{n,k} \leq 1.5, \\ \text{medium jump,} & 1.5 < \rho_{n,k} \leq 2.5, \\ \text{severe jump,} & \rho_{n,k} > 2.5. \end{cases}$$

The stable state corresponds exactly to an increment within the feature-specific tolerance. A mild jump exceeds that tolerance by no more than 50%, a medium jump represents an exceedance between 50% and 150%, and a severe jump exceeds it by more than 150%. The use of four coarse states limits Q-table sparsity in the present small dataset while retaining a distinction between moderate and pronounced instability. These boundaries are interpretable discretization choices rather than estimated optimal cut-off values.

The agent selects the smoothing coefficient from the discrete action set

$$\alpha_k \in \{0, 0.25, 0.50, 0.75, 1\}, \quad (32)$$

where $\alpha_k = 1$ corresponds to fully accepting the empirical observation and $\alpha_k = 0$ corresponds to discarding it entirely in favour of the historical smoothed estimate. Intermediate values implement the convex compromise

$$y_{\text{new}}^{(k)} = \alpha_k y_{\text{real}}^{(k)} + (1 - \alpha_k) y_{\text{smooth}}^{(k)}. \quad (33)$$

Action selection uses an ϵ -greedy policy over a feature-specific Q-table. With small probability the algorithm explores a uniformly random α_k ; otherwise it selects the action with the highest expected reward for the observed instability state. After applying the selected smoothing, the algorithm evaluates a reward (34) that penalizes two competing undesirable effects—excessive growth of the post-update Lipschitz constant and excessive distortion of the empirical observation:

$$r_{n,k} = - \left[w_K \left(\frac{\max\{0, K_{k,\text{post}} - K_{k,\text{before}} - \epsilon_k\}}{\epsilon_k} \right)^2 + w_D \left(\frac{|y_{\text{new}}^{(k)} - y_{\text{real}}^{(k)}|}{\max\{|y_{\text{real}}^{(k)}|, 10^{-4}\}} \right)^2 \right]. \quad (34)$$

The Q-values are then updated by the standard temporal-difference rule (35):

$$Q(s_{n,k}, \alpha_k) \leftarrow (1 - \eta) Q(s_{n,k}, \alpha_k) + \eta \left[r_{n,k} + \gamma \max_{\alpha'} Q(s_{n+1,k}, \alpha') \right], \quad (35)$$

where η is the learning rate and γ is the temporal discount factor. The numerical implementation uses

$$\eta = 0.20, \quad \gamma = 0.30, \quad \epsilon_{\text{expl}} = 0.10, \quad w_K = 0.70, \quad w_D = 0.30,$$

where ϵ_{expl} denotes the exploration probability of the ϵ -greedy action-selection rule. As in the previous section, the feature-specific Lipschitz tolerances are $\epsilon_{\text{Median}} = 0.35$, $\epsilon_{Q10} = 0.30$, $\epsilon_{Q90} = 0.40$, $\epsilon_{\text{IQR}} = 0.35$, $\epsilon_{\text{Mean}} = 0.30$, and $\epsilon_{\text{Skewness}} = 0.50$.

The exponentially weighted historical average uses $\lambda = 0.50$. Each feature-specific Q-table contains four instability states and five possible actions. The entries are initialized at zero, with a small numerical prior of 10^{-6} assigned to $\alpha = 0.50$, and the random exploration mechanism uses seed 123 to ensure reproducibility.

The present implementation performs a single online pass through the ordered sequence of 20 levels. Level 1 initializes the active memory, and Levels 2–20 generate 19 sequential decision opportunities. No repeated training episodes, offline pre-training, or independent evaluation pass is used for the Q-table. Accordingly, the final table should be interpreted as an exploratory one-pass policy estimate for this particular data sequence, rather than as an independently validated or globally optimal reinforcement learning policy.

During this single sequential pass, the algorithm updates a feature-dependent action-value table. Rather than applying a uniform correction, the online mechanism selects different smoothing coefficients according to the observed feature and instability regime.

For reproducibility, the numerical settings of the adaptive module are summarized in Table 6.

Table 6. Hyperparameters used in the sequential Lipschitz-regulation and adaptive Q-table procedure.

Parameter	Value	Description
η	0.20	Learning rate
γ	0.30	Temporal discount factor
ϵ_{expl}	0.10	Exploration probability
w_K	0.70	Lipschitz-growth penalty weight
w_D	0.30	Observation-distortion penalty weight
λ	0.50	Historical exponential-decay factor
ϵ_{Median}	0.35	Median tolerance
ϵ_{Q10}	0.30	Q10 tolerance
ϵ_{Q90}	0.40	Q90 tolerance
ϵ_{IQR}	0.35	IQR tolerance
ϵ_{Mean}	0.30	Mean tolerance
$\epsilon_{\text{Skewness}}$	0.50	Skewness tolerance
Action set	$\{0, 0.25, 0.50, 0.75, 1\}$	Available α_k values
State boundaries	1, 1.5, 2.5	Instability-state cut-offs
Rejection rule	At least 4 of 6	Simultaneous violations
Random seed	123	Reproducible exploration
Training passes	1	Single sequential pass

The resulting procedure can be interpreted as a Lipschitz-regulated sequential prediction method with a reinforcement learning-inspired adaptive smoothing layer. The McShane–Whitney extension formulas and the final least-squares Log-Normal reconstruction described in Section 3 remain unchanged, but they now operate on a memory space whose corrections have been selected through the evolving online Q-table rather than through a fixed value of α_k .

4.1. Case Study: RL Smoothing Module

The RL-extended algorithm is applied to two target configurations with the same spatial coordinates as those considered in Section 3.4, but with spawn times equal to 1 s and 1.5 s.

Figure 4 shows the evolution of the Lipschitz constants under the adaptive policy. Table 7 summarizes the status of each step; the alpha values selected by the Q-table are omitted here for brevity but are representative of the range in (32).

The RL regulation stage is performed once on the observed 20-level sequence. Therefore, the RL-regulated active memory, the sequential status, the learned Q-table, and the feature-specific Lipschitz trajectories are common to both target configurations considered below. The target point is introduced only after this stage, when the McShane–Whitney envelopes are evaluated.

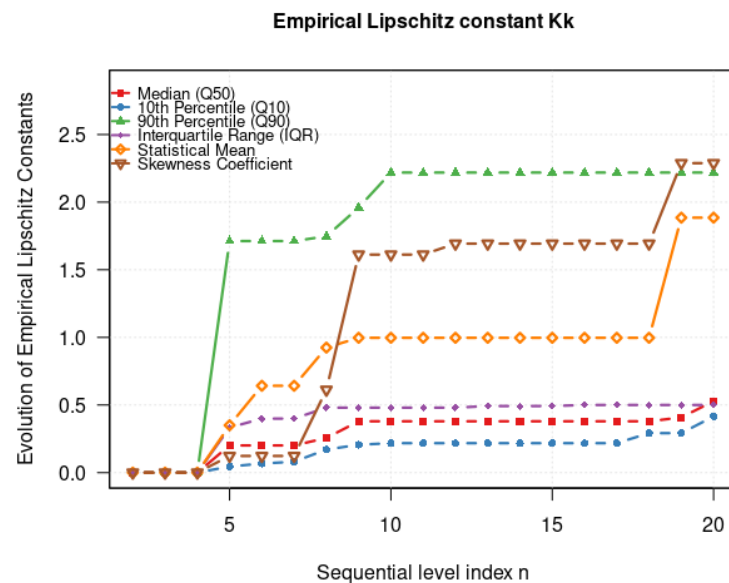


Figure 4. Evolution of the empirical Lipschitz constants under the RL smoothing policy.

In Table 7 the label SMOOTHED_RL indicates that the descriptor entered the adaptive regulation step after exceeding its tolerance. When the selected action is $\alpha_k = 1$, the empirical value is retained and no numerical smoothing is applied.

Table 7. Sequential RL-regulated status over the observed 20-level sequence.

Step	Level	Status	Active Features
Step 2	level2	REJECTED	6/6 active features forced jumps
Step 3	level3	REJECTED	4/6 active features forced jumps
Step 4	level4	REJECTED	4/6 active features forced jumps
Step 5	level5	SMOOTHED_RL	Q90, Mean
Step 6	level6	Accepted	–
Step 7	level7	Accepted	–
Step 8	level8	SMOOTHED_RL	Q90, Skewness
Step 9	level9	SMOOTHED_RL	Q90, Skewness
Step 10	level10	Accepted	–
Step 11	level11	REJECTED	5/6 active features forced jumps
Step 12	level12	SMOOTHED_RL	Q10, Skewness
Step 13	level13	Accepted	–
Step 14	level14	Accepted	–
Step 15	level16	REJECTED	5/6 active features forced jumps
Step 16	level15	Accepted	–
Step 17	level17	SMOOTHED_RL	Skewness
Step 18	level18	Accepted	–
Step 19	level19	SMOOTHED_RL	Mean, Skewness
Step 20	level20	Accepted	–

4.1.1. Spawn Time = 1 s

The predictive distribution for the target configuration with spawn time = 1 s is displayed in Figure 5. Table 8 reports the corresponding McShane–Whitney predictions, fitted Log-Normal descriptors, and normalized reconstruction loss. The fitted distribution has approximately $P_{25} = 0.634$ s, $P_{50} = 0.750$ s, and $P_{90} = 1.034$ s. The dashed curve represents the empirical distribution of the nearest active-memory configuration, level113.

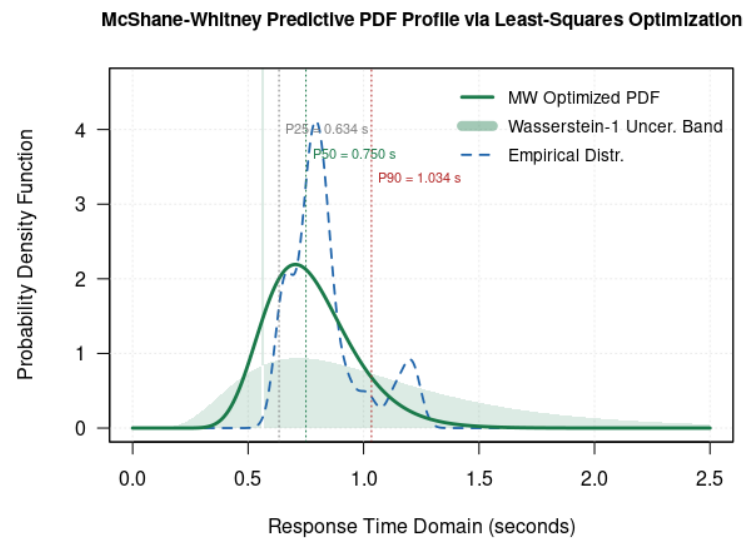


Figure 5. McShane–Whitney predictive PDF for spawn time = 1 s under the RL smoothing policy. Same legend as Figure 2.

Table 8. Multi-feature RL-optimized model output for spawn time = 1 s.

Feature	MW Prediction	Fitted Value
Median	0.7040	0.7504
Mean	0.7467	0.7743
Q10	0.5753	0.5445
Q90	1.1025	1.0342
Log-Normal parameters: $\mu = -0.28710$, $\sigma = 0.25027$		
Normalized Log-Normal reconstruction loss: $J = 0.012406$		
Mean $\pm$ SD of descriptor-wise component losses: $\bar{c} \pm s_c \approx 0.003101 \pm 0.001312$		

4.1.2. Spawn Time = 1.5 s

Table 9 presents the feature-alignment diagnostics for spawn time = 1.5 s, and Figure 6 shows the corresponding predictive PDF. The fitted Log-Normal parameters are $\mu = -0.21294$ and $\sigma = 0.36047$, and the normalized reconstruction loss is $J = 0.062218$. The differences with the spawn-time-1 s case arise from the target-specific McShane–Whitney predictions obtained from the same RL-regulated active memory.

Table 9. Multi-feature RL-optimized model output for spawn time = 1.5 s.

Feature	MW Prediction	Fitted Value
Median	0.6992	0.8082
Mean	0.8229	0.8625
Q10	0.5809	0.5092
Q90	1.4963	1.2828
Log-Normal parameters: $\mu = -0.21294$, $\sigma = 0.36047$		
Normalized Log-Normal reconstruction loss: $J = 0.062218$		
Mean $\pm$ SD of descriptor-wise component losses: $\bar{c} \pm s_c \approx 0.01555 \pm 0.00957$		

The analyzed examples show that changing the spawning-time coordinate of the target configuration modifies the McShane–Whitney descriptor predictions and the resulting Log-Normal reconstruction. Within the two RL examples, the fitted median increases from approximately 0.750 s for spawn time = 1 s to approximately 0.808 s for spawn time = 1.5 s, while the fitted upper quantile increases from approximately 1.034 s to 1.283 s. These illustrative cases do not establish a general monotone relationship between spawning time and every predicted descriptor or reconstruction loss.

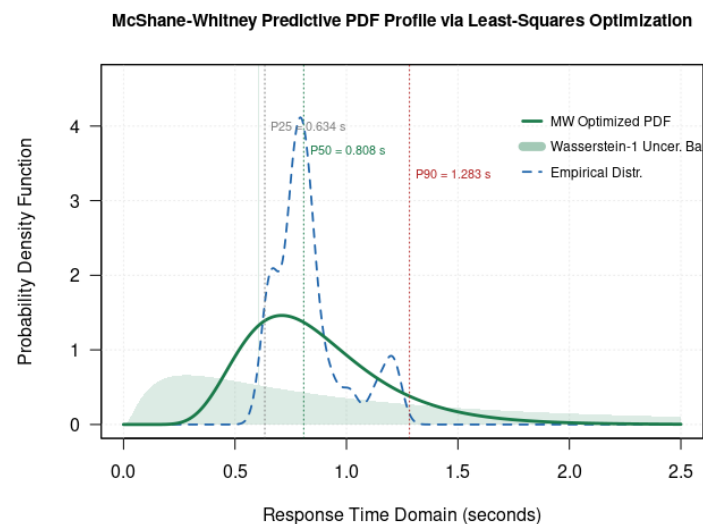


Figure 6. McShane–Whitney Same legend as Figure 2. Predictive PDF for spawn time = 1.5 s under the RL smoothing policy.

5. Comparative Analysis: Fixed $\alpha = 0.5$ Versus RL-Adaptive α

This section compares the fixed-smoothing algorithm (Section 3.4) and the RL-adaptive algorithm (Section 4) on a common target configuration:

$$\vec{x}_{\text{new}} = (1.5, 1, 0.3, 1.2, 0.0, 0.8), \quad (36)$$

where the coordinates correspond, in order, to size, color, coord-x, coord-y, coord-z, and spawn.

5.1. Fixed $\alpha_k = 0.5$

Table 10 lists the sequential status under the fixed smoothing coefficient. Steps 2–4, 11, and 15 are rejected, corresponding to level2, level3, level4, level11, and level16. The remaining levels are either accepted or smoothed in the descriptor-specific combinations reported in the table, involving Q90, mean, Q10, and Skewness.

Table 11 reports the final Log-Normal fit, and Figure 7 displays the predictive PDF with its geometric uncertainty band.

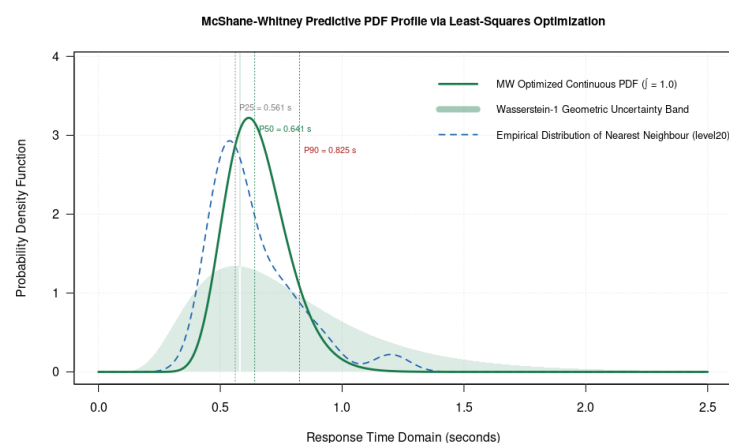


Figure 7. Predictive PDF under fixed $\alpha_k = 0.5$. Same legend as Figure 2. The uncertainty band is a Wasserstein-type interval derived from the Lipschitz bounds: feature-wise perturbations of the median and the upper quantile are propagated to the fitted Log-Normal density, yielding a plausible range of response-time distributions compatible with the local geometric uncertainty of the model.

Table 10. Sequential Lipschitz regulation over the observed 20-level sequence with fixed $\alpha_k = 0.5$.

Step	Level	Status	Details
Step 2	level2	REJECTED	6/6 active features forced jumps
Step 3	level3	REJECTED	4/6 active features forced jumps
Step 4	level4	REJECTED	4/6 active features forced jumps
Step 5	level5	SMOOTHED	Features: Q90, Mean
Step 6	level6	SMOOTHED	Feature: Q90
Step 7	level7	SMOOTHED	Feature: Q90
Step 8	level8	SMOOTHED	Features: Q90, Skewness
Step 9	level9	SMOOTHED	Features: Q90, Skewness
Step 10	level10	Accepted	–
Step 11	level11	REJECTED	5/6 active features forced jumps
Step 12	level12	SMOOTHED	Features: Q10, Skewness
Step 13	level13	Accepted	–
Step 14	level14	Accepted	–
Step 15	level16	REJECTED	5/6 active features forced jumps
Step 16	level15	Accepted	–
Step 17	level17	Accepted	–
Step 18	level18	Accepted	–
Step 19	level19	SMOOTHED	Features: Q90, Mean
Step 20	level20	Accepted	–

Table 11. Multi-feature optimized model output with fixed $\alpha_k = 0.5$.

Feature	MW Prediction	Fitted Log-Normal
Median	0.6162	0.6412
Mean	0.6368	0.6538
Q10	0.5157	0.4982
Q90	0.8583	0.8253
Log-Normal parameters: $\mu = -0.44436$, $\sigma = 0.19695$		
Normalized Log-Normal reconstruction loss: $J = 0.004994$		
Mean $\pm$ SD of descriptor-wise component losses: $\bar{c} \pm s_c \approx 0.001249 \pm 0.000411$		

5.2. RL-Adaptive α_k

The final one-pass alpha policy obtained after processing the 20-level sequence is reported in Table 12. Notably, the IQR and mean features assign $\alpha_k = 0$ to medium and severe instability states, meaning the final online Q-table tends to discard empirical IQR and mean observations entirely when Lipschitz violations are strong, favoring historical continuity over raw data fidelity for those features. Conversely, Q10 retains a relatively high α_k across all states, reflecting the greater stability of the lower tail.

Table 12. Learned alpha policy by feature and Lipschitz instability state.

Feature	Stable	Mild Jump	Medium Jump	Severe Jump
Median	0.50	0.50	0.25	0.50
Q10	0.50	0.75	0.50	0.50
Q90	0.50	0.25	0.50	0.00
IQR	0.50	0.00	0.00	0.00
Mean	0.50	0.50	0.00	0.00
Skewness	0.50	0.00	0.25	0.25

Table 13 details the RL step-by-step outcomes. Compared with the fixed- α schedule in Table 10, the RL policy accepts Levels 6 and 7 outright rather than smoothing them, and it applies targeted corrections to Skewness at Steps 17 and 19 that the fixed

policy does not trigger. These differences arise during the sequential processing of the observed levels because the evolving Q-table assigns different values of α_k to the observed instability states. They are determined before the target configuration is submitted to the McShane–Whitney predictor.

Table 13. Sequential RL-regulated status and selected α_k values over the observed 20-level sequence.

Step	Level	Status	Details/Alpha Parameters
Step 2	level2	REJECTED	(6/6 active features forced jumps) alpha: Median = 0.50, Q10 = 0.50, Q90 = 0.50, IQR = 0.50, Mean = 0.50, Skewness = 0.50
Step 3	level3	REJECTED	(4/6 active features forced jumps) alpha: Q90 = 0.00, IQR = 0.50, Mean = 0.00, Skewness = 0.50
Step 4	level4	REJECTED	(4/6 active features forced jumps) alpha: Q90 = 0.25, IQR = 0.50, Mean = 0.75, Skewness = 0.50
Step 5	level5	SMOOTHED_RL	Features: Q90, Mean alpha: Q90 = 1.00, Mean = 0.50
Step 6	level6	Accepted	–
Step 7	level7	Accepted	–
Step 8	level8	SMOOTHED_RL	Features: Q90, Skewness alpha: Q90 = 0.00, Skewness = 0.75
Step 9	level9	SMOOTHED_RL	Features: Q90, Skewness alpha: Q90 = 0.50, Skewness = 1.00
Step 10	level10	Accepted	–
Step 11	level11	REJECTED	(5/6 active features forced jumps) alpha: Median = 0.00, Q10 = 0.00, Q90 = 0.75, IQR = 0.75, Mean = 1.00
Step 12	level12	SMOOTHED_RL	Features: Q10, Skewness alpha: Q10 = 0.25, Skewness = 0.25
Step 13	level13	Accepted	–
Step 14	level14	Accepted	–
Step 15	level16	REJECTED	(5/6 active features forced jumps) alpha: Median = 0.75, Q90 = 0.25, IQR = 0.75, Mean = 0.25, Skewness = 0.00
Step 16	level15	Accepted	–
Step 17	level17	SMOOTHED_RL	Feature: Skewness alpha: Skewness = 0.00
Step 18	level18	Accepted	–
Step 19	level19	SMOOTHED_RL	Features: Mean, Skewness alpha: Mean = 0.25, Skewness = 0.75
Step 20	level20	Accepted	–

Table 14 presents the final Log-Normal fit under the RL policy, and Figure 8 displays the corresponding predictive PDF. The normalized Log-Normal reconstruction loss increases from $J = 0.004994$ under the fixed- α model to $J = 0.011107$ under the RL-adaptive model. The corresponding mean descriptor-wise component losses are 0.001249 ± 0.000411 and 0.002777 ± 0.000648 , respectively. At the same time, the Q90 prediction shifts upward from 0.8583 to 0.9166, indicating that the RL policy preserves more of the upper-tail information by accepting Level 6 and Level 7 without smoothing.

Table 15 provides a direct step-by-step comparison between the two regulation methods over the same observed level sequence.

Three patterns emerge from the comparison. First, the rejection decisions at Steps 2–4, 11, and 15 are identical for both methods, indicating that, for the analyzed sequence, the rejection decisions are unchanged by the choice of smoothing policy. Second, the RL policy accepts Levels 6 and 7 outright rather than smoothing them. At these steps, the evolving Q-table selects actions that retain the empirical Q90 values, whereas the fixed rule applies $\alpha_k = 0.5$ whenever the corresponding tolerance is exceeded. Third, the RL policy activates

Skewness corrections at Steps 17 and 19 that the fixed policy does not trigger, since the Q-table assigns non-trivial penalty to Skewness distortions in those instability regimes.

Table 14. Multi-feature RL-optimized model output for $\vec{x}_{\text{new}}$ in (36).

Feature	MW Prediction	Fitted Log-Normal	Absolute Error
Median	0.6162	0.6495	0.0333
Mean	0.6368	0.6655	0.0287
Q10	0.5157	0.4896	0.0261
Q90	0.9166	0.8618	0.0548
Log-Normal parameters: $\mu = -0.43150$, $\sigma = 0.22062$			
Normalized Log-Normal reconstruction loss: $J = 0.011107$			
Mean $\pm$ SD of descriptor-wise component losses: $\bar{c} \pm s_c \approx 0.002777 \pm 0.000648$			

Table 15. Step-by-step comparison between fixed $\alpha = 0.5$ and RL-adaptive α over the observed 20-level sequence.

Step	Level	Fixed $\alpha=0.5$	RL with Adaptive α
Step 2	level2	REJECTED (6/6 jumps)	REJECTED (6/6 jumps)
Step 3	level3	REJECTED (4/6 jumps)	REJECTED (4/6 jumps)
Step 4	level4	REJECTED (4/6 jumps)	REJECTED (4/6 jumps)
Step 5	level5	SMOOTHED: Q90, Mean	SMOOTHED_RL: Q90 ($\alpha = 1.00$), Mean ($\alpha = 0.50$)
Step 6	level6	SMOOTHED: Q90	Accepted
Step 7	level7	SMOOTHED: Q90	Accepted
Step 8	level8	SMOOTHED: Q90, Skewness	SMOOTHED_RL: Q90 ($\alpha = 0.00$), Skewness ($\alpha = 0.75$)
Step 9	level9	SMOOTHED: Q90, Skewness	SMOOTHED_RL: Q90 ($\alpha = 0.50$), Skewness ($\alpha = 1.00$)
Step 10	level10	Accepted	Accepted
Step 11	level11	REJECTED (5/6 jumps)	REJECTED (5/6 jumps)
Step 12	level12	SMOOTHED: Q10, Skewness	SMOOTHED_RL: Q10 ($\alpha = 0.25$), Skewness ($\alpha = 0.25$)
Step 13	level13	Accepted	Accepted
Step 14	level14	Accepted	Accepted
Step 15	level16	REJECTED (5/6 jumps)	REJECTED (5/6 jumps)
Step 16	level15	Accepted	Accepted
Step 17	level17	Accepted	SMOOTHED_RL: Skewness ($\alpha = 0.00$)
Step 18	level18	Accepted	Accepted
Step 19	level19	SMOOTHED: Q90, Mean	SMOOTHED_RL: Mean ($\alpha = 0.25$), Skewness ($\alpha = 0.75$)
Step 20	level20	Accepted	Accepted

The comparison of the final predictive distributions in this section is intended as an illustrative analysis of one target configuration. By contrast, the fixed and RL sequential regulation schedules are constructed before target evaluation and are independent of that target point. General claims about whether the adaptive policy reduces smoothing interventions would require independent observed sequences or datasets, repeated random seeds, and an independently validated policy-training procedure. The leave-one-level-out analysis evaluates the McShane–Whitney prediction component and should not be interpreted as an independent validation of the RL policy.

These differences propagate to the final predictive distribution: the RL version produces a slightly wider PDF, with a higher Q90 and a larger σ , together with a moderately

higher normalized reconstruction loss. It also produces more differentiated and interpretable feature-specific correction decisions.

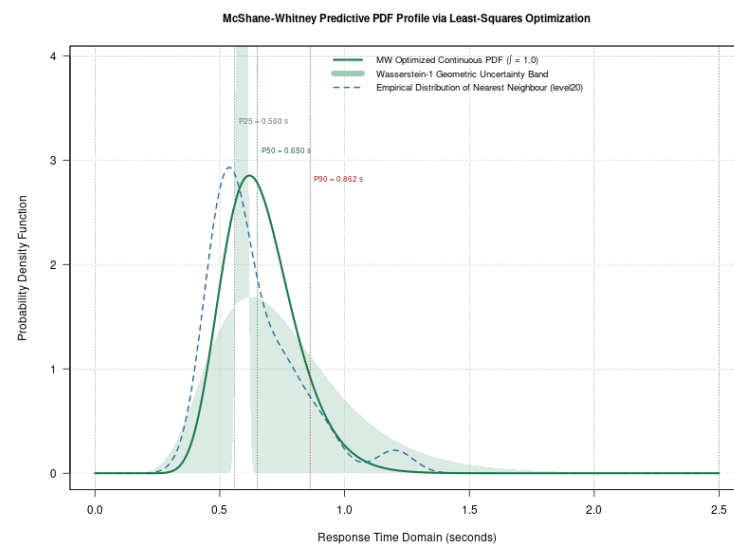


Figure 8. Predictive PDF under the RL-adaptive policy for $\vec{x}_{\text{new}}$ in (36). The predicted distribution is slightly wider than the fixed- α counterpart, reflecting the higher Q90 prediction produced by the learned smoothing policy.

Therefore, the improvement provided by the RL-adaptive version should not be interpreted as a uniform reduction in numerical fitting error. Its main potential benefit is structural: it permits feature-specific correction decisions according to the observed instability regime. During the one-pass processing of the observed level sequence, this results in the direct acceptance of Levels 6 and 7 and in different corrections for the upper-tail and Skewness descriptors. These observations should not be interpreted as evidence of a general reduction in regularization. In the present comparison, the normalized reconstruction loss is higher than for the fixed- α model, revealing an explicit trade-off between distributional fit, data fidelity, and Lipschitz regularity. The results are consistent with previous findings on asymmetric response-time distributions, while extending them to the prediction of unobserved game configurations under geometrically controlled uncertainty.

5.3. Leave-One-Level-Out Validation

A leave-one-level-out validation was performed to assess genuine out-of-sample descriptor-prediction error, as distinct from the internal Log-Normal reconstruction loss J reported for the synthetic target configurations. The deterministic fixed- α model was selected for this validation because it provides a reproducible reference implementation of the McShane–Whitney prediction framework.

At each of the 20 folds, one complete game level and all its response-time observations were removed. The normalization ranges, sequential regularization process, active memory, and empirical Lipschitz constants were then recomputed using only the remaining 19 levels. Finally, the six distributional descriptors of the omitted level were predicted and compared with their empirical values. No response-time information from the omitted level was used during model construction.

A one-nearest-neighbor regression model was also evaluated as a simple baseline. At each fold, the six configuration variables were normalized using only the 19 training levels. The omitted level was then matched to the nearest training configuration under the same Euclidean metric used by the proposed framework, and the complete descriptor vector of that nearest neighbor was used as the prediction. The value $k = 1$ was selected to

avoid introducing an additional tuning procedure in a dataset containing only 20 levels. This baseline therefore provides a direct comparison with a local prediction rule that uses the same input representation but does not apply Lipschitz regularization or McShane–Whitney bounds.

For each descriptor k , prediction accuracy was measured through

$$\text{MAE}_k = \frac{1}{20} \sum_{i=1}^{20} |\hat{y}_{i,k} - y_{i,k}|, \quad (37)$$

and

$$\text{RMSE}_k = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (\hat{y}_{i,k} - y_{i,k})^2}, \quad (38)$$

where $y_{i,k}$ denotes the empirical value of descriptor k for the omitted level and $\hat{y}_{i,k}$ denotes its McShane–Whitney prediction.

Table 16 summarizes the leave-one-level-out results.

Table 16. Leave-one-level-out prediction errors for the deterministic fixed- α Lipschitz model and the one-nearest-neighbor baseline across the 20 experimentally observed game levels. Lower values indicate better predictive performance.

Descriptor	Lipschitz MAE	1-NN MAE	Lipschitz RMSE	1-NN RMSE
Median	0.1270	0.1308	0.1633	0.1781
Q10	0.0919	0.0873	0.1095	0.1086
Q90	0.7224	0.6858	1.1342	1.1576
IQR	0.1543	0.1719	0.2193	0.2550
Mean	0.3041	0.3117	0.4870	0.5059
Skewness	0.4981	1.2413	0.5708	1.4775

The proposed Lipschitz model yields a lower error than the 1-NN baseline in nine of the twelve descriptor–metric comparisons. Improvements are observed for the median, IQR, mean, and especially Skewness. For Skewness, the Lipschitz model reduces the MAE from 1.2413 to 0.4981 and the RMSE from 1.4775 to 0.5708, indicating that the regularization procedure substantially limits the influence of unstable local estimates. The 1-NN baseline performs slightly better for Q10 and for the MAE of Q90, while the Lipschitz model retains a slightly lower Q90 RMSE.

These results do not establish uniform superiority of the proposed method. Rather, they show that Lipschitz regulation improves most of the analyzed distributional descriptors while retaining explicit McShane–Whitney admissible intervals and geometric uncertainty information, neither of which is provided by the nearest-neighbor baseline. Both methods confirm that Q90 remains the most difficult descriptor to predict in the present small-sample dataset.

5.4. Sensitivity to the Level-Ingestion Order

The sequential regulation procedure depends on the order in which levels are incorporated into the active memory. An order-sensitivity analysis was therefore performed for the deterministic fixed- α model and the common target configuration considered in Section 5. The deterministic model was used in order to isolate the effect of ingestion order from the additional randomness introduced by the ϵ -greedy RL mechanism.

Two complementary checks were conducted. First, the levels were sorted by their numerical labels, thereby exchanging only level15 and level16 relative to the experimental Index order. Second, 1000 random permutations of Levels 2–20 were generated while

retaining Level 1 as the common initial memory state. The random-number seed was fixed at 20260711 for reproducibility.

Table 17 summarizes the corresponding order-sensitivity results.

Table 17. Sensitivity of the fixed- α McShane–Whitney predictions to the level-ingestion order. Random-order results are reported as mean $\pm$ standard deviation over 1000 permutations.

Descriptor	Experimental Index Order	Numerical Level Order	Random Orders
Median	0.6162	0.6162	0.6413 $\pm$ 0.0624
Q10	0.5157	0.5157	0.5253 $\pm$ 0.0522
Q90	0.8583	0.8583	1.0514 $\pm$ 0.2660
IQR	0.2125	0.2125	0.2095 $\pm$ 0.0209
Mean	0.6368	0.6368	0.7525 $\pm$ 0.1441
Skewness	1.6413	1.6413	1.7272 $\pm$ 0.2005

Exchanging level15 and level16 does not alter any of the predicted descriptors. Thus, the particular non-numerical position identified by the reviewer is not responsible for the reported prediction. Nevertheless, the random-permutation analysis confirms that the sequential procedure is not fully order invariant. Median, Q10, IQR, and Skewness exhibit comparatively moderate variability, whereas mean and especially Q90 are more sensitive to the ingestion sequence. This result is consistent with the previously observed instability of the upper tail.

It should be noted, however, that this order dependence is consistent with the sequential-learning rationale of the proposed method. The system updates its active memory and its smoothing decisions as observations become available, so the current state necessarily depends on the previously processed configurations. When the ingestion order reproduces the actual experimental acquisition sequence, this dependence is not merely an implementation artifact, but reflects the temporal structure through which the model learns from the data. Nevertheless, when no natural or experimentally justified order exists, order-averaged predictions or an order-independent batch formulation should be considered.

Consequently, the predictions reported in the case studies should be interpreted conditionally on the experimental acquisition order. In the present study, this order is meaningful because it follows the sequence recorded by the data-acquisition system. For applications lacking a natural sequential structure, an order-averaged prediction over several permutations or an order-independent batch regularization strategy would provide a more robust alternative.

5.5. Sensitivity to the Rejection and Instability Thresholds

A sensitivity analysis was conducted to assess the supervisory rejection rule and the discretization of the relative instability ratio. The deterministic fixed- α implementation was used to isolate threshold effects from the stochastic exploration component of the RL module. The common target configuration considered in Section 5 was used throughout.

First, the minimum number of simultaneous descriptor violations required for complete rejection was varied from three to five. The current rule requires four of the six active descriptors to violate their feature-specific tolerances. Table 18 reports the corresponding predictions.

The three- and four-descriptor rules produce identical outcomes because no candidate in the analyzed sequence generates exactly three simultaneous violations. Requiring five violations is more permissive, reduces the number of rejected levels from five to three, and mainly affects the upper-tail and mean predictions. Median, Q10, and IQR remain unchanged.

Table 18. Sensitivity to the minimum number of simultaneous descriptor violations required for rejecting a level.

Minimum Violations	Rejected Levels	Median	Q10	Q90	IQR	Mean
3 of 6	5	0.6162	0.5157	0.8583	0.2125	0.6368
4 of 6	5	0.6162	0.5157	0.8583	0.2125	0.6368
5 of 6	3	0.6162	0.5157	0.9166	0.2125	0.6805

The instability-state boundaries were also varied. In addition to the implemented cut-offs (1, 1.5, 2.5), a narrower discretization (1, 1.25, 2.0) and a wider discretization (1, 2.0, 3.0) were examined. Table 19 reports the allocation of the 114 descriptor-level transitions.

Table 19. Sensitivity of the instability-state allocation to alternative boundaries. The three values identify the upper limits of the stable, mild, and medium states, respectively.

State Boundaries	Stable	Mild	Medium	Severe
(1, 1.25, 2.0)	78	4	9	23
(1, 1.50, 2.5)	78	8	10	18
(1, 2.00, 3.0)	78	13	9	14

The stable versus unstable classification is invariant across the three schemes because the first boundary remains fixed at $\rho_{n,k} = 1$, which is determined directly by the tolerance ϵ_k . The alternative boundaries only redistribute the unstable events among mild, medium, and severe categories. The implemented discretization provides an intermediate allocation and avoids concentrating most violations in either the mild or severe state. Nevertheless, the state-specific RL decisions remain conditional on these selected boundaries, and future applications with larger datasets should estimate or validate the cut-off values independently.

6. Discussion

This section compares the present findings with related response-time modeling and video-game performance approaches from both numerical and methodological perspectives. The discussion is organized at two complementary levels. First, direct numerical comparisons are considered within the present dataset: the fixed- α Lipschitz model is compared with the one-nearest-neighbor baseline under the same input representation, distance, and leave-one-level-out folds, while the fixed and RL-adaptive regulation policies are compared for the same target configuration. Second, the results are discussed in relation to performance measures used in response-time modeling and adaptive video-game systems. Numerical findings from previous studies are used only for contextual comparison, since their datasets, tasks, samples, and outcome variables differ from those considered here.

6.1. Direct Numerical Comparison with the Prediction Baseline

Table 16 provides the most direct evaluation of the proposed prediction method. The fixed- α Lipschitz model and the 1-NN baseline were evaluated on the same 20 leave-one-level-out folds, using the same normalized configuration variables and the same Euclidean metric. To make the magnitude and direction of the differences more explicit, Table 20 reports the relative error reduction of the Lipschitz model with respect to 1-NN. For an error measure E , the relative reduction is defined as

$$100 \frac{E_{1NN} - E_{\text{Lipschitz}}}{E_{1NN}}.$$

Positive values favor the Lipschitz model, whereas negative values favor 1-NN.

Table 20. Relative error reduction of the fixed- α Lipschitz model with respect to the one-nearest-neighbor baseline. Positive values favor the Lipschitz model.

Descriptor	Relative MAE Reduction	Relative RMSE Reduction
Median	2.9%	8.3%
Q10	−5.3%	−0.8%
Q90	−5.3%	2.0%
IQR	10.2%	14.0%
Mean	2.4%	3.7%
Skewness	59.9%	61.4%

The Lipschitz model produces a lower error in nine of the twelve descriptor–metric comparisons. Both MAE and RMSE are reduced for median, IQR, mean, and Skewness. The largest difference occurs for Skewness, for which the MAE decreases from 1.2413 to 0.4981 and the RMSE decreases from 1.4775 to 0.5708. These values correspond to relative reductions of approximately 59.9% and 61.4%, respectively. The result is consistent with the role of Lipschitz regulation: abrupt local changes in a comparatively unstable shape descriptor are not transferred directly to the omitted configuration.

The advantage is not uniform. The 1-NN baseline produces slightly lower MAE and RMSE for Q10 and a slightly lower MAE for Q90. The Lipschitz model produces a slightly lower Q90 RMSE, but Q90 remains the descriptor with the largest absolute prediction errors under both methods. This result agrees with the bootstrap analysis, which identifies the upper tail and Skewness as the least stable distributional characteristics in the available samples. Consequently, the results support descriptor-dependent improvements rather than a general claim that Lipschitz regulation is uniformly more accurate than local nearest-neighbor prediction.

A different comparison is provided by the fixed and RL-adaptive versions in Section 5. Tables 11 and 14 show that, for the common target configuration, the McShane–Whitney predictions of median, mean, and Q10 remain unchanged, whereas Q90 increases from 0.8583 under the fixed policy to 0.9166 under the RL-adaptive policy. This represents an increase of approximately 6.8% in the predicted upper quantile. The fitted Log-Normal dispersion parameter also increases from $\sigma = 0.19695$ to $\sigma = 0.22062$.

The larger upper quantile and dispersion reflect the direct acceptance of Levels 6 and 7 by the adaptive policy and the resulting preservation of more upper-tail information. However, the normalized reconstruction loss increases from $J = 0.004994$ to $J = 0.011107$. The RL-adaptive version therefore does not improve the internal parametric fit in this example. Instead, it changes the balance among empirical fidelity, Lipschitz regularity, and Log-Normal reconstruction accuracy. The comparison in Table 15 also shows that the main benefit of the adaptive policy is the ability to make descriptor-specific regulation decisions, rather than a uniform reduction in either prediction error or the number of smoothing interventions.

6.2. Similar Findings in Response-Time and Adaptive-Game Research

The difficulty observed for Q90 and Skewness is consistent with previous response-time research. Behavioral response-time distributions are commonly asymmetric and positively skewed, and their upper tails contain information that is not represented by the mean or median alone [2–4]. The present bootstrap and leave-one-level-out analyses lead to the same qualitative conclusion: central and lower-tail descriptors are comparatively stable, whereas upper-tail and shape descriptors require greater caution.

A second similarity concerns the absence of a universally superior adaptive strategy. Comparative DDA studies have shown that the relative performance of an adaptive method depends on whether the evaluation criterion is objective game performance, adjustment accuracy, engagement, affect, or subjective player experience [15,16]. The present findings exhibit the same type of metric dependence. Lipschitz regulation improves most descriptor–metric combinations, but not Q10 or the Q90 MAE. Similarly, the RL-adaptive policy preserves more upper-tail information but produces a higher Log-Normal reconstruction loss.

Related findings have been reported for engagement-oriented DDA. Xue et al. [42] formulated difficulty adjustment as an optimization problem based on player progression and reported improvements of up to 9% in player engagement. The outcome optimized in that study was player retention and continued progression, rather than the statistical distribution of behavioral response times. Nevertheless, both approaches replace an exclusively heuristic adjustment rule with an explicit quantitative objective.

Darzi et al. [43] compared several DDA methods based on performance, personality, and physiological information. The all-data method achieved adjustment accuracies of 62.5% for ball speed and 55.0% for paddle size, compared with 53.7% and 46.2%, respectively, for the performance-only method. However, no significant overall differences among the DDA methods were found for motivation or flow. Adjustment accuracy for ball speed was correlated with enjoyment ($r = 0.38$) and pressure ($r = 0.43$). This result is similar to the present finding that an improvement under one numerical criterion does not necessarily imply improvement under every other criterion. In particular, preservation of the upper tail by the RL policy does not imply a lower reconstruction loss.

Zook and Riedl [44] developed a temporal player model for predicting changes in player performance and found a significant relationship between the predicted performance ratings and subjective difficulty. This supports the general use of objective behavioral measures to inform adaptive game design. The present method differs by predicting several descriptors of a complete response-time distribution instead of producing a single performance or difficulty estimate.

6.3. Dissimilar Findings and Differences in the Evaluation Measures

The principal difference from conventional DDA systems concerns both the predicted outcome and the role of reinforcement learning. Existing adaptive systems frequently use score, accuracy, win rate, deaths, level-completion time, retention, or self-reported difficulty as performance measures. The resulting model then modifies a game variable, such as opponent strength, paddle size, object speed, or level difficulty [12,17,42,43]. By contrast, the present method predicts median, mean, Q10, Q90, IQR, and Skewness for an unobserved game configuration.

The reinforcement-learning component also has a different function. It does not directly change target size, position, spawning time, movement, or another difficulty parameter. Instead, it selects the smoothing coefficient used when an observed descriptor destabilizes the empirical Lipschitz structure. Consequently, the RL output is a descriptor-specific data-regulation decision, not a direct gameplay intervention.

Table 21 summarizes the main numerical and non-numerical similarities and differences. The external numerical results are included to identify the measures and scales used by related systems; they are not direct head-to-head benchmarks against the present dataset.

Table 21. Comparison between the proposed distributional framework and performance measures used in related video-game systems.

Study or Approach	Evaluated Measure	Numerical or Qualitative Finding	Relationship with the Present Method
Present Lipschitz model	MAE and RMSE for Median, Q10, Q90, IQR, Mean, and Skewness	Lower error than 1-NN in nine of twelve comparisons; approximately 59.9% MAE reduction and 61.4% RMSE reduction for Skewness	Direct prediction of several distributional descriptors for unobserved game configurations, with explicit McShane–Whitney admissibility bounds
Xue et al. [42]	Engagement, progression, stay time, and win-rate-based difficulty	Up to 9% improvement in player engagement	Optimizes retention and progression rather than the distribution of behavioral response times
Darzi et al. [43]	Difficulty-adjustment accuracy, motivation, enjoyment, pressure, and flow	All-data adjustment accuracy of 62.5% and 55.0% versus performance-only accuracy of 53.7% and 46.2%; no significant overall difference in motivation or flow	Shows that higher adjustment accuracy does not necessarily produce a uniform improvement in subjective experience
Zook and Riedl [44]	Predicted player performance and subjective difficulty	Significant relationship between predicted performance ratings and subjective difficulty	Produces a player-performance estimate, whereas the present model predicts central, dispersion, tail, and shape characteristics
Gaussian-process regression [37–39]	Predictive mean, variance, and probabilistic interval	Kernel-dependent probabilistic prediction under a stochastic model	Provides probabilistic uncertainty, whereas McShane–Whitney formulas provide deterministic geometric bounds under an assumed Lipschitz constant
Kernel and conditional-density estimation [40,41]	Estimated empirical or conditional response-time density	Flexible density estimation whose accuracy depends on bandwidth selection and the effective local sample size	Provides greater distributional flexibility, whereas the present method prioritizes descriptor-level interpretability and geometric control in a small configuration sample

The comparison identifies a complementary rather than substitutive role for the proposed method. Conventional game-performance measures quantify success, progression, retention, or subjective experience. Gaussian-process and kernel-based methods provide alternative forms of probabilistic prediction or density estimation. The Lipschitz method contributes a different capability: the distributional profile of an unobserved configuration is predicted under explicit metric regularity constraints.

A complete evaluation of a future adaptive game could combine the predicted response-time descriptors with conventional outcomes such as accuracy, score, completion, perceived difficulty, engagement, and flow. Such a study would determine whether changes in Q90, IQR, or Skewness are associated with meaningful differences in player experience. The present experiment does not establish those relationships. It establishes that distributional behavioral information can be predicted from a limited number of observed configurations and that the benefit of Lipschitz or RL-based regulation depends on the descriptor and evaluation criterion considered.

7. Conclusions

The framework proposed in this paper provides an interpretable method for predicting complete response-time distributions in controlled video-game-like tasks. Instead of reducing each level to a single average response time, the method represents each empirical distribution through several statistical descriptors, including the median, the mean, selected quantiles, dispersion and Skewness. These descriptors are then treated as real-valued functions on a normalized metric space of game configurations.

The main contribution of the approach is the combination of Lipschitz-regulated regression with McShane–Whitney extension formulas. This gives a geometrically controlled prediction rule: each predicted descriptor is constrained by the empirical Lipschitz regularity of the observed data. As a result, the model does not produce an unconstrained numerical estimate, but a prediction compatible with the metric structure of the experimental benchmark.

A second contribution is the sequential smoothing mechanism used to incorporate new observations into the active memory. When a candidate level produces an excessive increase in the Lipschitz constant, the algorithm regularizes the corresponding descriptor through a convex combination of the observed value and a recent historical estimate. In the extended version, the coefficient of this combination is selected through a reinforcement learning-inspired tabular mechanism. This allows the model to balance two competing objectives: preserving geometric regularity and incorporating recent empirical information. The adaptive Q-table is currently updated during a single ordered pass through the 20 observed levels and is not evaluated on an independent sequence. Therefore, the RL-related results are exploratory and may depend on the order of the levels, the initial Q-table, and the selected hyperparameters. Repeated training episodes, order-randomized sensitivity analyses, and validation on independent game configurations are required before making general claims about policy convergence or out-of-sample RL performance.

The final reconstruction currently uses a Log-Normal family. The goodness-of-fit analysis shows that Log-Normality is not rejected for 11 of the 20 levels at the unadjusted 5% significance level. After Holm correction, significant departures remain for level11 and level17. The Q–Q plots indicate that the principal discrepancies occur in the upper tails. Consequently, the Log-Normal density should be interpreted as a parsimonious positive-support reconstruction of the predicted descriptors rather than as a universally valid distributional model.

The current implementation reports the normalized reconstruction loss J , but it does not yet apply an automatically calibrated rejection threshold. Future work should estimate

such a threshold from cross-validated losses and issue a warning when the optimization fails, a boundary solution is obtained, or J is unusually large. In these cases, Gamma, Weibull, shifted Log-Normal, ex-Gaussian, or finite-mixture families should be compared using cross-validated reconstruction loss and empirical goodness-of-fit criteria. Reliable modeling of multimodal distributions will additionally require a richer vector of predicted quantiles or distributional descriptors.

The method is especially useful in small experimental settings, where the number of participants and tested configurations is limited and where fully data-driven models may be difficult to train reliably. Its main advantage is not only predictive performance, but also transparency: each stage of the procedure has a clear geometric or statistical interpretation.

The comparative analysis in Section 6 shows that the proposed method complements, rather than replaces, conventional video-game-performance and player-experience measures. In the leave-one-level-out evaluation, the fixed- α Lipschitz model produces a lower error than the 1-NN baseline in nine of the twelve descriptor–metric comparisons, with the clearest improvement for Skewness. Nevertheless, 1-NN remains slightly more accurate for Q10 and for the Q90 MAE, so the results do not support uniform numerical superiority.

The RL-adaptive policy also provides a criterion-dependent result. For the common target configuration, it preserves more upper-tail information and increases Q90 from 0.8583 to 0.9166, while the normalized Log-Normal reconstruction loss increases from 0.004994 to 0.011107. Its contribution is therefore the introduction of feature-specific regulation decisions rather than a general reduction in prediction or reconstruction error.

Related adaptive-game studies generally evaluate score, win rate, difficulty-adjustment accuracy, engagement, perceived difficulty, enjoyment, or flow. By contrast, the present approach predicts several descriptors of the behavioral response-time distribution for an unobserved configuration. The comparison therefore identifies a complementary distributional measure that can be combined with conventional performance and player-experience outcomes in future game-design studies.

There are, however, some limitations. The computation of the empirical Lipschitz constants requires pairwise comparisons between observed configurations, which may become expensive for large datasets. The method also depends on control parameters, such as the Lipschitz tolerance thresholds and the smoothing policy. In addition, the current implementation assumes that the final response-time distribution can be represented by a Log-Normal law. This is reasonable for positive and asymmetric response-time data, but more flexible distributional families or mixture models may be necessary when the empirical distributions are multimodal. Moreover, application to substantially different game designs requires redefining and validating the configuration space, its metric, and the selected distributional model. The sequential regulation mechanism is sensitive to the level-ingestion order. This dependence is consistent with the online-learning rationale of the method, because the active memory and the smoothing decisions are updated as new configurations are incorporated. In the present study, the ingestion order follows the sequence recorded by the experimental system and therefore has a meaningful interpretation. Although exchanging level15 and level16 does not change the predictions, random permutations reveal greater order sensitivity for the mean and Q90 descriptors. Accordingly, the reported one-pass RL results should be interpreted as conditional on the experimental acquisition sequence. Applications without a natural sequential order should employ permutation averaging or an order-independent batch formulation. Each level-specific distribution is estimated from 24 observations. The bootstrap analysis shows that the median and Q10 are comparatively stable, whereas Q90 and Skewness may have wide confidence intervals. Consequently, some of the large upper-tail Lipschitz jumps may partly reflect sampling variability rather than only structural differences between game configurations.

The sequential smoothing mechanism reduces the influence of such abrupt variations, but does not eliminate the uncertainty associated with small level-specific samples. Future studies should therefore consider larger participant samples and uncertainty-aware or bootstrap-weighted versions of the Lipschitz regulation procedure.

The rejection and state-discretization thresholds are interpretable supervisory choices rather than theoretically optimal constants. The sensitivity analysis shows that the main predictions are unchanged when the rejection threshold is reduced from four to three simultaneous violations, whereas requiring five violations mainly affects Q90 and mean. Alternative instability-state boundaries preserve the stable/unstable classification but redistribute events among the mild, medium, and severe states. Future studies with larger datasets should select these thresholds through independent validation.

Overall, the proposed framework should be understood as a controlled and interpretable baseline for distributional response-time prediction in video-game design. It shows how metric regularity, Lipschitz extensions, sequential smoothing and reinforcement-inspired parameter selection can be combined to model response-time distributions from limited experimental data.

Author Contributions: Validation, A.C.-F.; Formal analysis, E.A.S.-P.; Conceptualization, A.C.-F. and E.A.S.-P.; Investigation, A.C.-F. and E.A.S.-P.; Writing—review and editing, A.C.-F.; Writing—original draft, E.A.S.-P.; Supervision, E.A.S.-P.; Visualization, A.C.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Generalitat Valenciana (Spain) through the PROMETEO 2024 CIPROM/2023/32 grant.

Data Availability Statement: The anonymized participant-level response-time data, the complete R analysis script, and the learned feature-specific Q-tables are publicly available in Zenodo at <https://doi.org/10.5281/zenodo.21323313>.

Acknowledgments: The support of the Instituto Universitario de Matemática Pura y Aplicada (IUMPA-UPV), Universitat Politècnica de València, is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ratcliff, R.; McKoon, G. The Diffusion Decision Model: Theory and Data for Two-Choice Decision Tasks. *Neural Comput.* **2008**, *20*, 873–922. [[CrossRef](#)] [[PubMed](#)]
2. Whelan, R. Effective Analysis of Reaction Time Data. *Psychol. Rev.* **2008**, *58*, 475–482. [[CrossRef](#)]
3. Palmer, E.M.; Horowitz, T.S.; Torralba, A.; Wolfe, J.M. What Are the Shapes of Response Time Distributions in Visual Search? *J. Exp. Psychol. Hum. Percept. Perform.* **2011**, *37*, 58–71. [[CrossRef](#)] [[PubMed](#)]
4. Vankov, I.I. The Hazards of Dealing with Response Time Outliers. *Front. Psychol.* **2023**, *14*, 1220281. [[CrossRef](#)] [[PubMed](#)]
5. De Boeck, P.; Jeon, M. An Overview of Models for Response Times and Processes in Cognitive Tests. *Front. Psychol.* **2019**, *10*, 102. [[CrossRef](#)] [[PubMed](#)]
6. Osmon, D.C.; Kazakov, D.; Santos, O.; Kassel, M.T. Non-Gaussian Distributional Analyses of Reaction Times (RT): Improvements that Increase Efficacy of RT Tasks for Describing Cognitive Processes. *Neuropsychol. Rev.* **2018**, *28*, 359–376. [[CrossRef](#)] [[PubMed](#)]
7. Castro-Palacio, J.C.; Fernández-de-Córdoba, P.; Isidro, J.M.; Sahu, S.; Navarro-Pardo, E. Human Reaction Times: Linking Individual and Collective Behaviour through Physics Modeling. *Symmetry* **2021**, *13*, 451. [[CrossRef](#)]
8. Voormann, A.; Miller, J. Sequential Effects on Reaction Time Distributions: Commonalities and Differences Across Paradigms. *J. Cogn.* **2024**, *7*, 68. [[CrossRef](#)] [[PubMed](#)]
9. Ziv, G.; Lidor, R.; Levin, O. Reaction Time and Working Memory in Gamers and Non-Gamers. *Sci. Rep.* **2022**, *12*, 6798. [[CrossRef](#)] [[PubMed](#)]
10. Deleuze, J.; Christiaens, M.; Nuyens, F.; Billieux, J. Shoot at First Sight! First Person Shooter Players Display Reduced Reaction Time and Compromised Inhibitory Control in Comparison to Other Video Game Players. *Comput. Hum. Behav.* **2017**, *72*, 570–576. [[CrossRef](#)]

11. Klačnja, A.; Milenović, N.; Lukač, S.; Knežević, A.; Drid, P.; Popović, B.; Trajković, N. The Effects of Regular Physical Activity and Playing Video Games on Reaction Time in Adolescents. *Int. J. Environ. Res. Public Health* **2022**, *19*, 9278. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Hunnicke, R. The Case for Dynamic Difficulty Adjustment in Games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*; ACM: New York, NY, USA, 2005; pp. 429–433. [\[CrossRef\]](#)
13. Mortazavi, F.; Moradi, H.; Vahabie, A.-H. Dynamic Difficulty Adjustment Approaches in Video Games: A Systematic Literature Review. *Multimed. Tools Appl.* **2024**, *83*, 83227–83274. [\[CrossRef\]](#)
14. Yannakakis, G.N.; Hallam, J. Real-Time Game Adaptation for Optimizing Player Satisfaction. *IEEE Trans. Comput. Intell. Games* **2009**, *1*, 121–133. [\[CrossRef\]](#)
15. Fisher, N.; Kulshreshtha, A.K. Exploring Dynamic Difficulty Adjustment Methods for Video Games. *Virtual Worlds* **2024**, *3*, 230–255. [\[CrossRef\]](#)
16. Guo, Z.; Thawonmas, R.; Ren, X. Rethinking Dynamic Difficulty Adjustment for Video Game Design. *Entertain. Comput.* **2024**, *50*, 100663. [\[CrossRef\]](#)
17. Climent, L.; Longhi, A.; Arbelaez, A.; Mancini, M. A Framework for Designing Reinforcement Learning Agents with Dynamic Difficulty Adjustment in Single-Player Action Video Games. *Entertain. Comput.* **2024**, *50*, 100686. [\[CrossRef\]](#)
18. Fuchs, R.; Gieseke, R.; Dockhorn, A. Personalized Dynamic Difficulty Adjustment: Imitation Learning Meets Reinforcement Learning. In *Proceedings of the 2024 IEEE Conference on Games (CoG)*; IEEE: New York, NY, USA, 2024; pp. 1–2. [\[CrossRef\]](#)
19. Paraschos, P.D.; Koulouriotis, D.E. Fuzzy Logic-Based Dynamic Difficulty Adjustment for Adaptive Game Environments. *Electronics* **2025**, *14*, 146. [\[CrossRef\]](#)
20. Souza, C.H.R.; de Oliveira, S.S.; Berretta, L.D.O.; de Carvalho, S.T. Extending a MAPE-K Loop-Based Framework for Dynamic Difficulty Adjustment in Single-Player Games. *Entertain. Comput.* **2025**, *52*, 100842. [\[CrossRef\]](#)
21. Rezapour, M.M.; Fatemi, A.; Nematbakhsh, M.A. A Methodology for Using Players' Chat Content for Dynamic Difficulty Adjustment in Metaverse Multiplayer Games. *Appl. Soft Comput.* **2024**, *156*, 111497. [\[CrossRef\]](#)
22. Li, X.; Ye, Z.; Xia, Y.; Thawonmas, R. Dynamic Difficulty Adjustment Using a Large Language Model: A Case Study in Magic: The Gathering. *Entertain. Comput.* **2025**, *55*, 100997. [\[CrossRef\]](#)
23. Khalifa, A.; Bontrager, P.; Earle, S.; Togelius, J. PCGRL: Procedural Content Generation via Reinforcement Learning. *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.* **2020**, *16*, 95–101. [\[CrossRef\]](#)
24. Earle, S.; Jiang, Z.; Togelius, J. Scaling, Control and Generalization in Reinforcement Learning Level Generators. In *Proceedings of the 2024 IEEE Conference on Games (CoG)*; IEEE: New York, NY, USA, 2024; pp. 1–8. [\[CrossRef\]](#)
25. Rupp, F.; Eckert, K. G-PCGRL: Procedural Graph Data Generation via Reinforcement Learning. In *Proceedings of the 2024 IEEE Conference on Games (CoG)*; IEEE: New York, NY, USA, 2024; pp. 1–8. [\[CrossRef\]](#)
26. Rupp, F.; Eberhardinger, M.; Eckert, K. Simulation-Driven Balancing of Competitive Game Levels with Reinforcement Learning. *IEEE Trans. Games* **2024**, *16*, 903–913. [\[CrossRef\]](#)
27. Barthet, M.; Branco, D.; Gallotta, R.; Khalifa, A.; Yannakakis, G.N. Closing the Affective Loop via Experience-Driven Reinforcement Learning Designers. In *Proceedings of the 2024 12th International Conference on Affective Computing and Intelligent Interaction (ACII)*; IEEE: New York, NY, USA, 2024; pp. 257–265. [\[CrossRef\]](#)
28. Dutra, P.V.M.; Villela, S.M.; Neto, R.F. A Mixed-Initiative Design Framework for Procedural Content Generation Using Reinforcement Learning. *Entertain. Comput.* **2025**, *52*, 100759. [\[CrossRef\]](#)
29. Lin, Y.; Wei, Y.; Chen, D.; Li, Y.; Erkan, U.; Toktas, A.; Gao, S.; Zhang, Y. Cryptanalysis and Improvement of a Video Cryptosystem via Chaos and S-Box. *ACM Trans. Multimed. Comput. Commun. Appl.* **2026**, *22*, 173. [\[CrossRef\]](#)
30. McShane, E.J. Extension of Range of Functions. *Bull. Am. Math. Soc.* **1934**, *40*, 837–842. [\[CrossRef\]](#)
31. Whitney, H. Analytic Extensions of Differentiable Functions Defined in Closed Sets. *Trans. Am. Math. Soc.* **1934**, *36*, 63–89. [\[CrossRef\]](#)
32. Calabuig, J.M.; Falciani, H.; Sánchez-Pérez, E.A. Dreaming Machine Learning: Lipschitz Extensions for Reinforcement Learning on Financial Markets. *Neurocomputing* **2020**, *398*, 172–184. [\[CrossRef\]](#)
33. Falciani, H.; Sánchez-Pérez, E.A. Semi-Lipschitz Functions and Machine Learning for Discrete Dynamical Systems on Graphs. *Mach. Learn.* **2022**, *111*, 1765–1797. [\[CrossRef\]](#)
34. Watkins, C.J.C.H.; Dayan, P. Q-Learning. *Mach. Learn.* **1992**, *8*, 279–292. [\[CrossRef\]](#)
35. Klein, N. Distributional Regression for Data Analysis. *Annu. Rev. Stat. Its Appl.* **2024**, *11*, 321–346. [\[CrossRef\]](#)
36. Puig Castro, D.; Coronado Ferrer, A.; Castro Palacio, J.C.; Fernández de Córdoba, P.; Ortigosa, N.; Sánchez Pérez, E.A. Non-Centered Chi Distributions as Models for Fair Assessment in Sports Performance. *Symmetry* **2025**, *17*, 1039. [\[CrossRef\]](#)
37. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006. Available online: <http://www.GaussianProcess.org/gpml> (accessed on 17 December 2025).
38. Marrel, A.; Iooss, B. Probabilistic Surrogate Modeling by Gaussian Process: A Review on Recent Insights in Estimation and Validation. *Reliab. Eng. Syst. Saf.* **2024**, *247*, 110094. [\[CrossRef\]](#)

39. Polke, D.; Ahle, E.; Söffker, D. Adaptive Learning with Gaussian Process Regression: A Comprehensive Review of Methods and Applications. *Mach. Learn. Knowl. Extr.* **2026**, *8*, 101. [[CrossRef](#)]
40. Silverman, B.W. Density Estimation for Statistics and Data Analysis. In *Monographs on Statistics and Applied Probability*; Chapman and Hall: London, UK, 1986; Volume 26. [[CrossRef](#)]
41. Zhao, W.; Tabak, E.G. Adaptive Kernel Conditional Density Estimation. *Inf. Inference J. IMA* **2025**, *14*, iaae037. [[CrossRef](#)]
42. Xue, S.; Wu, M.; Kolen, J.; Aghdaie, N.; Zaman, K.A. Dynamic Difficulty Adjustment for Maximized Engagement in Digital Games. In *Proceedings of the 26th International Conference on World Wide Web Companion*; ACM: New York, NY, USA, 2017; pp. 465–471. [[CrossRef](#)]
43. Darzi, A.; McCrea, S.M.; Novak, D. User Experience with Dynamic Difficulty Adjustment Methods for an Affective Exergame: Comparative Laboratory-Based Study. *JMIR Serious Games* **2021**, *9*, e25771. [[CrossRef](#)] [[PubMed](#)]
44. Zook, A.; Riedl, M. A Temporal Data-Driven Player Model for Dynamic Difficulty Adjustment. *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.* **2012**, *8*, 93–98. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.