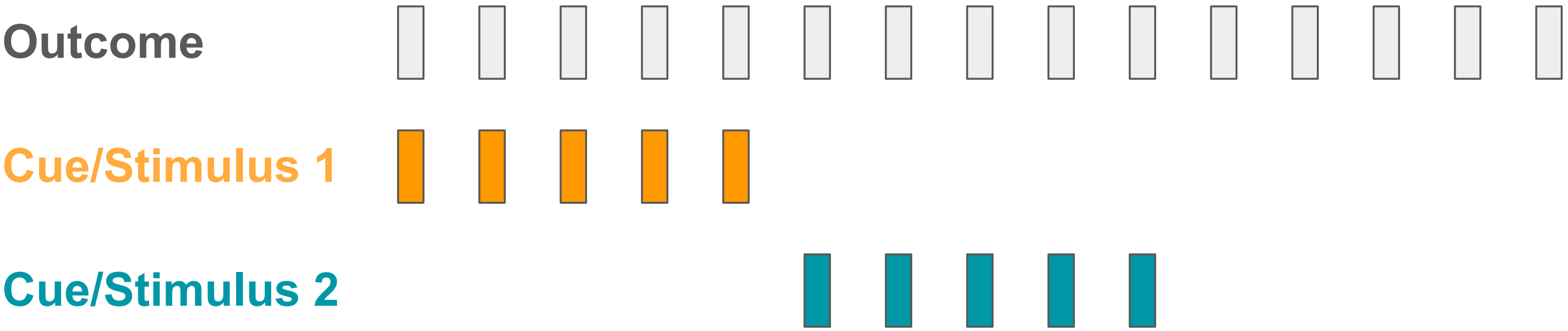
Models of Associative Learning

Sam Gershman (2015) a la Abu

Gershman, Samuel J. “A Unifying Probabilistic View of Associative Learning.” *PLOS Computational Biology*, vol. 11, no. 11, Nov. 2015, p. e1004567. *PLoS Journals*, https://doi.org/10.1371/journal.pcbi.1004567.

Associative Learning

- Reward prediction based on cues/stimuli



Types of models

Estimator	Target: Reward	Target: Value
Point	*Rescorla-Wagner*	*TD learning* (a little)
Bayesian	*Kalman filter*	*Kalman TD* (a little)

Rescorla-Wagner

- $\alpha \in [0, 1]$ = learning rate parameter (*associability*)
- $\delta n = rn - vn$ = *prediction error*
- *vn* is the reward expectation

Model

- Model for effects of "Pavlovian" conditioning
- Independent updating of weights per stimulus

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n,$$

$$v_n = \mathbf{w}_n^\top \mathbf{x}_n$$

$$= \mathrm{w}_0 * \mathrm{x}_0 + \mathrm{w}_1 * \mathrm{x}_1 + \mathrm{w}_2 * \mathrm{x}_2 + \ldots$$

Rescorla RA, Wagner AR. A theory of of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In: Black AH, Prokasy WF, editors. Classical Conditioning II: Current Research and theory. New York, NY: Appleton-Century-Crofts; 1972. p. 64–99.

Rescorla-Wagner

- $\alpha \in [0, 1]$ = learning rate parameter (*associability*)
- $\delta n = rn - vn$ = *prediction error*
- *vn* is the reward expectation

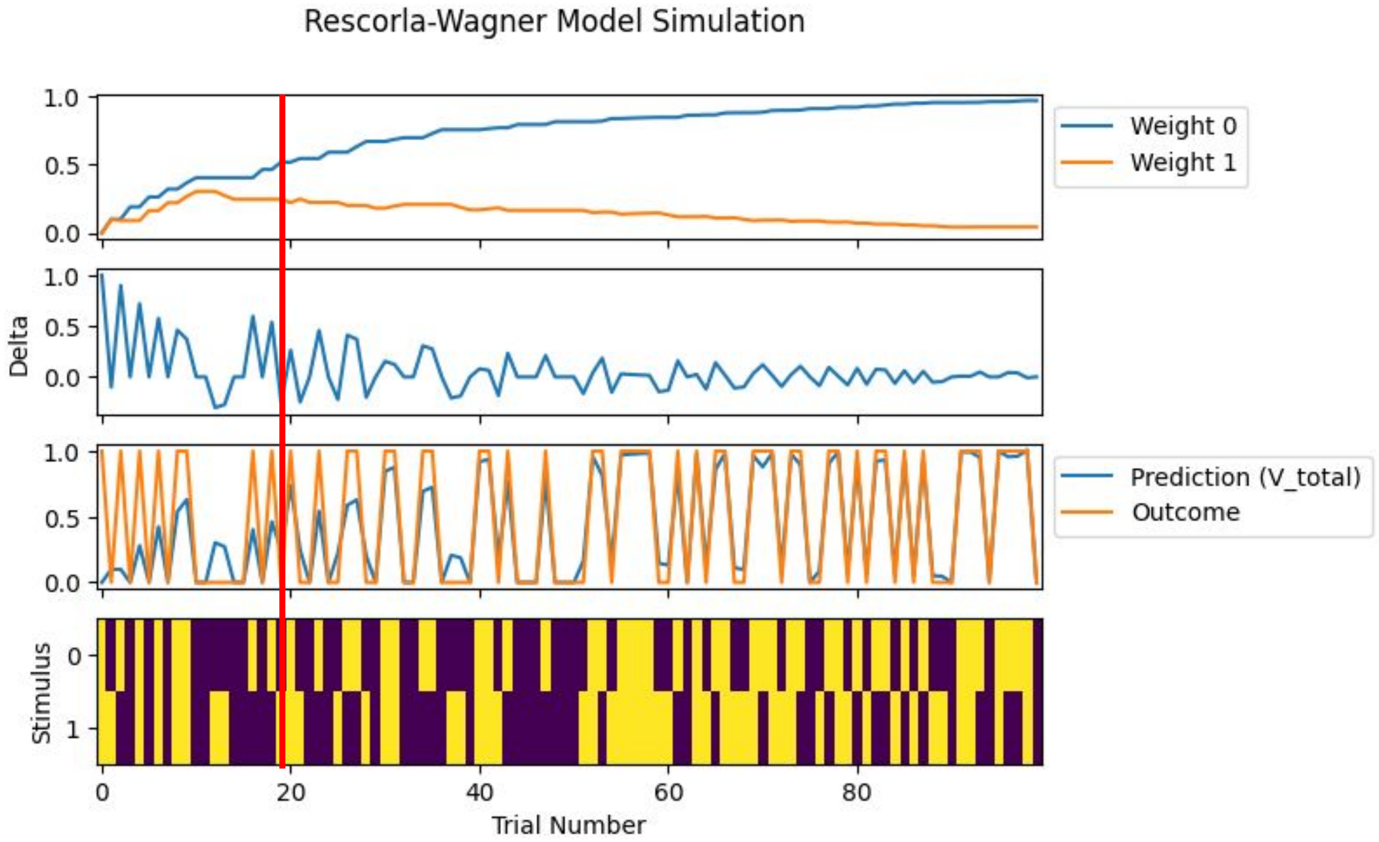
Model

- Independent updating of weights per stimulus

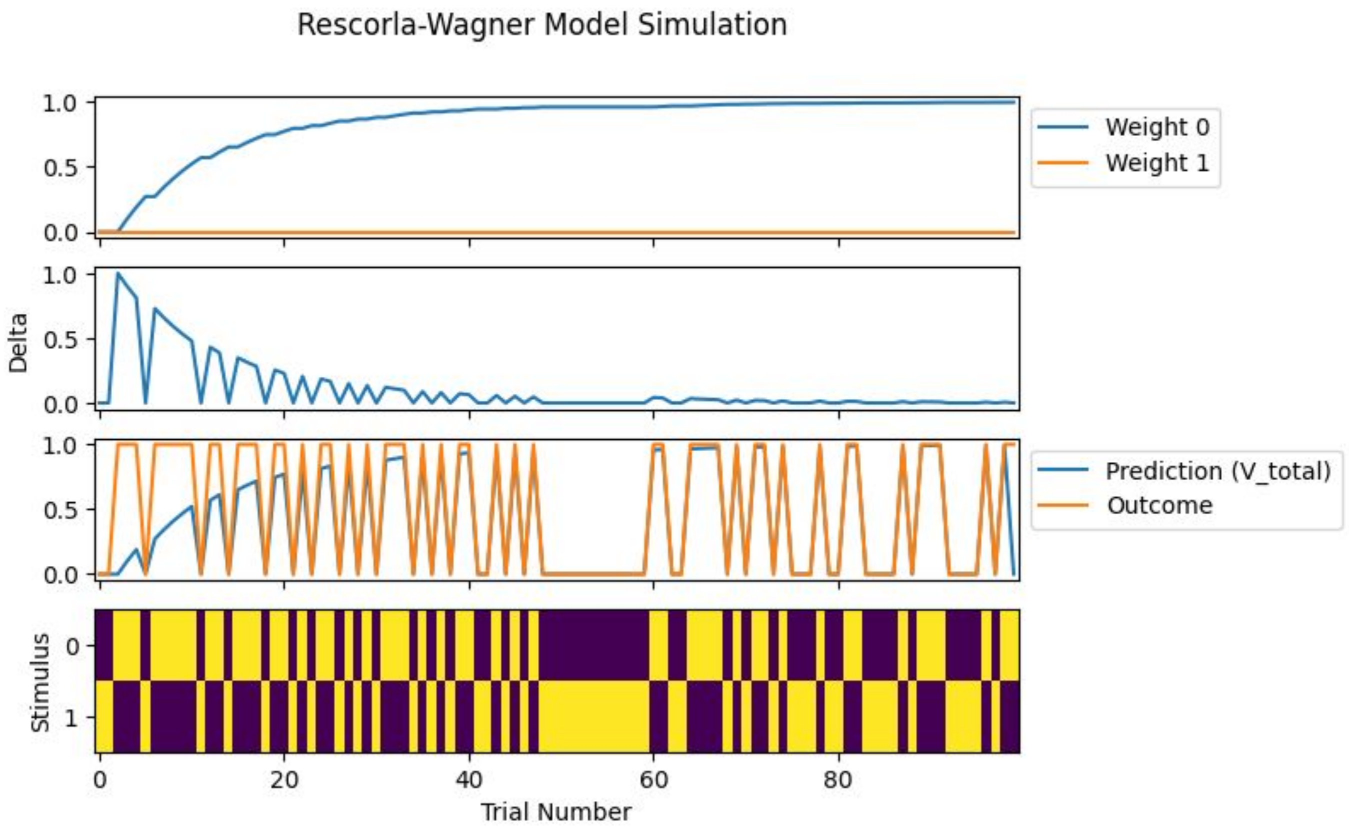
$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n,$$
$$= \mathbf{w}_n + \alpha \mathbf{x}_n (\mathbf{r}_n - \mathbf{v}_n)$$
$$= \mathbf{w}_n + \alpha \mathbf{x}_n (\mathbf{r}_n - \mathbf{w}_n^T \mathbf{x}_n)$$

Rescorla RA, Wagner AR. A theory of of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In: Black AH, Prokasy WF, editors. Classical Conditioning II: Current Research and theory. New York, NY: Appleton-Century-Crofts; 1972. p. 64–99.

Rescorla-Wagner results



Rescorla-Wagner results



Rescorla-Wagner

Pros

- Shows learning is driven by *reward prediction error* which has been experimentally shown
- Skipping more pros (Overexpectation, Conditioned inhibition, Forward blocking)

Cons

- Point estimate of reward…brain keeps track of reward uncertainty
- Learning happens independently for each stimulus
- Cannot account for latent inhibition (in non-taste cases) as pre-exposure does not change weights…hence it cannot affect future learning

$$\mathrm{r}_n = 0, \mathrm{v}_n = 0 \therefore \delta_n = \mathrm{r}_n - \mathrm{v}_n = 0$$

$$\mathrm{w}_{n+1} = \mathrm{w}_n + \alpha \mathrm{x}_n \delta_x = \mathrm{w}_n$$

- No concept of "real" time and temporal delays…only works on discrete trials
  - Therefore, can't model 2nd degree associations

Rescorla-Wagner

- Also no backward blocking

Table 1
Structure of the Training Trials for the Backward-Blocking Paradigm

Phase	Freq.	Cue 1	Cue 2	Outcome
I	10	1	1	1
II	10	1	0	1

Note—Cells with a 1 indicate the presence of a cue or outcome. Cells with a 0 indicate the absence of a cue or outcome.

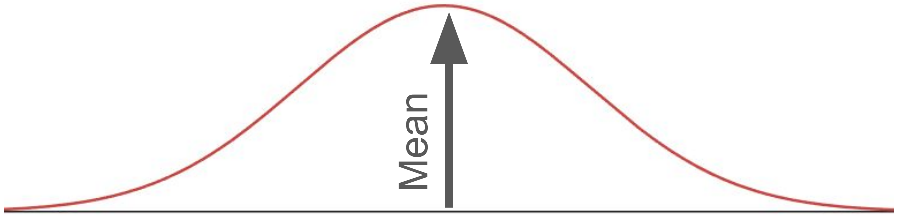
Training: BackwardBlocking
Beginning of Trial 1
Mean: 0 0

Training: BackwardBlocking
Beginning of Trial 11
Mean: 0.413 0.413

Training: BackwardBlocking
Beginning of Trial 21
Mean: 0.745 0.413

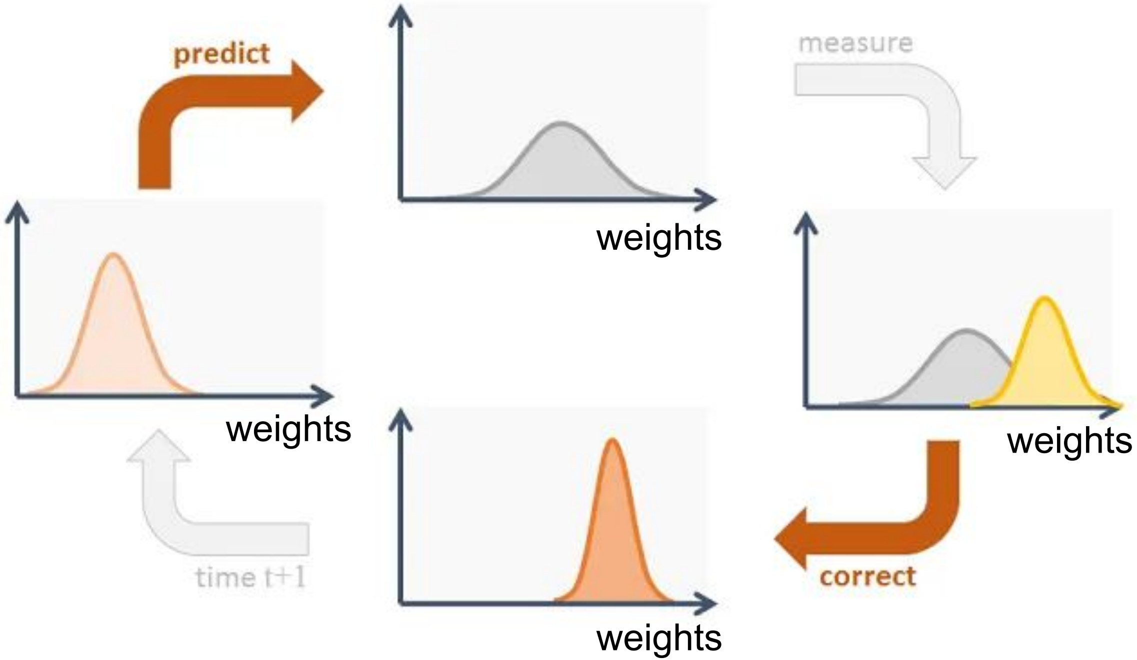
Point-Estimates vs. Bayesian Inference

- Neural and behavioral evidence for uncertainty estimation is abundant
- Rescorla-Wagner = Point-estimate
  - Point estimates of both weights and reward values



How much do I think this cue will result in a reward?
←Less Confident | More Confident →

Kalman Filter - “Bayesian Rescorla-Wagner”



- **<u>Consequences:</u>**
- The more evidence you have, the harder it is to change (“inertia”?)
- Since all probabilities need to add to 1, “Law of Exoneration” is implemented
    - If one cue is learned to more strongly predict an effect, weight of other known cues are diminished

Kalman Filter - “Bayesian Rescorla-Wagner”

Model

- Developed for estimation of linear dynamical systems (1960) and commonly used in robotics for tracking
- Introduced in associative learning in 1992
- Uncertainty weighted updated of weights

Rescorla-Wagner

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n,$$

- *δn = rn − vn = prediction error*
- *vn* is the reward expectation

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mathbf{k}_n \delta_n$$

$$\Sigma_{n+1} = \Sigma_n + \tau^2 \mathbf{I} - \mathbf{k}_n \mathbf{x}_n^\top (\Sigma_n + \tau^2 \mathbf{I}),$$

$$\mathbf{k}_n = \frac{(\Sigma_n + \tau^2 \mathbf{I}) \mathbf{x}_n}{\mathbf{x}_n^\top (\Sigma_n + \tau^2 \mathbf{I}) \mathbf{x}_n + \sigma_r^2}.$$

Kalman Filter - “Bayesian Rescorla-Wagner”

Model

- Uncertainty weighted updated of weights
- Update depends on both stimulus AND covariance between stimuli

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mathbf{k}_n \delta_n$$

$$\Sigma_{\text{with noise}} = \Sigma_n + \tau^2 \mathbf{I}$$

$$\Sigma_{n+1} = \Sigma_{\text{with noise}} - \mathbf{k}_n \mathbf{x}_n^T \Sigma_{\text{with noise}}$$

$$\mathbf{k}_n = \frac{\Sigma_{\text{with noise}} \mathbf{x}_n}{\mathbf{x}_n^T \Sigma_{\text{with noise}} \mathbf{x}_n + \sigma_r^2}$$

Rescorla-Wagner

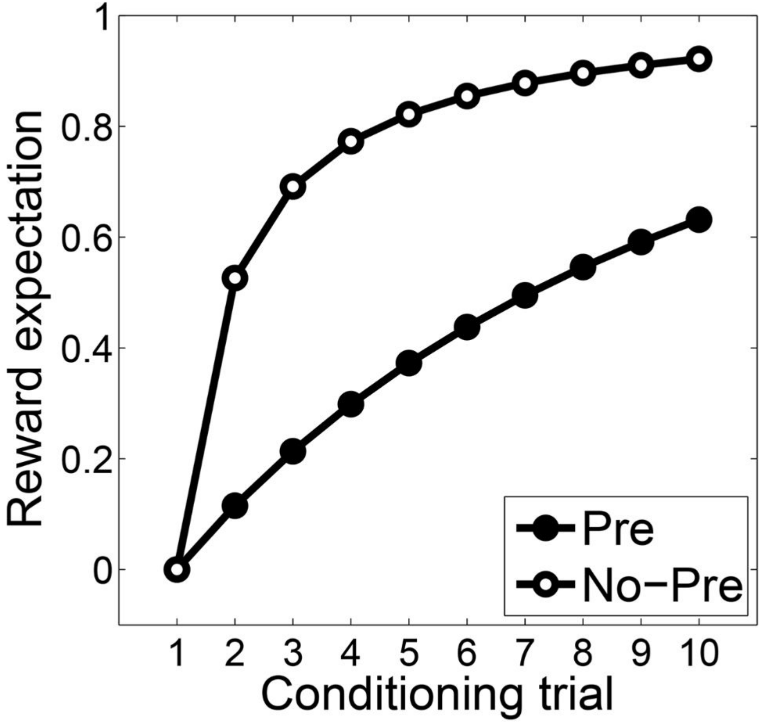
$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n,$$

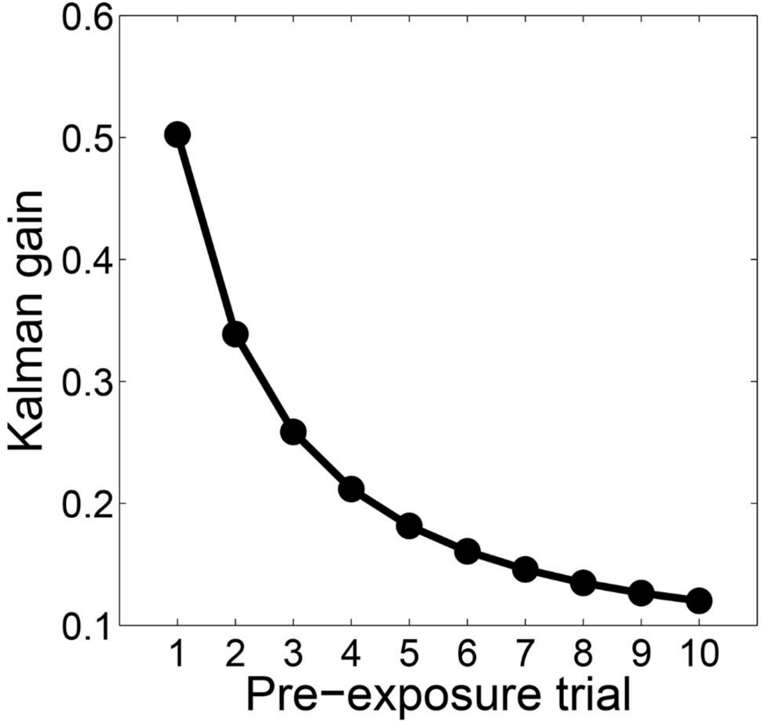
- *δn = rn − vn = prediction error*
- *vn* is the reward expectation

$\sigma_r =$ Prevent divide by 0

Kalman Filter - Latent Inhibition

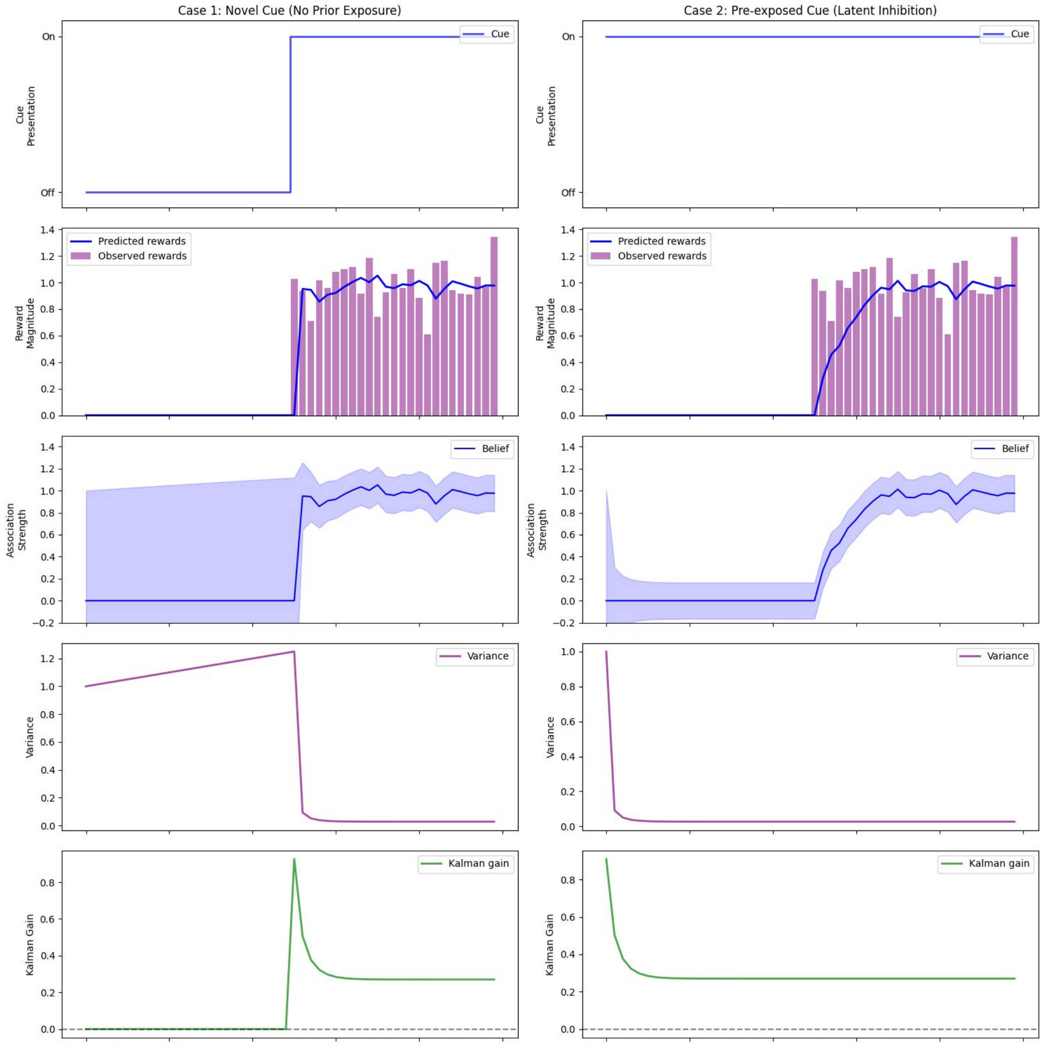
INERTIA!!





Kalman Filter - Latent Inhibition

INERTIA!!



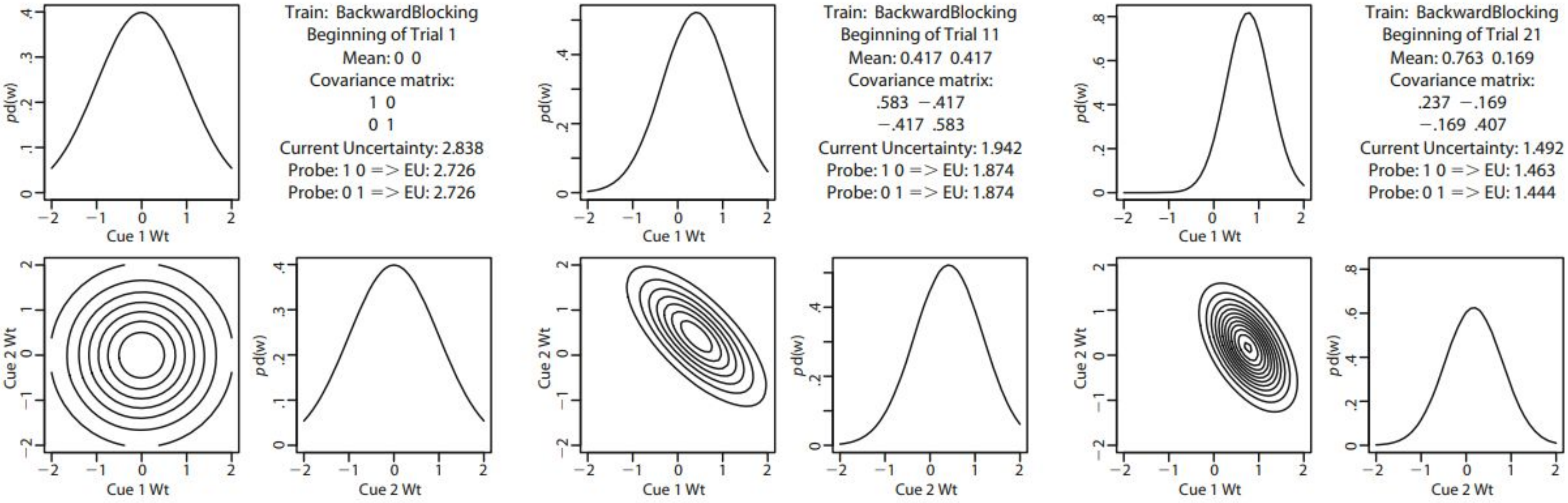
Kalman Filter - Back Block

LAW OF EXONERATION!

Table 1
Structure of the Training Trials for the Backward-Blocking Paradigm

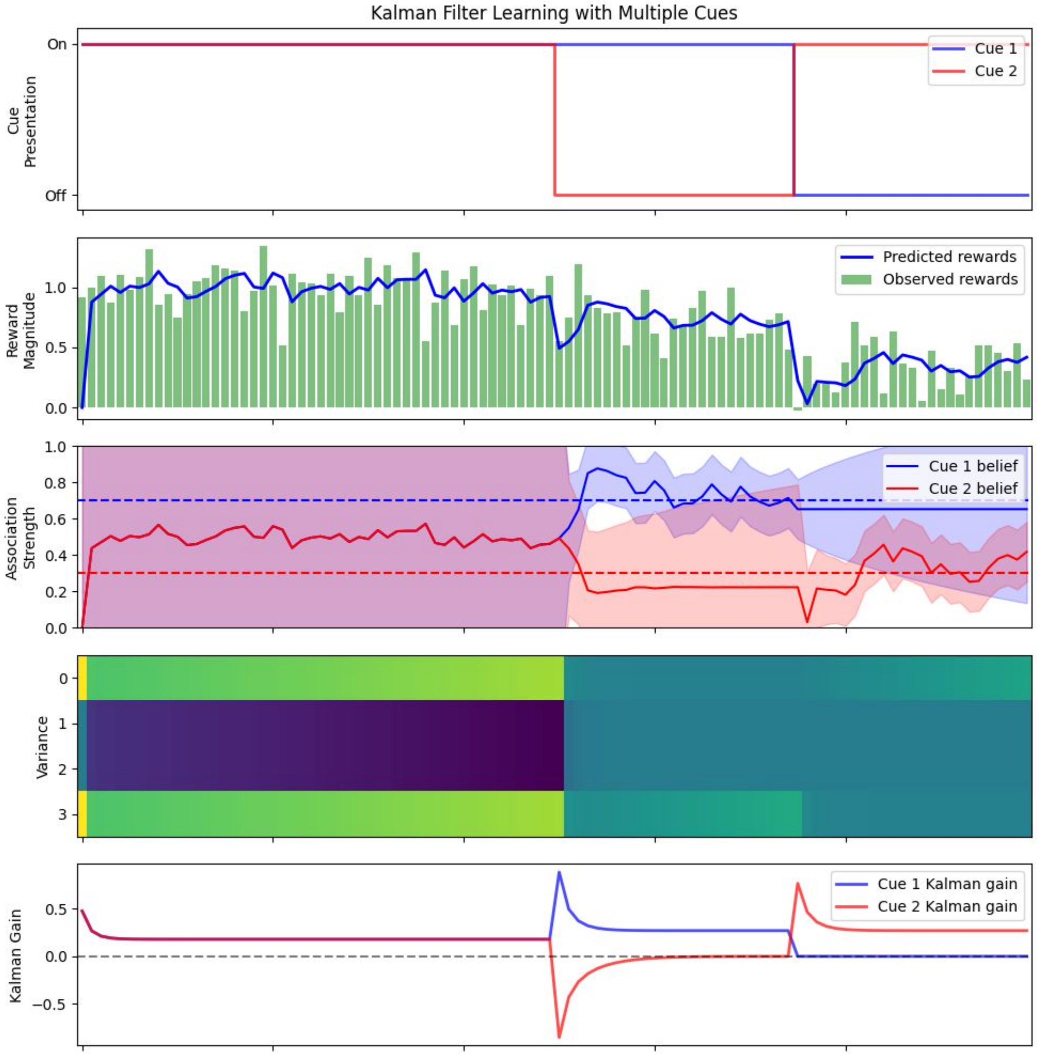
Phase	Freq.	Cue 1	Cue 2	Outcome
I	10	1	1	1
II	10	1	0	1

Note—Cells with a 1 indicate the presence of a cue or outcome. Cells with a 0 indicate the absence of a cue or outcome.



Kalman Filter
- Back Block

LAW OF EXONERATION!



Intermission

- Kalman Filter is a Bayesian extension of the Rescorla-Wagner model

- Both operate in “trial” time
- No concept of “ongoing” time…so no ability to represent intra-trial timing
- This allows for only direct associations between CS and US (coincidence detection)…no ability to develop secondary associations

Temporal Difference Learning

- One of the first reinforcement learning algorithm taught for "complex problems" with delayed reward
- Developed 1983 (cart-pole balancing problem)
- Applied to associative learning in 1990
- Non-bayesian (point estimates of weights)
- But can deal with non-coincident cue and reward

$$\hat{\mathbf{w}}_{t+1} = \hat{\mathbf{w}}_t + \alpha \mathbf{x}_t \delta_t,$$

$$\delta_t = r_t + \underline{\gamma \hat{\mathbf{w}}_t^\top \mathbf{x}_{t+1} - \hat{\mathbf{w}}_t^\top \mathbf{x}_t}.$$

Temporal Difference Prediction Error

Rescorla-Wagner

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \alpha \mathbf{x}_n \delta_n,$$

- $\delta n = rn - vn$ = *prediction error*
- vn is the reward expectation

- γ = Discount factor to devalue future rewards (constraint to maximize reward / time)

Temporal Difference Learning - Second-order learning (hand-wavy)

- Rescorla-Wagner and Kalman filter optimize instantaneous predictive accuracy of reward given stimuli
- Temporal Difference Learning optimizes cumulative reward (predictability) over the entire session (across all time in a session)
- 2nd order learning:= Order of stimuli : A+B → B+Reward
  - Stimulus A becomes associated with the reward despite no direct pairing
  - This is not possible using Rescorla-Wagner model or Kalman Filter
- Given that TD learning attempts to optimize for predictive capacity over all time, the system learns that presentation of stimulus A will eventually predict Reward

Kalman Temporal Difference Learning

- Temporal delays with Bayesian inference
- Estimating distribution over / uncertainty of cumulative future rewards
- “These properties allow the model to capture both within-trial structure and retrospective revaluation.”
- What does this add?
  - Recapitulating paradigms with second-order conditioning + updating of unpresented stimuli
  - Paradigms got very complicated…please read paper :p