**Toward Validating Artificial Organisms Animated by the**

**Mathematical Principles of Reinforcement**

David J. Cox1, 2

1 Endicott College; 2 Mosaic Pediatric Therapy

**Conflict of Interest:** None.

**Funding:** None.

**Data Availability**: All code and data associated with this manuscript is available in an open repository and freely accessible.

**ORCID:** 0000-0003-4376-2104

**Abstract:** [Fill in at end.]

***Keywords:***

**Toward Validating Artificial Organisms Animated by the**

**Mathematical Principles of Reinforcement**

Computational simulations are dynamic, process-oriented implementations of models using mathematical equations or algorithms to represent and predict the behavior of real-world systems (Gilbert & Troitzsch, 2005; Winsberg, 2010). Computational simulations have become increasingly valuable across many scientific disciplines, including physics (e.g., Feynman, 1982; Metropolis & Ulam, 1949), chemistry (e.g., Karplus & McCammon, 2002; Warshel & Levitt, 1976), biology (e.g., Noble, 2002; Reeke & Edelman, 1988), ecology (e.g., Grimm & Railsback, 2005), psychology (e.g., Epstein, 2006; Smaldino, 2017), neuroscience (e.g., Dayan & Abbott, 2001; Izhikevich, 2004), economics (e.g., Tesfatsion, 2002; LeBaron, 2006), and artificial intelligence (e.g., Russell & Norvig, 2021; Silver et al., 2016). These simulations allow researchers to test theories, examine parameter interactions, and generate predictions that can guide empirical studies (Sun, 2008).

In behavior analysis, computational modeling allows researchers to explore and potentially explain behavior under various environmental contingencies without the immediate need for nonhuman or human participants (McDowell, 2004). Though scant, researchers in behavior analysis have used computational modeling to mimic or understand behavioral phenomena (Donahoe & Palmer, 1994; McDowell, 2013a). For example, Donahoe and Palmer (1994) developed connectionist models to simulate operant processes such as reinforcement, stimulus control, and the interaction of motivational and discriminative variables. Others have used computational models for *x* (*citations*) and *y* (*citations*).

One major development in this area has been the Evolutionary Theory of Behavior Dynamics (ETBD; McDowell, 2004). ETBD is a rules-based genetici algorithmic system that models behavior based on adaptation to environmental feedback. When used to animate artificial organisms (AOs), ETBD produces data at equilibria similar to biological organisms on a range of phenomena such as conforming to: the generalized hyperbola on single interval schedules (McDowell, 1986, 2019); power function matching on concurrent interval schedules (Baum, 1979; McDowell, 2019); and absolute response rates on concurrent interval schedules relative to a modified generalized hyperbola for two responses (McDowell, 2013, 2019; McDowell & Calvin, 2015; McDowell & Popa, 2010).

Beyond its utility in modeling laboratory behavior, ETBD has also been extended to model clinically relevant behavior (Falligant et al., 2022; Morris & Hagopian, 2020). For example, Falligant et al. (2022) demonstrated how variations in reinforcement density and schedule thinning could influence the likelihood of resurgence following treatment, offering insight into the dynamics of relapse in applied settings. Similarly, Morris and Hagopian (2020) used an ETBD-based modeling framework to examine the effects of functional communication training and reinforcement schedule adjustments, simulating behavioral patterns often observed during clinical interventions. These applications of ETBD show promise for improving clinical decision-making by enabling behavior analysts to anticipate how specific environmental manipulations may produce complex, dynamic patterns of behavior over time.

Despite the success of ETBD to date, it also has several limitations for researchers interested in directly modeling behavioral processes. For example, ETBD successfully captures the behavioral patterns described above through the emergent properties of a rule-based, agent-oriented system rather than by modeling the underlying behavioral processes in a way that maps directly onto established behavioral principles (McDowell, 2013a). Specifically, ETBD is a genetic algorithm that updates response repertoires and response probabilities using the evolutionary concepts of selection, reproduction, and mutation. Thus, though ETBD’s simulations capture a wide range of behavioral patterns, they do not correspond to clearly defined or independently measurable behavioral processes, making its output functionally equivalent to the behavior of biological organisms—but not materially equivalent (McDowell, 2019). This can make it difficult to empirically calibrate the model based on observable features of an organism or environment.

An alternative approach to simulating behavior via artificial organisms involves using theory-driven mechanistic models that strive for functional and material equivalence (Donahoe & Palmer, 1994; Killeen, 1994). Theory-driven mechanistic models attempt to simulate behavior by computationally instantiating theoretical constructs as states that correspond to measurable behavioral processes. This alignment between model components and operant principles improves interpretability and allows for more precise hypothesis testing, as each parameter has a clear experimental analog (Killeen, 2003; Shull, 2004). Mechanistic models also facilitate empirical calibration and generalization, making them potentially more useful in basic and applied research settings.

One example of a theory-driven mechanistic model is the Mathematical Principles of Reinforcement (MPR) (Killeen, 1994). MPR provides closed-form equations that formalize the relationships among core behavioral processes. In MPR, response rate is determined by the joint influence of several components. These include arousal, which reflects the activating effects of reinforcement; coupling, which captures the extent to which responses are associated with reinforcement; and motor capacity, which reflects biomechanical limitations on responding. Reinforcement is conceptualized as energizing behavior through arousal, while coupling governs the degree to which behavior is directed toward the response that produces reinforcement. The model produces characteristic curves describing response output under various fixed and variable schedule conditions and has been used to explain phenomena such as scalloping on fixed interval schedules, break-and-run patterns on ratio schedules, and rate differences across reinforcer types.

The purpose of the present study was to take an initial attempt to translate the foundations of MPR into a set of equations that could be implemented recursively in code to animate the behavior of AOs. Once instantiated, AO responding could be tested across a range of single VRs, single VIs, concurrent VRs, and concurrent VIs, and spanning parametric manipulations of the parameters driving MPR (i.e., , , ). In turn, similar to ETBD, we could test whether this approach to animating AOs would lead to the quantitative emergent equilibria phenomena observed in biological organisms, such as conformation to the generalized hyperbola (McDowell, 1986), the generalized matching law (Baum, 1979), and absolute response rates in each component of concurrent VI schedules (McDowell, 2013b; McDowell & Calvin, 2015; McDowell & Popa, 2010).

**METHOD**

Four steps were taken to initially translate MPR into recursive code to simulate how an AO updates its behavior in response to reinforcement over time. First, the arousal equation (equation 1; Killeen, 2023), originally expressed as a function of reinforcement rate, was adapted into a recursive smoothing function that continuously updates arousal based on new reinforcement events and a learning rate parameter (*α*):

*A*ₜ = *α*·(activation·*r*ₜ) + (1 - *α*)·*A*ₜ₋₁. (Equation 1)

Second, the total response rate was modeled as a hyperbolic function of arousal constrained by response duration (*δ*) (2023; equation 2b):

*b*Total = (1/*δ*)·(*A* / (1 + *A*)), *δ* > 0. (Equation 2)

Third, target response rate can be calculated using equation 7 from Killeen (2023):

*b*Target = (*C*/*δ*)·(*A* / (1 + *A*)), (Equation 3)

where C is the coupling coefficient, which varies between 0 and 1. Finally, coupling can be updated using a recursive learning rule that allows it to increase toward 1 when reinforced and decay toward a low proximity value (e.g., 0.1) when not. This can be implemented as:

C*t*= *C*t−1 + *η* ⋅ (*proximity*−*C*t−1), (Equation 4).

where η is the coupling learning rate and proximity is 1 for reinforcement and 0.1 otherwise.

**Simulating Responding to Single VR Schedules**

The first simulation involved testing responding to the single VR schedules of VR1, VR3, VR10, VR30, VR100, VR300, and VR1000. A full grid of simulation conditions was generated by crossing five values of the activation parameter (0.1, 0.2, 0.4, 0.8, 1.6), four values of the response duration parameter *δ* (0.25, 0.50, 1.0, 2.0), five values of the arousal learning rate *α* (0.01, 0.03, 0.1, 0.3, 1.0), and four values of the coupling learning rate *η* (0.01, 0.03, 0.3, 1.0. This produced a total of 2,800 unique parameter combinations (including VR schedules). For each combination, a simulation was conducted for 1,000 time steps.

At each step, reinforcement was delivered probabilistically with a fixed probability equal to the inverse of the VR schedule (𝑝=1/*VR*). The model maintained evolving values for arousal (*A*) and coupling (*C*), which were initialized to 0.6 and 0.5, respectively, and updated at each time step. For every step, the model computed the total response rate and the target response rate based on the current values of *A* and *C* aligned with Equations 2 and 3 above.

**Simulating Responding to Single VI Schedules**

A second set of simulations evaluated responding under single variable interval (VI) schedules. The schedules tested included VI1, VI3, VI10, VI30, VI100, VI300, and VI1000, defined in seconds. Each VI schedule was crossed with the same full parameter grid used in the VR simulations, again resulting in 2,800 unique simulations. In each simulation, reinforcement availability times were drawn from an exponential distribution with a mean equal to the VI schedule value. The simulated organism was programmed to respond at an average rate of 2.5 responses per second, producing interresponse times sampled accordingly. Reinforcement was delivered immediately when the current time exceeded the next available reinforcement time, after which the next VI time was scheduled. Within each simulation, arousal (*A*) and coupling (*C*) values were initialized to 0.6 and 0.5, respectively, and were updated on each time step (Equations 1-4). At each time step, the model computed the total response rate and the target response rate based on the current state of *A* and *C*. This process continued for 1,000 simulated steps.

**Simulating Responding to Concurrent VR Schedules**

The third simulation modeled behavior on concurrent variable ratio (VR) schedules. Five concurrent VR pairs were tested: VR 1 vs. VR 9, VR 1 vs. VR 3, VR 2 vs. VR 2, VR 3 vs. VR 1, and VR 9 vs. VR 1. These pairs span symmetrical and asymmetrical reinforcement contingencies. For each pair, we conducted the same grid search across a full set of model parameters: five activation values (0.1, 0.2, 0.4, 0.8, 1.6), four values of the response duration parameter *δ* (0.25, 0.50, 1.0, 2.0), five values of the arousal learning rate *α* (0.01, 0.03, 0.1, 0.3, 1.0), and four values of the coupling learning rate *η* (0.01, 0.03, 0.3, 1.0). This led to 2,000 unique simulation conditions (5 VR pairs × 5 activations × 4 *δ* × 5 *α* × 4 *η*). Each condition was simulated for 1,000 time steps.

At each step, the artificial organism (AO) selected between two response alternatives according to the relative strength of coupling to each alternative. Specifically, the probability of selecting an option was proportional to the coupling coefficient for that option (e.g., 𝑝1=𝐶1/(𝐶1+𝐶2)). A response was emitted to one alternative, and reinforcement was delivered probabilistically based on the corresponding VR schedule (i.e., 𝑝=1/VR). After each action, arousal and coupling were updated for the chosen option, and the unchosen option experienced decay in arousal and coupling based on its reinforcement omission (equations 1-4). At each step, the model computed arousal, coupling, and binary indicators of choice for both alternatives.

**Simulating Responding to Concurrent VI Schedules**

The fourth simulation modeled responding on concurrent variable interval (VI) schedules. Five concurrent VI pairs were tested: VI 1 vs. VI 9, VI 1 vs. VI 3, VI 2 vs. VI 2, VI 3 vs. VI 1, and VI 9 vs. VI 1. These pairs were chosen to span symmetrical and asymmetrical reinforcement contingencies. For each schedule pair, we ran the same a grid search across activation values, response duration parameter *δ*, arousal learning rate *α*, and coupling learning rate *η*. This produced 2,000 unique simulation conditions (5 VI pairs × 5 activations × 4 *δ* × 5 *α* × 4 *η*), with each condition simulated for 1,000 time steps.

In each simulation, reinforcers for each schedule were scheduled using independent exponential inter-reinforcement intervals drawn from the specified VI parameters. At every time step, the AO emitted a response to one of the two alternatives, selected probabilistically according to the relative coupling strength to each option (i.e., 𝑝₁=𝐶₁/(𝐶₁ + 𝐶₂)). If a reinforcer was available on the selected alternative at that time step, it was delivered; otherwise, no reinforcement occurred. Upon responding, arousal and coupling were updated for the chosen option based on whether reinforcement was obtained, and the unchosen option underwent passive decay.

**All Simulations**

After 1,000 steps, each simulation was programmed to compute and return the following summary statistics: arithmetic mean (average), minimum, maximum, and 95% confidence interval (based on the standard error and a two-tailed *t*-distribution) for each variable (and for each alternative in concurrent arrangements). The results from the simulations were compiled into a single DataFrame for downstream analysis.

**Data Analytics**

All simulations, analyses, and visualizations were conducted using Python (version 3.10) and a range of open-source packages. Numerical operations and data manipulation were performed with NumPy (Harris et al., 2020) and pandas (McKinney, 2010). Scientific computations and statistical tests relied on SciPy (Virtanen et al., 2020), scikit-learn (Pedregosa et al., 2011), and Statsmodels (Seabold & Perktold, 2010). Clustering, classification, and linear modeling routines were implemented using scikit-learn’s Random Forest, KMeans, and linear regression modules. Visualization was conducted using Matplotlib (Hunter, 2007) and Seaborn (Waskom, 2021). Model explainability was supported using SHAP (Lundberg & Lee, 2017). All code was run in a Python environment with default configuration, and reproducibility was maintained via fixed random seeds where applicable. A copy of the code can be found at [redacted for blind review; an .html converted output was added as a supplemental file].

**RESULTS**

Figure 1 shows a basic demonstration of the output obtained a single, 1000-step simulation run with *δ*=0.25, activation=0.6, arousal=0.6, *α*=0.1, *η*=0.05, and VR-10 as the schedule of reinforcement for a single target response. Based on contacted reinforcement for responding (vertical red lines), each panel shows how the value updated across steps with the top panel showing updating arousal (Eq.1), the second to top showing updating coupling (Eq. 4), the third panel showing updating overall rate of responding (Eq. 2), and the bottom panel showing updating target responding. Figure 1 is presented to show give a sense for how the simulations proceeded, and to highlight how relatively stable levels of values were obtained within each simulation after approximately 200 steps.

Figure 2 shows two analyses of AO target response rates for all simulations involving a single VR schedule. The left panel shows the average response rate (*y-*axis) as a function of the specific VR schedule in effect (*x*-axis). The right panel shows how the AOs animated by equations 1-4 exhibit behavior (gray markers) on single VR schedules of reinforcement where the overall average rates of responding (open white markers) that conform at equilibrium to the generalized hyperbola (McDowell, 1986):

. (Equation 5)

where *B* represents response rate, *r* represents reinforcement rate, *a* represents sensitivity to changing schedules of reinforcement, *b* represents deviations from strict matching not captured by reinforcement schedules (*aka* bias), and *c* represents reinforcement amounts for non-target responding (McDowell, 2019). Together, the two panels in Figure 2 suggest that AOs animated by the recursive MPR equations 1-4 emit behavior consistent with past research consistent with biological organisms behaving relative to single VR schedules. Rates of responding systematically decreased as the VR schedule increased (left panel) and the generalized hyperbola described well average rates of responding (*r*21.00). In turn, this allows us to examine how changing values of activation, *δ*, *α*, and *η* impacted arousal, coupling, target response rate, and total response rate at equilibrium.

Figure 3 shows the average values for arousal (*A* Eq. 1), coupling (*C* Eq. 2), target response rate, and total response rate change at equilibrium, as a function of the tested activation, *δ*, *α*, and *η* parameters. [*Add description and stats*].

Figure 4 shows two analyses of AO target response rates for all simulations involving a single VI schedule. The left panel shows the average response rate (*y-*axis) as a function of the specific VI schedule in effect (*x*-axis). The right panel shows how the AOs animated by equations 1-4 exhibit behavior (gray markers) on single VI schedules of reinforcement where the overall average rates of responding (open white markers) that conform at equilibrium to the generalized hyperbola (Equation 5). Together, the two panels in Figure 4 suggest that AOs animated by the recursive MPR equations 1-4 emit behavior consistent with past research consistent with biological organisms behaving relative to single VI schedules. Rates of responding systematically decreased as the VI schedule increased (left panel) and the generalized hyperbola described well average rates of responding (*r*20.99). In turn, this allows us to examine how changing values of activation, *δ*, *α*, and *η* impacted arousal, coupling, target response rate, and total response rate at equilibrium.

Figure 5 shows the average values for arousal (*A* Eq. 1), coupling (*C* Eq. 2), target response rate, and total response rate change at equilibrium, as a function of the tested activation, *δ*, *α*, and *η* parameters. [*Add description and stats*].

Figure 6 shows AO target response rates for all simulations involving a concurrent VR schedules. The *y-*axis shows logged response ratios of *B*1/*B*2 as a function of the logged reinforcement ratios of *R*1/*R*2. The dashed diagonal line shows what would be expected with perfect matching, the gray markers are individual simulation runs, the white markers represent the average rate of responding across all simulations, and the black line shows the fit of the power-function matching equation (Baum, 1979):

, (Equation 6)

where *b* and *a* represent the same parameters as in Equation 5. Overall, Equation 6 described the data well (*r*2=0.95) and AOs exhibited the classic S-shaped pattern of responding commonly observed with biological organisms responding to concurrent VR schedules (e.g., *citations*). We also fit equations 3 and 4 from McDowell (2019) to test whether the AOs animated by the recursive MPR equations would exhibit absolute response rates in each component of concurrent VR schedules conforming to:

, (Equation 7)

and

. (Equation 8)

Fits of both equations were very good (*r*2=0.98 for both) with the parameters estimated to be: *a*=1.08, *b*1=1.37, *b*2=1.37, *c*1=0.44, and *c*2=0.44. Given the good overall fits of Equations 6-8 to the obtained data, we could examine how changing values of activation, *δ*, *α*, and *η* impacted arousal, coupling, and choice rate for *B*1 at equilibrium.

Figure 7 shows the average values for arousal, coupling, and choice rate for *B*1 change at equilibrium as a function of the tested activation, *δ*, *α*, and *η* parameters for responding to concurrent VR schedules. [*Add description and stats*].

Figure 8 shows AO target response rates for all simulations involving a concurrent VI schedules. The *y-*axis shows logged response ratios of *B*1/*B*2 as a function of the logged reinforcement ratios of *R*1/*R*2. The dashed diagonal line shows what would be expected with perfect matching, the gray markers are individual simulation runs, the white markers represent the average rate of responding across all simulations, and the black line shows the fit of the power-function matching equation (Equation 6). Equation 6 described the data well (*r*2=0.96) and AOs exhibiting undermatching commonly observed with biological organisms responding to concurrent VR schedules (e.g., *citations*). Fits of Equations 7 and 8 were good (*r*2=0.97 for both) with the parameters estimated to be: *a*=0.44, *b*1=1.32, *b*2=1.32, *c*1=0.48, and *c*2=0.48. Given the good overall fits of Equations 6-8 to the obtained data, we could examine how changing values of activation, *δ*, *α*, and *η* impacted arousal, coupling, and choice rate for *B*1 at equilibrium.

Figure 9 shows the average values for arousal, coupling, and choice rate for *B*1 change at equilibrium as a function of the tested activation, *δ*, *α*, and *η* parameters for responding to concurrent VR schedules. [*Add description and stats*].

**DISCUSSION**

In our simulations, we were able to generate behavioral patterns that closely resemble those seen under single ratio and interval schedules. For example, under VR schedules, response rates scaled predictably with the ratio requirement, and under VI schedules, response rates stabilized around values consistent with the programmed reinforcement rate. These simulations validate that the recursive implementation of MPR is capable of producing empirically meaningful behavior.

This is significant because it allows researchers to explore the impact of parameters like activation, arousal, and coupling in a principled manner. It also enables rapid exploration of hypothetical manipulations, which can then be tested empirically. Compared to black-box models, our approach preserves interpretability while supporting dynamic analysis.

We also demonstrated that the model could reproduce patterns of choice behavior under concurrent VR and VI schedules. In particular, the simulated agents showed sensitivity to reinforcement ratios and developed response allocations consistent with the matching law. By modeling coupling independently for each option, the system could dynamically adjust preference in response to relative reinforcement outcomes.

These findings underscore the potential of recursive MPR models to account for both local and molar behavioral patterns. Whereas some models require hand-crafted heuristics or population-level summaries, our approach relies solely on moment-to-moment updates to latent behavioral states.

A major advantage of this approach is that it integrates mechanistic precision with simulation flexibility. Parameters such as arousal and coupling are not just tunable—they reflect measurable behavioral processes. This makes the model useful both for theorists and for experimentalists looking to fit data.

Compared to ETBD, our approach provides greater control and interpretability at the level of individual time steps. It also facilitates integration with other modeling frameworks, such as reinforcement learning, which are increasingly used in computational neuroscience and behavioral economics.

Nonetheless, there are limitations. The model assumes static parameter values across time, which may not hold in real-world settings. Extensions to allow adaptive or context-sensitive parameters could improve realism. Additionally, the current model assumes fixed decision processes, and does not yet incorporate aspects like response competition or motor cost.

Another limitation is that the model requires assumptions about the initial values of arousal and coupling. Although we found the results to be robust to reasonable variation, future work could examine sensitivity to these assumptions.

To further explore the potential of this modeling approach, we extended the MPR simulation to include a Q-learning mechanism for decision-making. In this hybrid model, arousal and coupling are updated via MPR, while choice behavior is governed by reinforcement history via Q-values. This enables simulation of adaptive, learning-based decision-making in dynamic environments.

Initial simulations with this hybrid agent show that it can learn to prefer options that yield more reinforcement, even when schedules are probabilistic. This aligns with observed behavior in animals and opens the door to studying the interaction of local state variables and global learning strategies.

Looking forward, we see several exciting directions for computational simulations in behavior analysis. Recursive MPR models could be extended to capture more complex contingencies, such as chained schedules, delayed reinforcement, or punishment. They could also be integrated with empirical data pipelines to support model-based behavioral analysis.

With the rising interest in AI, reinforcement learning, and mechanistic psychology, behavior analysis has an opportunity to reassert itself at the forefront of modeling behavior. Tools like the recursive MPR implementation presented here can serve as bridges between experimental rigor and computational innovation.

In conclusion, we have shown that the mathematical principles of reinforcement can be implemented recursively to produce meaningful simulations of behavior. This approach balances mechanistic fidelity with computational flexibility and offers a viable alternative to both heuristic models and more abstract simulation methods like ETBD. We believe this work provides a foundation for further integration of behavior analytic theory with modern simulation practices.

**REFERENCES**

Baum, W. M. (1979). Matching, undermatching, and overmatching in studies of choice. *Journal of the Experimental Analysis of Behavior, 32*, 269–281. <https://doi.org/10.1901/jeab.1979.32-269>

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. MIT Press.

Donahoe, J. W., & Palmer, D. C. (1994). *Learning and complex behavior*. Allyn & Bacon.

Epstein, J. M. (2006). *Generative social science: Studies in agent-based computational modeling.* Princeton University Press.

Falligant, J. M., Morris, S. L., Pence, S. T., & Hagopian, L. P. (2022). Evaluating treatment relapse of self-injurious behavior using a computational model based on the evolutionary theory of behavior dynamics. *Journal of the Experimental Analysis of Behavior, 117*(3), 694–711. <https://doi.org/10.1002/jeab.738>

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics, 21*(6), 467–488. <https://doi.org/10.1007/BF02650179>

Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the social scientist (2nd ed.)*. Open University Press.

Grimm, V., & Railsback, S. F. (2005). *Individual-based modeling and ecology*. Princeton University Press.

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. *Nature, 585*(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks, 15*(5), 1063–1070. <https://doi.org/10.1109/TNN.2004.832719>

Karplus, M., & McCammon, J. A. (2002). Molecular dynamics simulations of biomolecules. *Nature Structural Biology, 9*(9), 646–652. <https://doi.org/10.1038/nsb0902-646>

Killeen, P. R. (1994). Mathematical principles of reinforcement. *Behavioral and Brain Sciences, 17*(1), 105–172. <https://doi.org/10.1017/S0140525X00034016>

Killeen, P. R. (2003). Complex dynamic behavior. *Journal of the Experimental Analysis of Behavior, 80*(3), 345–366. <https://doi.org/10.1901/jeab.2003.80-345>

Killeen, P. R. (2023). Theory of reinforcement schedules. *Journal of the Experimental Analysis of Behavior, 120*(3), 289-319. <https://doi.org/10.1002/jeab.880>

Killeen, P. R., & Fetterman, J. G. (1988). A behavioral theory of timing. *Psychological Review, 95*(2), 274–295. <https://doi.org/10.1037/0033-295X.95.2.274>

LeBaron, B. (2006). Agent-based computational finance. In *Handbook of computational economics* *(Vol. 2*) (pp. 1187–1233). Elsevier.   
<https://doi.org/10.1016/S1574-0021(05)02024-1>

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 4765–4774).

McDowell, J. J (1986). On the falsifiability of matching theory. *Journal of the Experimental Analysis of Behavior, 45*, 63-74. <https://doi.org/10.1901/jeab.1986.45-63>

McDowell, J. J. (2004). A computational model of selection by consequences. *Journal of the Experimental Analysis of Behavior, 81*(3), 297–317. <https://doi.org/10.1901/jeab.2004.81-297>

McDowell, J. J. (2013a). On the theoretical and empirical status of the evolutionary theory of behavior dynamics. *Psychological Bulletin, 139*(5), 902–919. <https://doi.org/10.1037/a0030992>

McDowell, J. J (2013b). A quantitative evolutionary theory of adaptive behavior dynamics. Psychological Review, 120, 731-750. <https://doi.org/10.1037/a0034244>

McDowell, J. J. (2019). On the current status of the evolutionary theory of behavior dynamics. *Journal of the Experimental Anlaysis of Behavior, 111*(1), 130-145. <https://doi.org/10.1002/jeab.495>

McDowell, J. J., Bass, A. D., & Kessel, R. (2016). Evolutionary theory of behavior dynamics: A model-based approach to understanding behavior*. Behavior Analysis: Research and Practice, 16*(3), 222–234. <https://doi.org/10.1037/bar0000045>

McDowell, J. J, & Calvin, N. T. (2015). Against matching theory: Predictions of an evolutionary theory of behavior dynamics. *Behavioural Processes, 114*, 14–25. <https://doi.org/10.1016/j.beproc.2015.02.007>

McDowell, J. J., & Caron, M. L. (2010). Matching theory and the allocation of behavior in varying environments. *Journal of the Experimental Analysis of Behavior, 93*(3), 411–427. <https://doi.org/10.1901/jeab.2010.93-411>

McDowell, J. J, & Popa, A. (2010). Toward a mechanics of adaptive behavior: Evolutionary dynamics and matching theory statics. *Journal of the Experimental Analysis of Behavior, 94*, 241-260. <https://doi.org/10.1901/jeab.2010.94-241>

McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51–56).

Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association, 44*(247), 335–341. <https://doi.org/10.1080/01621459.1949.10483310>

Morris, S. L., & Hagopian, L. P. (2020). A computational theory of relapse of operant behavior and its application to treatment of severe problem behavior. *Perspectives on Behavior Science, 43*(4), 859–889. <https://doi.org/10.1007/s40614-020-00252-0>

Nevin, J. A., & Grace, R. C. (2000). Behavioral momentum and the law of effect. *Behavioral and Brain Sciences, 23*(1), 73–130. <https://doi.org/10.1017/S0140525X00002405>

Noble, D. (2002). Modeling the heart—from genes to cells to the whole organ. *Science, 295*(5560), 1678–1682. <https://doi.org/10.1126/science.1069881>

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Reeke, G. N., & Edelman, G. M. (1988). Real brains and artificial intelligence. *Daedalus, 117*(1), 143–173.

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A Modern Approach (4th ed.)*. Pearson.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference*.

Shull, R. L. (2004). BPR and other models of response rate and their relation to the matching law. *Behavioral Processes, 66*(2), 153–162. <https://doi.org/10.1016/j.beproc.2004.01.007>

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., …, & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*, 484–489. <https://doi.org/10.1038/nature16961>

Smaldino, P. E. (2017). Models are stupid, and we need more of them. In R. R. Vallacher, S. J. Read, & A. Nowak (Eds.), *Computational social psychology* (pp. 311–331). Routledge. <https://doi.org/10.4324/9781315173726>

Sun, R. (Ed.). (2008). *The Cambridge handbook of computational psychology*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511816772>

Tesfatsion, L. (2002). Agent-based computational economics: Growing economies from the bottom up. *Artificial Life, 8*(1), 55–82. <https://doi.org/10.1162/106454602753694765>

Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods, 17*(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

Warshel, A., & Levitt, M. (1976). Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of Molecular Biology, 103*(2), 227–249. <https://doi.org/10.1016/0022-2836(76)90311-9>

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software, 6*(60), 3021. <https://doi.org/10.21105/joss.03021>

Winsberg, E. (2010). *Science in the age of computer simulation*. University of Chicago Press.

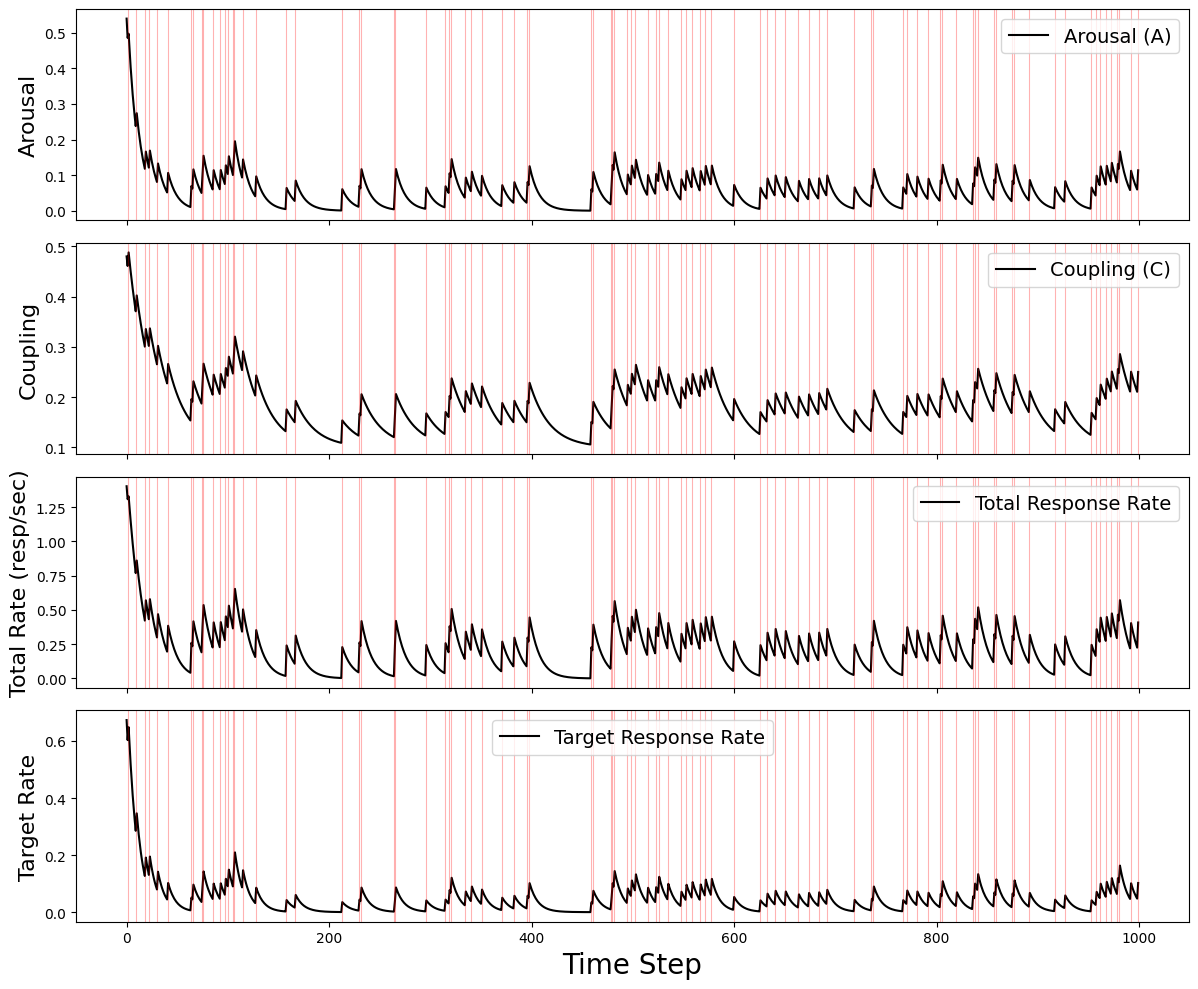
**Table 1.** Equations from MPR (Kileen, 2023) and their recursive counterparts allowing for simulation in the present study.

|  |  |
| --- | --- |
| **MPR Equation** | **Simulation Recursive Equation** |
|  |  |
|  | ,  where: |

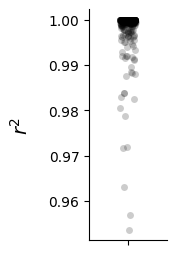
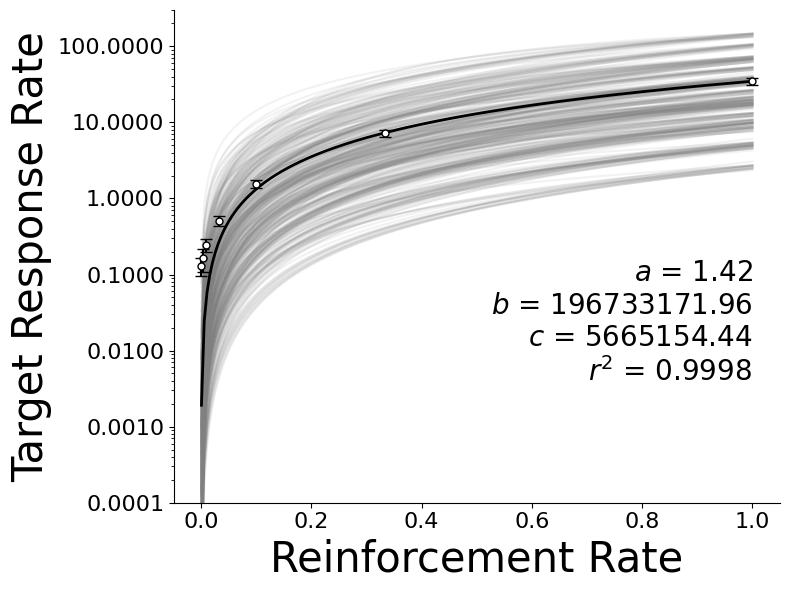
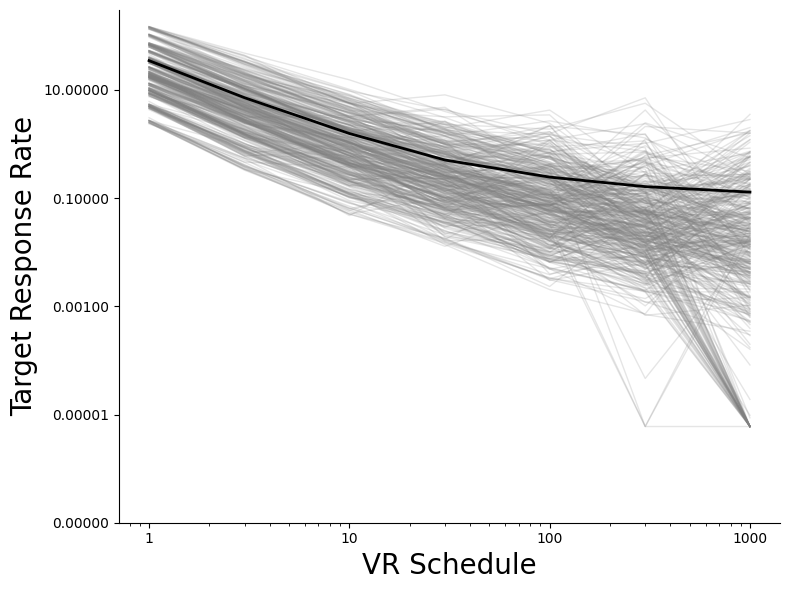
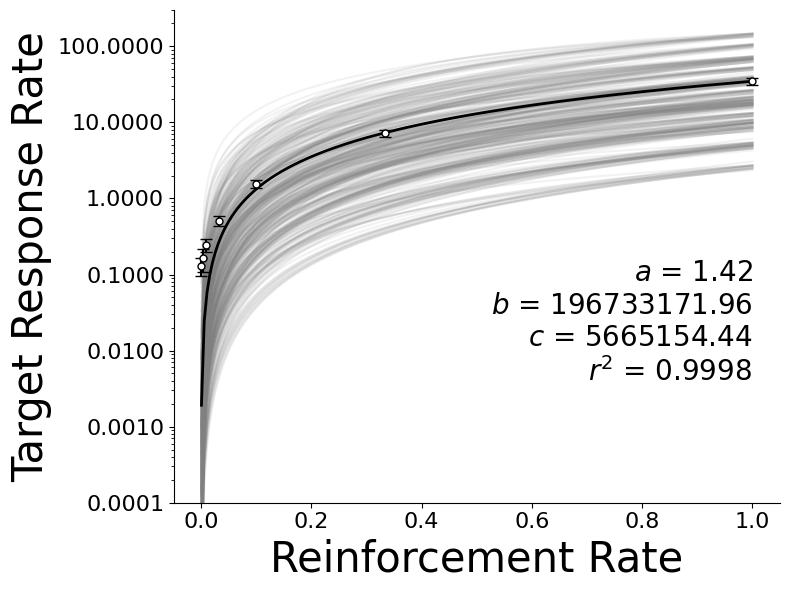
**Table 2.** Pseudocode for an MPR-driven AO responding on single VR schedule.

|  |
| --- |
| **Inputs**:  A\_prev: Previous arousal level  C\_prev: Previous coupling level  r\_t: Reinforcement at time t (1 if reinforced, 0 if not)  alpha: Arousal learning rate (0 < alpha < 1)  activation: Specific activation  delta: Duration of a response  eta: Coupling learning rate (0 < eta < 1)  **Procedure after each time step**:  # Update arousal:  A\_t = alpha \* (activation\*r\_t) + (1-alpha)\*A\_prev  # Determine proximity:  if r\_t == 1, then proximity = 1.0  else, proximity = 0.1  # Update coupling:  C\_t = C\_prev + eta \* (proximity - C\_prev)  # Compute total response rate:  b\_total = (1 / delta) \* (A\_t / (1 + A\_t))  # Compute target response rate:  b\_target = C\_t \* b\_total  **Outputs**:  A\_t: Updated arousal  C\_t: Updated coupling  b\_total: Total response rate (all behavior)  b\_target: Target response rate (e.g., lever press) |

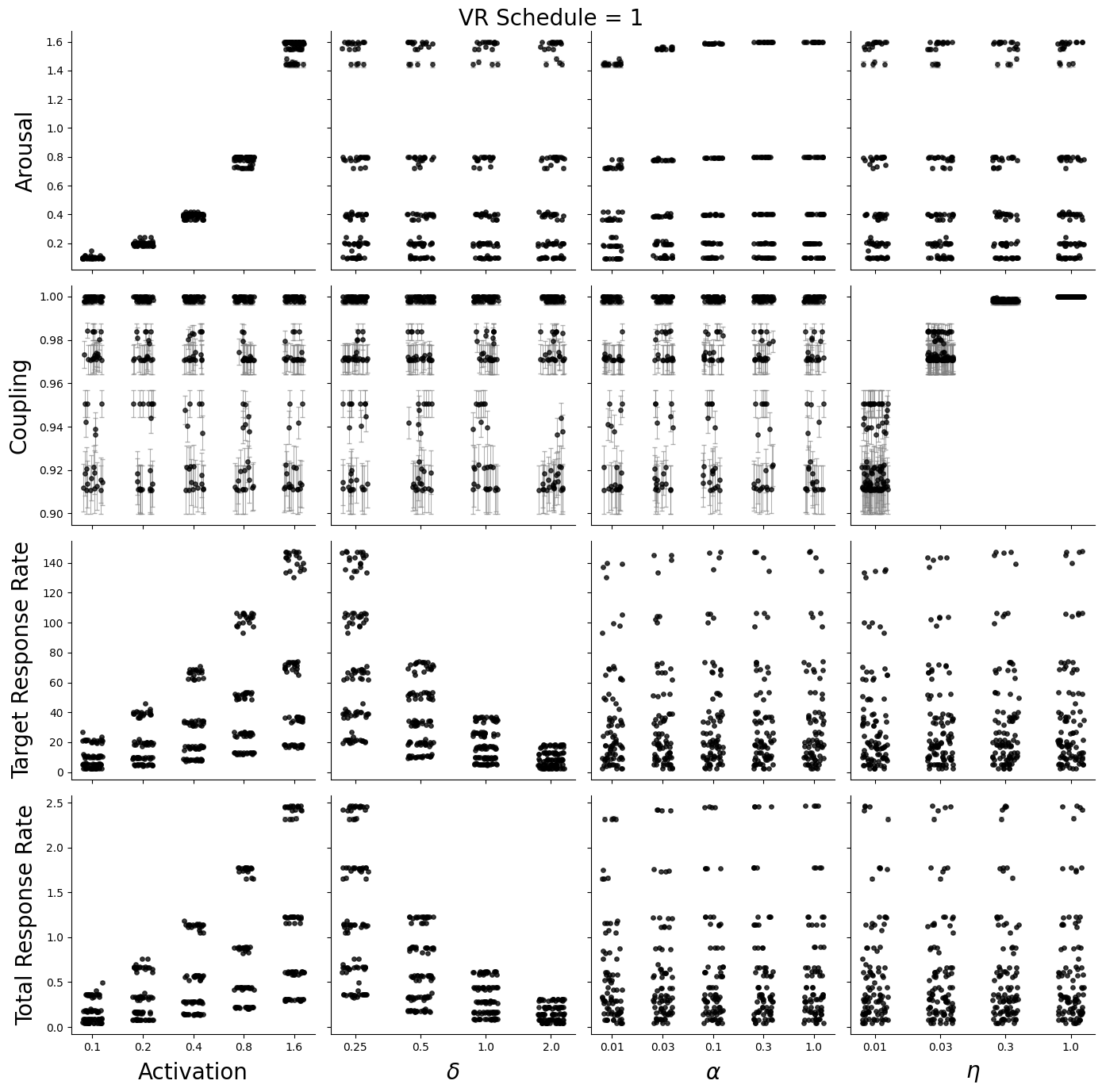
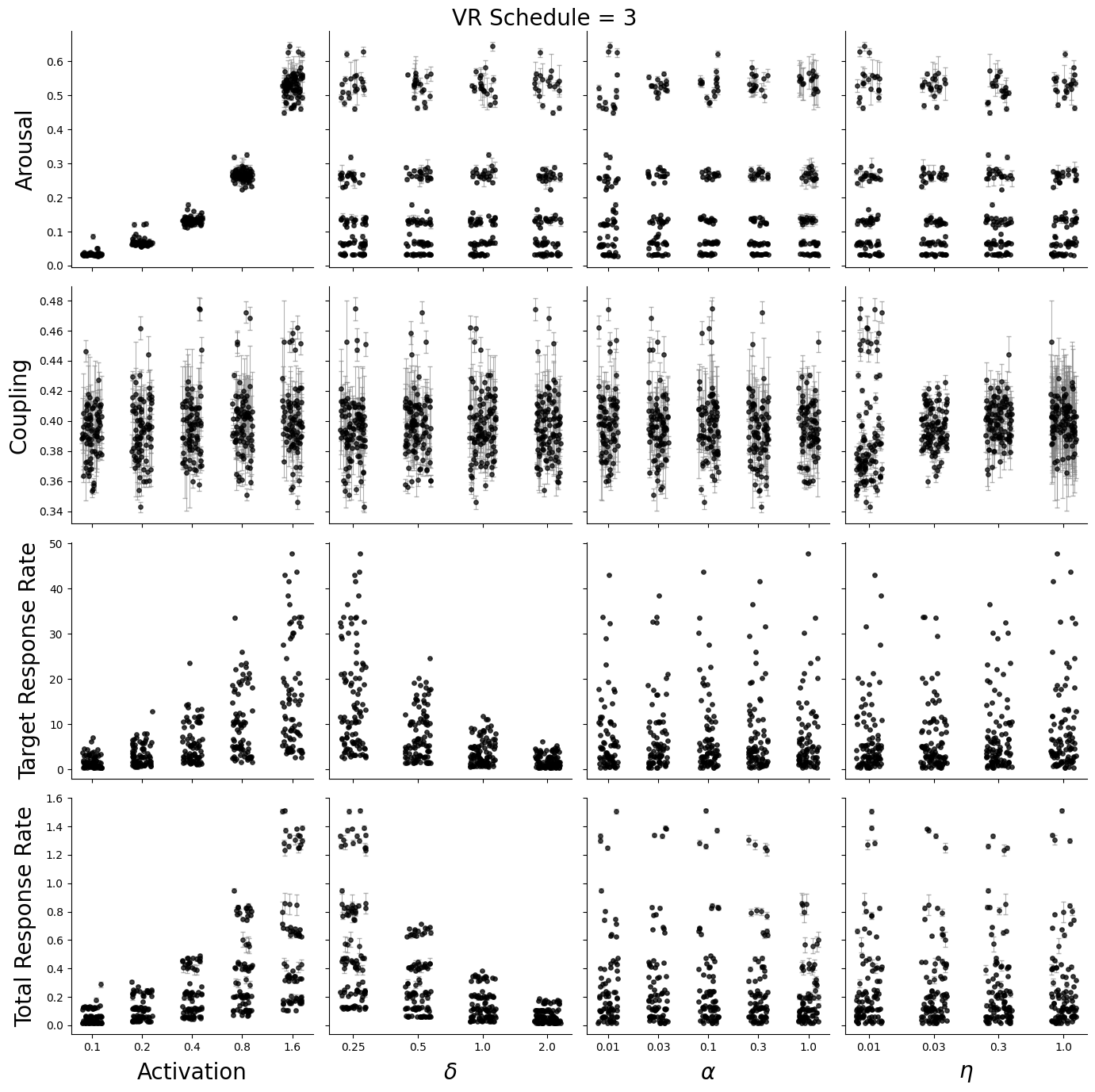
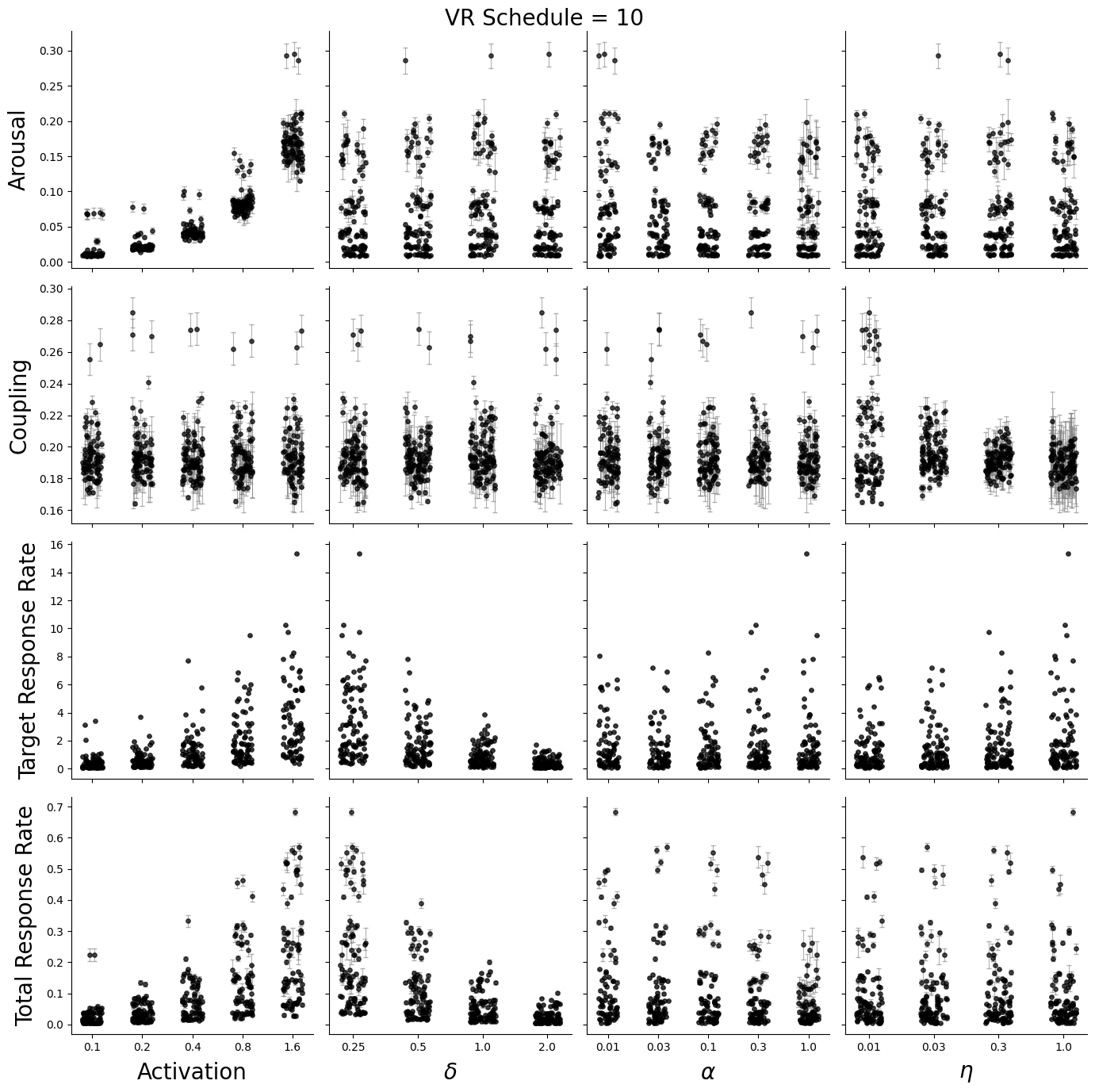
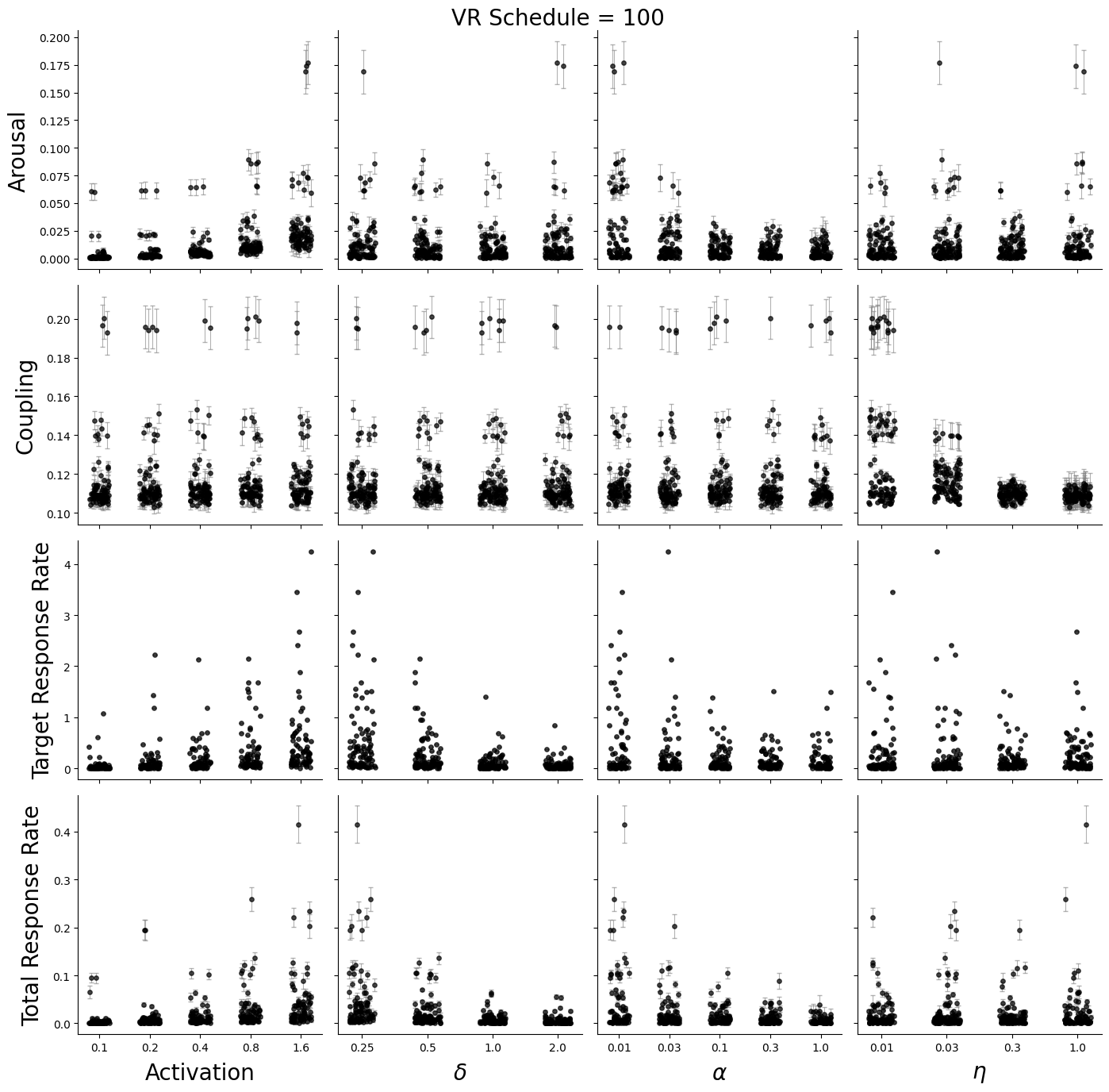
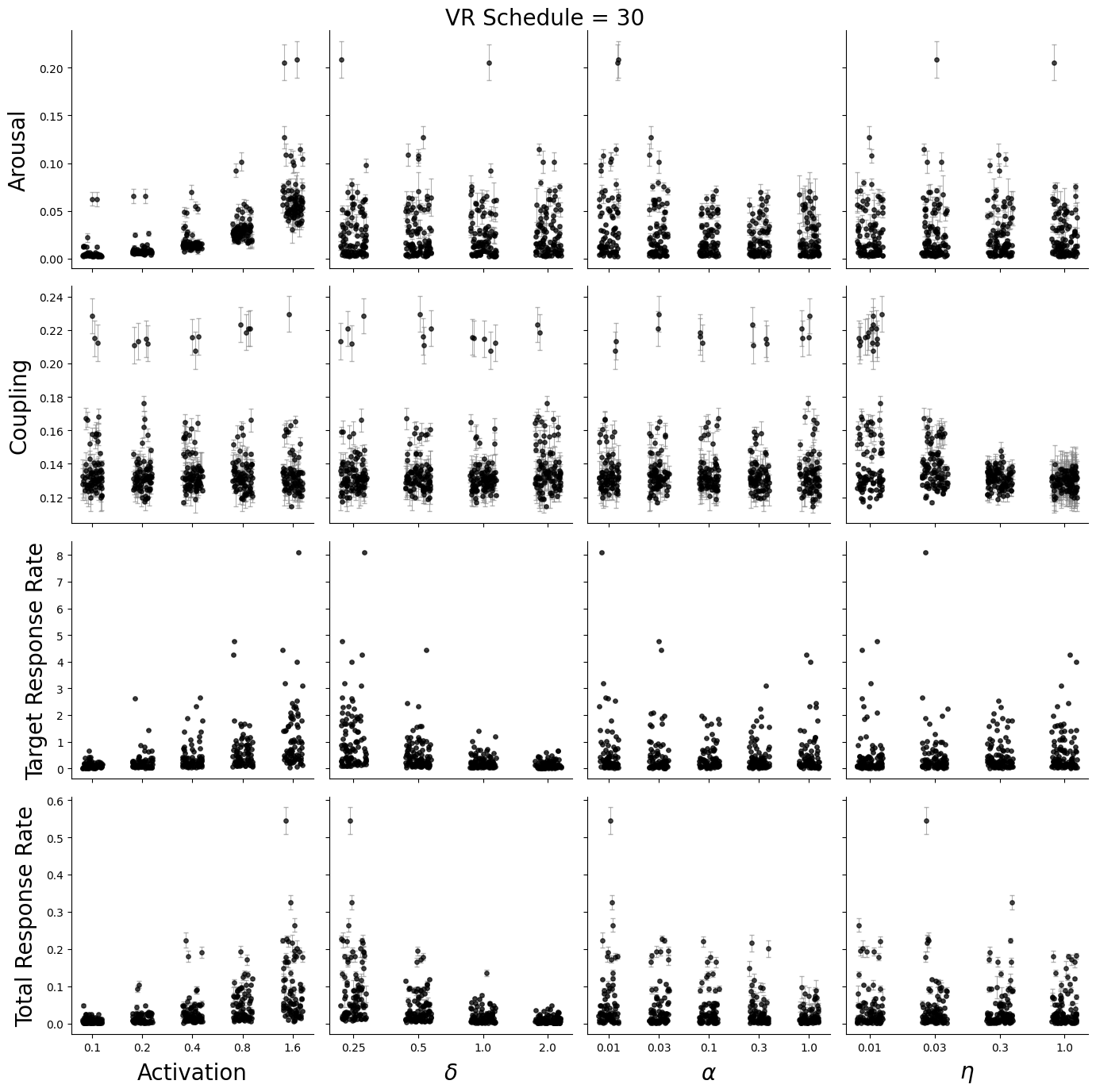
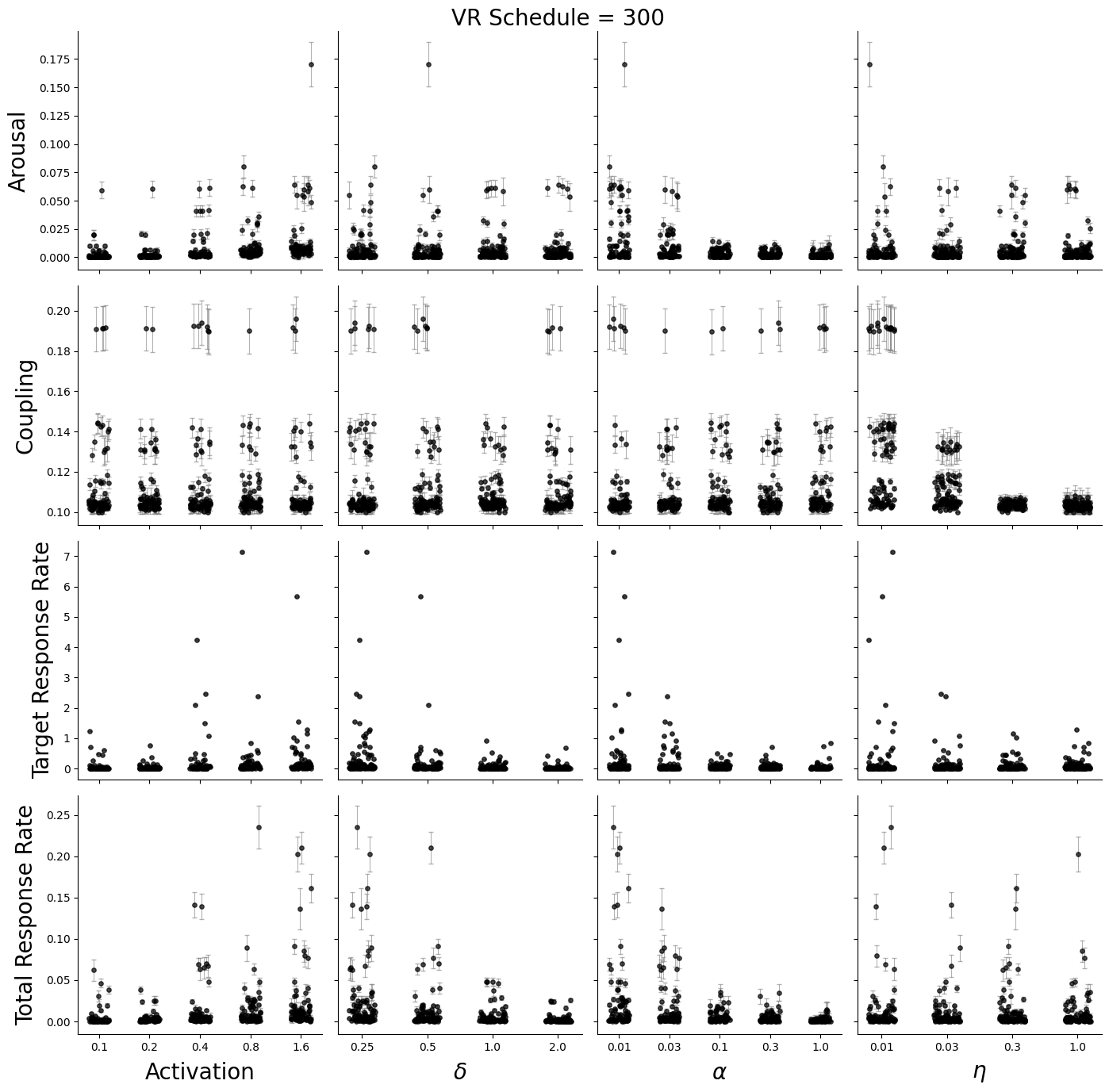
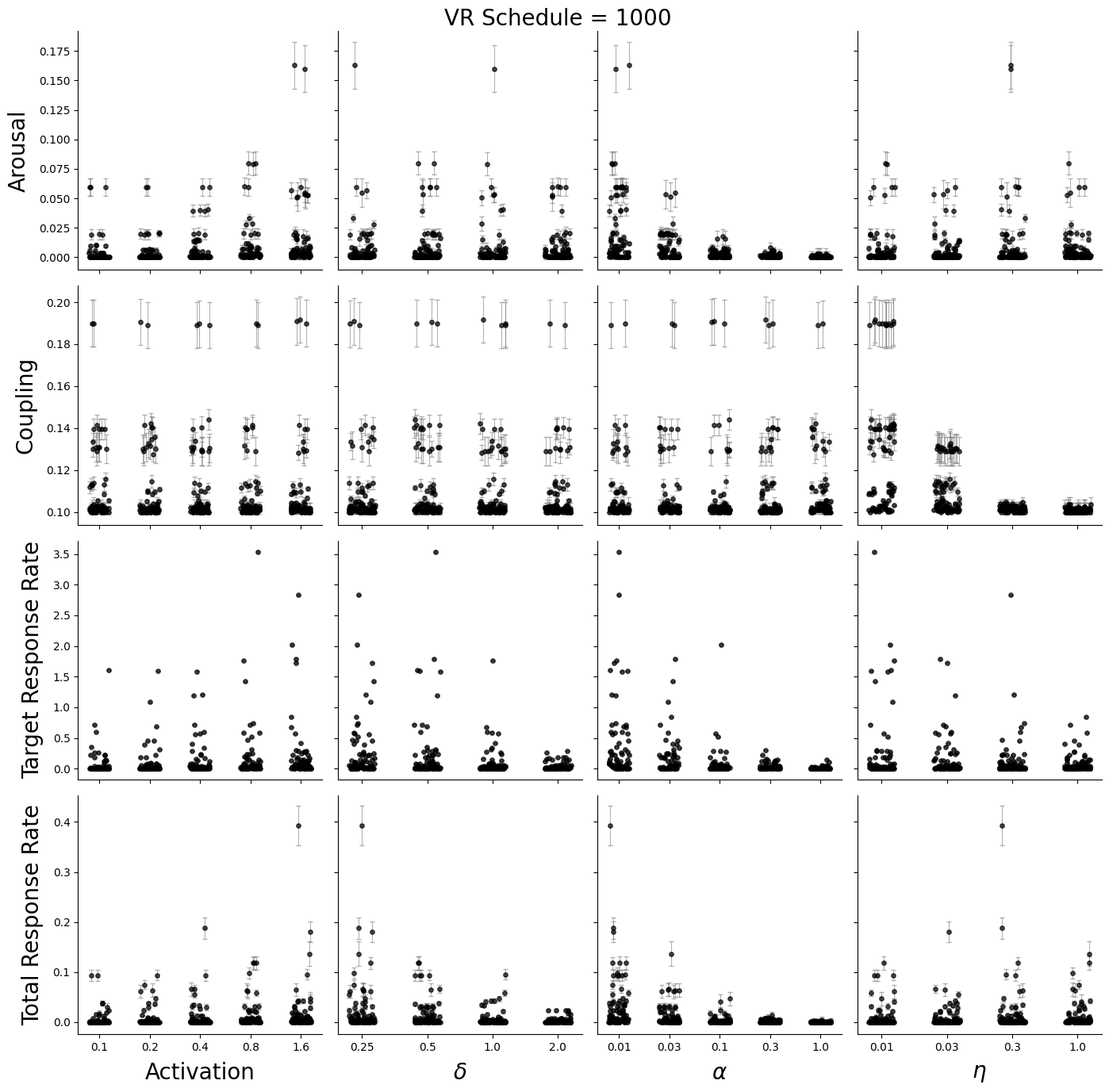
**Figure 1.** Demonstration of a single 1000-step simulation with *δ*=0.25, activation=0.6, arousal=0.6, *α*=0.1, *η*=0.05, and VR-10 as the schedule of reinforcement for the target response. Vertical red lines denote when reinforcement was delivered.



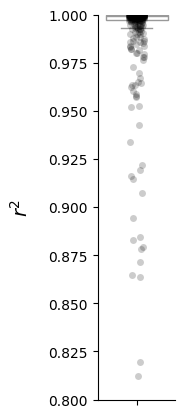
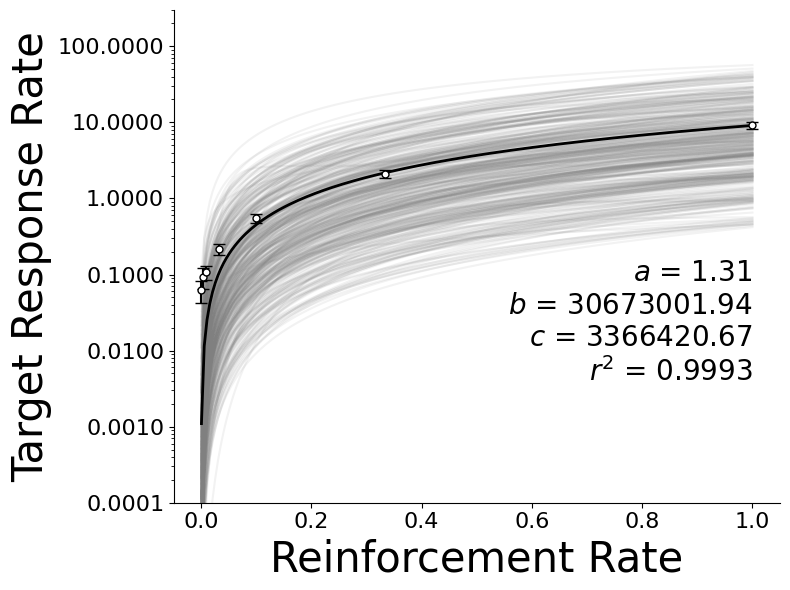
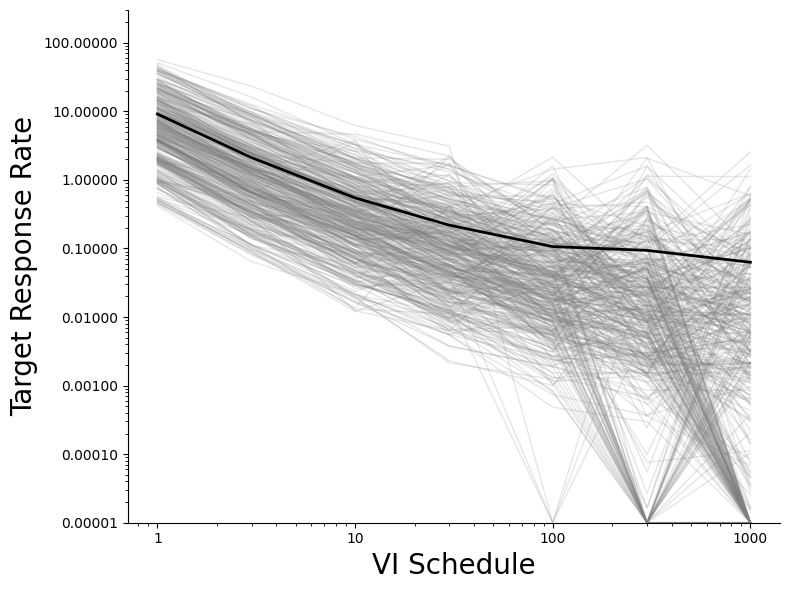
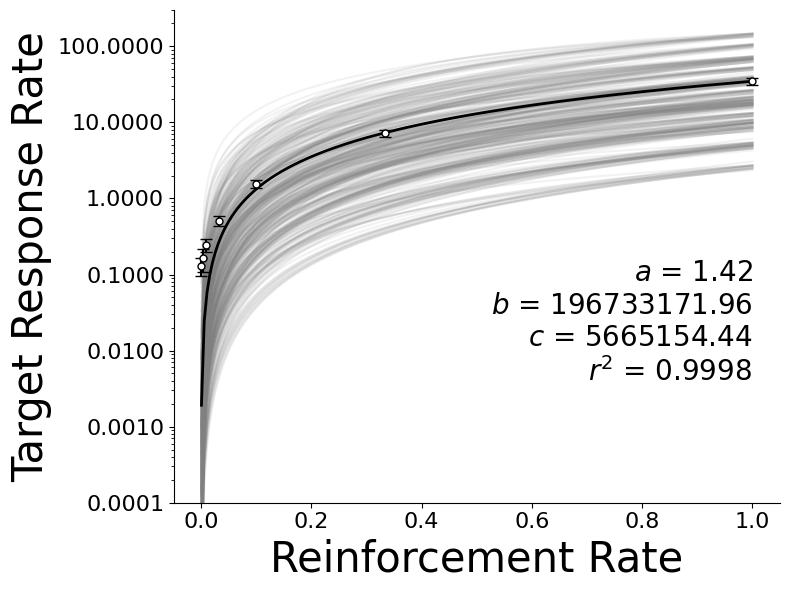
**Figure 2**. AOs animated by MPR responses to single VR schedules (left panel), the predicted values of generalized hyperbola (middle panel), and the *r*2 values for those fits to the AO data (right panel). Each gray line represents a single AO animated by the recursive MPR equations. The black line corresponds to the arithmetic mean of AO behavior (left panel) or the predicted values from the generalized hyperbola (middle panel) when fit to mean AO responding (white markers, middle panel).



**Figure 3**. Relations between average parameters and response rates as a function of varying simulation parameters when responding to a single VR schedule. NOTE: Each marker represents the average from one simulation. Error bars represent 95% confidence intervals.

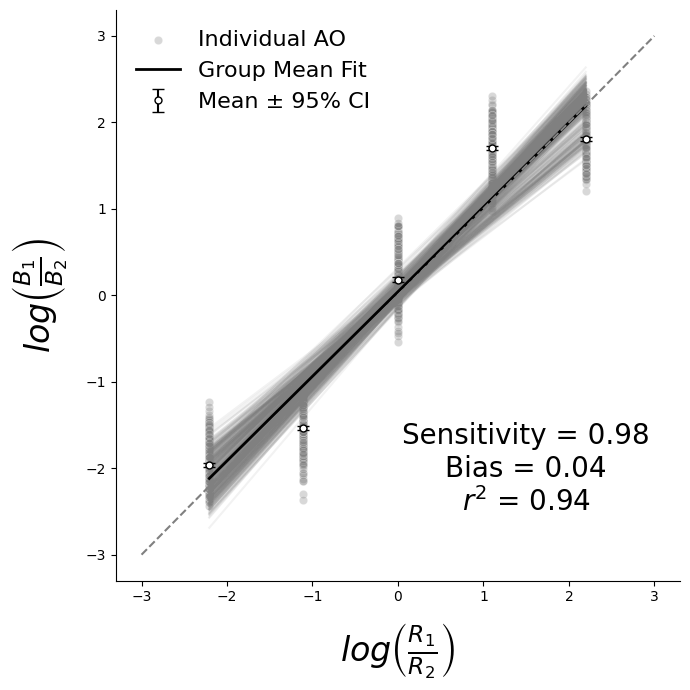
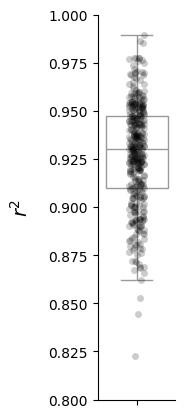
**Figure 4**. AOs animated by MPR responses to single VI schedules (left panel), the predicted values of generalized hyperbola (middle panel), and the *r*2 values for those fits to the AO data (right panel). Each gray line represents a single AO animated by the recursive MPR equations. The black line corresponds to the arithmetic mean of AO behavior (left panel) or the predicted values from the generalized hyperbola (middle panel) when fit to mean AO responding (white markers, middle panel).



**Figure 5**. Relations between average parameters and response rates as a function of varying simulation parameters when responding to a single VI schedule. NOTE: Each marker represents the average from one simulation. Error bars represent 95% confidence intervals.

< Add. >

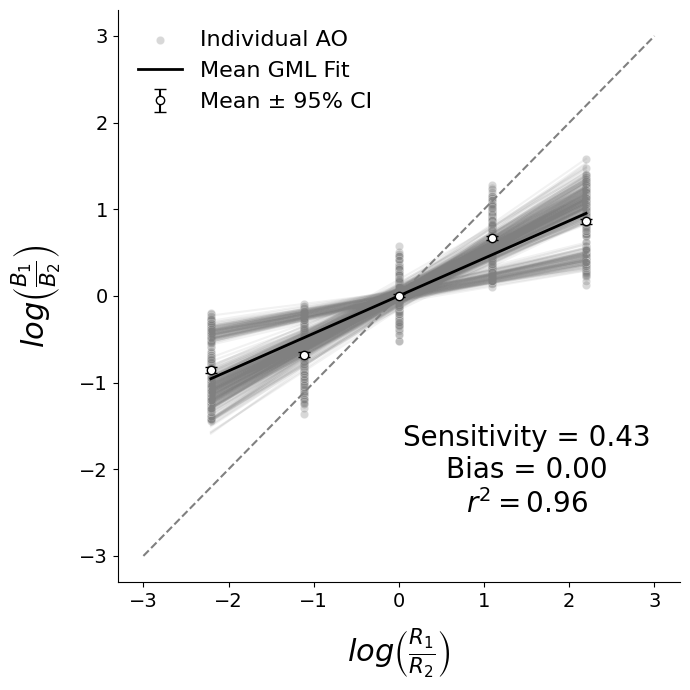
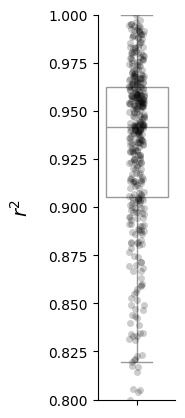
**Figure 6**. Probability matching demonstrated when AOs animated by MPR respond to a range of concurrent VR schedules (left panel) and the corresponding *r*2 values when the generalized matching equation is fit to individual AO behavior (right panel). Gray markers correspond to individual AO behavioral allocation. Each gray line represents a single AO animated by the recursive MPR equations.



**Figure 7**. Relations between average parameters and response rates as a function of varying simulation parameters when for *B*1 onconcurrent VR schedules. NOTE: Each marker represents the average from one simulation. Error bars represent 95% confidence intervals.

< Add. >

**Figure 8**. Probability matching demonstrated when AOs animated by MPR respond to a range of concurrent VI schedules (left panel) and the corresponding *r*2 values when the generalized matching equation is fit to individual AO behavior (right panel). Gray markers correspond to individual AO behavioral allocation. Each gray line represents a single AO animated by the recursive MPR equations.



**Figure 9**. Relations between average parameters and response rates as a function of varying simulation parameters when for *B*1 onconcurrent VI schedules. NOTE: Each marker represents the average from one simulation. Error bars represent 95% confidence intervals.

< Add. >

**Figure 10**. Demo of results when combining recursive theory-driven MPR with the adaptive optimization of Q-learning.

