Validating model-based Bayesian integration using prior-cost metamers

Hansem Sohn, Mehrdad Jazayeri

This file includes:

Supplementary Materials and Methods

RSG task

Stimuli were presented on a fronto-parallel 23-inch display (distance: approximately 67 cm; refresh rate: 60 Hz; resolution: 1920 by 1200) and behavioral responses were registered using a standard Apple keyboard. Participants' task was to measure a sample time interval (t_s) between two visual flashes ('Ready' and 'Set') and subsequently initiate a delayed motor response ('Go') such that the produced interval relative to Set (t_p) would match t_s as accurately as possible. Each trial began with the presentation of a central fixation point (FP; white circle, diameter: 0.5 degree in visual angle) along with a peripheral stimulus (white circle, diameter: 0.25 deg) that was presented 5 deg to the left of FP. We asked participants to maintain their gaze on FP throughout the trial but their eye positions were not monitored. The peripheral stimulus only served as a spatial reference and was otherwise irrelevant. After a random delay (exponentially distributed; mean of 500 ms with a minimum of 250 ms), Ready and Set flashes (white circle, diameter: 0.25 deg; flash duration: 50 ms) were presented in sequence to the right and top of FP. On each trial, t_s was sampled from the prior probability distribution (see below for details about the prior). After Set, participants had to produce a matching interval by pressing the spacebar on a keyboard and received a graded numerical feedback ('reward') below FP based on the magnitude of the error (see below for details about the cost/reward function). Trials were separated by a fixed 500 ms inter-trial interval.

VMR task

Participants viewed the stimuli from above on a 21.5-inch Samsung SyncMaster SA200 display (distance: approximately 60 cm; refresh rate: 60 Hz; resolution: 1920 by 1080) and responded by controlling a custom manipulandum built from a stylus for a pen digitizer tablet (Wacom Intuous5 touch) with their hand occluded by the display. Participants' task was to use a manipulandum to move a partially occluded cursor from its initial position at the center of the display to a target positioned over the circumference of a ring around the initial position. Each trial began with the presentation of a central point (FP; red square, size: 1 deg). To proceed, participants had to use the manipulandum to move the cursor, which at this stage of the trial was visible, to the center of the screen (over FP). As soon as the cursor entered an electronic circular window (diameter: 0.5 deg) around FP, a visual target (gray circle, contrast: 40%, diameter: 1 deg) was presented briefly (duration: 500 ms) on a circular ring (radius: 10 deg, thickness: 0.1 deg, contrast: 50%) around FP. The ring remained visible throughout the trial. The position of the target was chosen from a uniform distribution between 80 to 100 deg with its midpoint aligned to 12 o'clock over the circle. Afterwards, participants had to move the cursor to the remembered location of the flashed target. The cursor was made invisible as soon as it passed the electronic window, and remained invisible throughout its movement except for a brief reappearance (gray circle, contrast: 40%, diameter: 0.5 deg, duration: 100 ms) when it was 3 deg from FP. We titrated the contrast and duration of the reappearing mid-movement cursor so that participants' measurement of the cursor position was not too accurate to perform the task. Participants were asked to move the cursor smoothly with no interruption (movement duration from the inner fixation window to the target: 0.96 ± 0.60 sec).

Critically, while the cursor was invisible and before its reappearance at 3 deg from FP, we applied an angular rotation (x_s) to the cursor relative to the hand position. On each trial, x_s was taken from a prior distribution (see below for details about the prior). Participants had to use their knowledge about the prior distribution and the visual feedback along the path to infer x_s and maneuver the cursor to produce a counter rotation (x_p) that matches x_s as accurately as possible. We defined x_p as the angular distance between the target and hand position over the ring to conceptually match the dependent variables between the RSG and VMR tasks (i.e., slope of 1 between x_p and x_s indicates no bias and full compensation of the visuomotor rotation; see Supplementary figure 9 for data from all participants). Note that this is different from how the dependent variable was defined in the original work (11). Similar to RSG, participants received a graded numerical feedback below FP based on the magnitude of the error (see below for details about the cost/reward function). Trials were separated by a fixed 500 ms inter-trial interval. For VMR, we asked participants to use their dominant hand throughout the experiment.

Prior-cost metamers for RSG

On each trial, t_s was sampled from a prior distribution, and the numerical feedback was computed based on an experimentally imposed cost function. We measured performance in the context of two prior-cost pairs, the 'original' pair, which was introduced in the first session, and its 'metamer' that was additionally introduced in the remaining sessions.

We denote the original prior by $p_o(t_s)$ and the original cost function by $C_o(t_e,t_s)$. For all participants, $p_o(t_s)$ was a Gaussian distribution with a mean of 750 ms and a standard deviation of 144 ms (Figure 2c). $C_o(t_e,t_s)$ was a concave quadratic function of error (t_e-t_s) with a y-intercept of 100 such that participants received a maximum score of 100 when t_e=t_s (Figure 2c). For errors larger than 1000 ms associated with negative scores, we truncated the cost function and set the score to 0. Mathematically, $p_o(t_s)$ and $C_o(t_e,t_s)$ were defined as follows:

$$p_o(t_s) = \frac{1}{144\sqrt{2\pi}} e^{-\frac{(t_s-750)^2}{2(144)^2}} \quad \text{(Equation 1)}$$

$$C_o(t_e,t_s) = \begin{cases} 100 - \frac{9.9649}{10^5}(t_e - t_s)^2 & |t_e - t_s| < 1000^{ms} \\ 0 & \text{otherwise.} \end{cases} \quad \text{(Equation 2)}$$

Following previous work (44), we used a generative model to compute the ideal observer policy associated with the original prior-cost pair. We assumed that the ideal observer measurement of t_s, denoted by t_m, is perturbed by zero-mean Gaussian noise. Based on the scalar property of timing variability (1), we additionally assumed that the standard deviation of noise scales with t_s with a constant of proportionality w_m, which we refer to as the Weber fraction for measurement. Accordingly, we formulated the likelihood function, $p(t_m|t_s)$ as $N(t_s, w_m t_s)$. We computed the ideal observer policy for the original pair, denoted by f_o^{ideal}, by integrating $p(t_m|t_s)$ with $p_o(t_s)$ and the original cost function by $C_o(t_e,t_s)$ and maximizing the expected reward:

$$f_o^{\text{ideal}}(t_m) = \arg\max_{t_e} \int p(t_m|t_s) p_o(t_s) C_o(t_e,t_s) dt_s \quad \text{(Equation 3)}$$

To compare the behavior of the participants to that of the ideal observer policy, we augmented the model to include production noise such that the produced interval, t_p, was a sample from a Gaussian distribution with mean t_e, and standard deviation $w_p t_e$ ($N(t_e, w_p t_e)$) where w_p reflects additional scalar variability associated with the production (2). We fit the ideal observer model to data from the first session to compare participants' behavior to the optimal policy and to estimate each participant's w_m and w_p, which remained relatively stable throughout the entire experiment (Supplementary figure 11). Note that the scores participants received during the experiment were based on the error in t_p, not t_e, since t_e is not observable. However, we verified that the expected cost function with t_e is similar to the cost function with t_p when the production noise is symmetric with respect to t_e. In computing the ideal observer policy, we did not take into account the fact that the quadratic cost function was truncated as errors were rarely within the truncated sidebands (mean±SD: 0.0064±0.0211% across subjects).

For the subsequent experimental sessions, we had to design a metamer for the original prior-cost pair. To do so, we considered a new prior, $p_m(t_s)$ and a new cost function, $C_m(t_e,t_s)$ such that the ideal observer policy for the metamer, f_m^{ideal} would be identical to f_o^{ideal}. Mathematically, this imposed the following equality constraint:

$$\arg\max_{t_e} \int p(t_m|t_s)p_o(t_s)C_o(t_e,t_s)dt_s = \arg\max_{t_e} \int p(t_m|t_s)p_m(t_s)C_m(t_e,t_s)dt_s \quad \text{(Equation 4)}$$

Evidently, there are an infinitude of solutions to this problem since we can freely adjust both $p_m(t_s)$ and $C_m(t_e,t_s)$. For the purpose of our experiment, we wanted the original and its metamer to be generally similar but different enough to provide statistical power for detecting a transient learning effect. We focused on a specific form of manipulation that involved a shift in the cost function toward lower values, and commensurate adjustments to the prior to satisfy the constraint of generating a metamer. Accordingly, we defined $C_m(t_e,t_s)$ to be the same as $C_o(t_e,t_s)$ but shifted by an interval, Δt. Note that we favored this choice because of its computational simplicity but other manipulations to the cost function (e.g., introducing asymmetry) are also permissible.

Another factor we had to consider beyond matching the general form of the ideal observer policy was to make sure that the average score a participant received (the integral term in Equation 4) was similar between the original pair and its metamer. This was important because a sudden change in average reward (score) could signal a change in the environment and motivate learning a new policy. To address this point, we scaled $C_m(t_e,t_s)$ by a factor k whose value was adjusted so that the expected reward for the two pairs were the same. With these considerations in mind, the new cost function was formulated as follows:

$$C_m(t_e,t_s) = \begin{cases} k(100 - \frac{9.9649}{10^5}(t_e - t_s - \Delta t)^2) & |t_e - t_s - \Delta t| < 1000^{ms} \\ 0 & \text{otherwise.} \end{cases} \quad \text{(Equation 5)}$$

Using simulations, we determined that, depending on the Weber fraction, a shift in the cost function in the order of 100 to 200 ms (larger shifts for larger Weber fractions) was sufficient for detecting learning effects. Accordingly, we set Δt to 150 ms if the fitted w_m was less than 0.2, and 200 ms otherwise. Matching the average score also provided a participant-specific value for k (mean across participants: 0.73, SD: 0.06).

Having fixed the choice of $C_m(t_e,t_s)$, we next had to design $p_m(t_s)$. To do so, we modeled $p_m(t_s)$ as a Gaussian mixture model (GMM), $\sum w_k N(\mu_k,\sigma_k)$, and estimated w_k and σ_k such that the new cost-prior pair was associated with the same ideal observer policy as the original pair (Equation 4; 'fmincon' function in MATLAB). To make this optimization process more tractable, we fixed the number of components in the GMM to 20 and fixed their corresponding mean to be equidistant between 350 to 1300 ms. Note that $p_m(t_s)$ had to be customized for each participant separately because the exact shape $p_m(t_s)$ depends on w_m.

We then used the new prior-cost pair in the subsequent sessions. For the second and fourth sessions, we started the experiment with the original pair and then switched covertly to the metamer pair. For the third and fifth sessions, the order of the original and its metamer were switched. In all these sessions, the first prior-cost pair was present for the first 170 trials, and the other pair was present for the remaining trials (approximately 590 trials). More trials were dedicated to the post-transition so that we could observe the learning-related effects for longer periods.

Prior-cost metamers for VMR

The original prior, $p_o(x_s)$ and cost function $C_o(x_e,x_s)$ were governed by the following equations:

$$p_o(x_s) = \frac{1}{7.5\sqrt{2\pi}} e^{-\frac{(x_s-(-12))^2}{2(7.5)^2}}$$ (Equation 6)

$$C_o(x_e, x_s) = \begin{cases} 100 - 0.111(x_e - x_s)^2 & |x_e - x_s| < 30^\circ \\ 0 & \text{otherwise.} \end{cases}$$ (Equation 7)

To compute the ideal observer policy for the VMR task, we assumed that the measured angle, x_m is subject to zero-mean Gaussian noise, and formulated the likelihood function, $p(x_m|x_s)$ as $N(x_s, \sigma_m)$. In this setting with a Gaussian prior and a Gaussian likelihood function, f_o^{ideal} is a simple linear mapping between x_e and x_m as follows:

$$x_e = \frac{7.5^2}{7.5^2 + \sigma_m^2} x_m + \frac{\sigma_m^2}{7.5^2 + \sigma_m^2}(-12)$$ (Equation 8)

To compare the participants' behavior to the ideal observer, we additionally assumed that the produced angular correction, x_p, was also subject to zero-mean Gaussian noise with standard deviation, σ_p; i.e., $x_p \sim N(x_e, \sigma_p)$. We fit the ideal observer model to data from the first session to compare participants' behavior to the optimal policy and to estimate each participant's σ_m and σ_p, which remained relatively stable throughout the entire experiment (Supplementary figure 12). Note that we defined the cost function in terms of error in x_e, and computed the scores based on the error in x_p. Similar to the RSG task, we verified that the expected cost function with x_e is similar to the cost function with x_p when the production noise is symmetric with respect to x_e.

Our procedure for devising a metamer for the original prior-cost pair was as follows: we formulated $p_m(x_s)$ as a Gaussian distribution that was shifted by 10 deg, and $C_m(x_e, x_s)$ as a quadratic function that was shifted in the opposite direction, as follows:

$$\arg\max_{x_e} \int p(x_m|x_s) p_o(x_s) C_o(x_e, x_s) dx_s = \arg\max_{x_e} \int p(x_m|x_s) p_m(x_s) C_m(x_e, x_s) dx_s$$ (Equation 9)

Note that the design of the metamer for the VMR is significantly easier than the RSG task because, in the VMR task, changing the mean of the prior only alters the intercept of the linear policy and not its slope, which can be readily compensated by an opposite shift in the cost function. We computed the requisite shift in the cost function (Δx) analytically by matching f_o^{ideal} to f_m^{ideal} as follows:

$$C_m(x_e, x_s) = \begin{cases} k(100 - 0.111(x_e - x_s - \Delta x)^2) & |x_e - x_s - \Delta x| < 30^\circ \\ 0 & \text{otherwise.} \end{cases}$$ (Equation 10)

where Δx can be computed as follows:

$$\Delta x = -10 \frac{\sigma_m^2}{7.5^2 + \sigma_m^2}$$ (Equation 11)

Finally, we used the original and metamer prior-cost pairs in the subsequent sessions using the same procedure as in the RSG task.

Models and Analysis

All analyses were performed using custom code in MATLAB (Mathworks, Inc.). We first removed outlier trials for each data set across all participants and sessions. We applied different algorithms to the two tasks as their response profile was inherently different (i.e., nonlinear metronomic function for the RSG task and linear policy for the VMR). For the RSG task, we excluded trials in which the relative error, defined as $(t_p-t_s)/t_s$, deviated more than 3 standard deviations from its mean (mean: 0.50%; SD: 0.28% across subjects). For the VMR task, we first fitted a linear regression model relating x_p and x_s, and excluded trials for which the error from the linear fit was more than 3.5 times larger than the median absolute deviation (mean: 3.4%; SD: 2% across data sets). We verified that outlier trials were not concentrated immediately after the switch between the prior-cost pairs.

Decision policy for the implicit (H_1) and explicit (H_2) optimal strategies in the RSG task

For the original prior-cost pair, which uses a quadratic cost function, the optimal policy is to choose the mean of the posterior:

$$t_e = f_o(t_m) = \frac{\int t_s p(t_m \mid t_s) p_o(t_s) dt_s}{\int p(t_m \mid t_s) p_o(t_s) dt_s} \quad \text{(Equation 12)}$$

This function captures the asymptotic optimal policy for both the implicit and explicit learning strategies because the two pairs were designed to be metamers. The same optimal policy can also be written in terms of the $p_m(t_s)$ as follows:

$$f_m(t_m) = \frac{\int t_s p(t_m \mid t_s) p_m(t_s) dt_s}{\int p(t_m \mid t_s) p_m(t_s) dt_s} + \Delta t \quad \text{(Equation 13)}$$

Note that the shift of the cost function, Δt, adds a constant offset to the policy, which is compensated by the change in the prior.

Decision policy for prior-sensitive (H_3) and cost-sensitive (H_4) suboptimal strategies

Under H_3, the observer only learns the new prior and assumes that there has been no change in the quadratic cost function. The corresponding decision policy is therefore to use the mean of the posterior under the new prior:

$$f_{ps}(t_m) = \frac{\int t_s p(t_m \mid t_s) p_m(t_s) dt_s}{\int p(t_m \mid t_s) p_m(t_s) dt_s} \quad \text{(Equation 14)}$$

Under H_4, the observer only learns the new cost function and assumes that there has been no change in the Gaussian prior. The corresponding decision policy can be computed as follows:

$$f_{cs}(t_m) = \frac{\int t_s p(t_m \mid t_s) p_o(t_s) dt_s}{\int p(t_m \mid t_s) p_o(t_s) dt_s} + \Delta t \quad \text{(Equation 15)}$$

The policy for the prior-sensitive and cost-sensitive models has the same form as $f_m(t_m)$ and $f_o(t_m)$, respectively, but their intercepts are different from the optimal policies (Figure 2d).

Model comparison after switch

To distinguish between H_1, H_2, H_3, and H_4, we compared the log-likelihood of the data under the four models (Equations 12-15) and asked which model better captures behavior of the participants immediately after the switch between the two prior-cost pairs. The likelihood for each trial (superscript *i*) under each model (*j*) can be computed as follows:

$$p(t_p^i \mid t_s^i) = \int p(t_p^i \mid f_j(t_m), w_p) p(t_m \mid t_s^i, w_m) dt_m \quad \text{(Equation 16)}$$

The first and second term within the integral correspond to the noisy generative process of production and measurement, respectively (w_p and w_m for its Weber faction). $f_j(t_m)$ can be replaced by f_o (H_1/H_2: implicit and explicit optimal policy; *p(optimal)* in Figure 3a), f_{ps} (H_3: prior-sensitive), and f_{cs} (H_4: cost-sensitive) to test the data under the corresponding hypotheses (Figure 4c). Note that the likelihood in Equation 16 can be more sensitive than other metrics that directly rely on raw behavioral data because it took into account the subject-specific measurement and production noises, w_p and w_m, which were independently measured in the first session (see Procedures above).

Decision policies and model comparisons in the VMR task

Equations 12-16 also apply to the VMR task by substituting t_s, t_m, t_e, t_p with x_s, x_m, x_e, x_p, respectively.

Fitting subjective prior and cost function for re-learning strategy

To better characterize the relearning process after the prior-cost switch, we fitted the behavior after switch using an ideal observer model in which the prior and cost function could change dynamically across trials. This exercise allowed us to examine the time course of relearning and provided a basis for later analyses where we compared the learning time-constants between the prior and cost function. To do so, we assumed participants used a Gaussian prior and a quadratic cost function, and treated the mean of the prior, μ, and shift in the cost function, Δt, as free parameters of the model (in addition to the variance of the prior and the two Weber fractions for measurement and production; Supplementary figure 6) that were fitted based on the behavioral data after the switch. The two Weber fractions as additional parameters did not change the result in Figure 4 as a simpler model without free w_m and w_p led to qualitatively similar results. We estimated the model parameters by maximizing the likelihood of data t_p and t_s (Equation 16 with the decision policy $f_j(t_m)$ from Equation 13; 'fminsearch' function in MATLAB) under the model. We used a running window of 100 trials, which led to 1 pre-transition and 4 post-transition estimates of μ and Δt. We did not fit a subjective model to the VMR data as we could not reject the implicit learning strategy (Figure 7a).

Finally, we compared different models (implicit, explicit, prior-sensitive, cost-sensitive) across all data sets using Bayesian Information Criteria ($BIC=-2 \times L + N_{param} \times log(N_{obs})$ where L is the likelihood of data given model parameters, N_{param} is the number of free parameters in the model, N_{obs} is the number of data points, i.e., trials). For implicit learning, prior-sensitive, and cost-sensitive models, we used the first session's w_m and w_p estimates, which remained relatively stable across sessions (Supplementary figure 11). Therefore, N_{param} was zero for all models except the subjective re-learning model (5 per each data set, 100-trial window).

Analysis of learning time-constants and simulation

The subjective re-learning model allowed us to capture how the internal prior and cost function is updated after the transition. We fitted an exponential function to the time course of the subjective μ and Δt to quantify and compare the corresponding learning time-constants for each participant. To make the fits more robust, we constrained the initial and final values of μ and Δt to match the experimentally imposed values for the pre- and post-transition (μ_{pre}, Δt_{pre} and μ_{post}, Δt_{post}, respectively). With this consideration, we only needed to fit the parameter associated with the learning time-constant (τ_{μ}, $\tau_{\Delta t}$; Figure 5a), as follows.

$$\mu^{i} = (\mu_{pre} - \mu_{post})\exp(\frac{-i}{\tau_{\mu}}) + \mu_{post}$$ (Equation 17)

$$\Delta t^{i} = (\Delta t_{pre} - \Delta t_{post})\exp(\frac{-i}{\tau_{\Delta t}}) + \Delta t_{post}$$ (Equation 18)

Where μ^{i}, Δt^{i} are the prior mean and cost shift in trial *i* after the transition. We estimated τ_{μ}, $\tau_{\Delta t}$ by maximizing the likelihood of data t_p and t_s (Equation 16 with the decision policy $f_j(t_m)$ from Equation 13; ‘fminsearch’ function in MATLAB). Similar to the subjective model fitting, we allowed both the mean and variance of the prior to change throughout learning.

We also performed simulations to examine how the learning time-constants for μ and Δt influence performance (Figure 5f). The procedure for each simulation was as follows: 1) we chose specific values for the learning time-constants, τ_{μ} and $\tau_{\Delta t}$; 2) we computed μ and Δt as a function of time governed by those learning time-constants; 3) we computed the moment by moment value of expected reward (*EC*) as a function of the instantaneous values of μ and Δt as follows:

$$EC^{i} = \int_{t_p}\int_{t_m} C\big(t_p, t_s^{i}\big)p(t_p \mid f_j(t_m), w_p)p\big(t_m \mid t_s^{i}, w_m\big)dt_m dt_p$$ (Equation 19)

In this equation, we computed EC^{i} (expected cost in trial *i*) in the context of the metamer prior-cost pair for an observer with w_m =0.14 and w_p = 0.08, close to averages across subjects (Supplementary Figure 11). In this equation, different values of μ and Δt influence *EC* through the decision policy $f_j(t_m)$ (Equation 13). We also measured the overall average reward (marginalizing over time) for different choices of τ_{μ} and $\tau_{\Delta t}$.

Reinforcement learning models with exploration

As a concrete instantiation of the implicit learning strategy (Supplementary figure 5), we implemented reinforcement learning (RL) models that incrementally build reward expectation based on trial-and-error experiences (i.e., measurement, estimate, and associated reward). We again focused on the RSG task as we could not reject the implicit learning strategy in VMR. In RL terms, representation of the expected reward is called ‘value function’ and in the RSG task, ‘action value’ (Q value, *Q(state,action)*; colormap in Figure 2c) is the critical information that determines the decision policy as follows:

$$f_{RL}(t_m) = \arg\max_{t_e} Q(t_m, t_e)$$ (Equation 20)

Note that, in our setting, state and action in the RL literature correspond to measurement (t_m) and estimate (t_e), respectively, and RL agents have access to the true state (t_s) only with measurement noise (w_m: 0.1577, mean across all participants).

We implemented three RL models that are different mainly in two aspects: i) whether the space for t_m and t_e is discrete or continuous and ii) whether deep neural network was used as a function approximator to compute the action value (critic or value network) and/or the action itself (actor or policy network). The first RL model is Q-learning agent (52), which update action value for a bin of a discrete table as follows:

$$Q^{i+1}\left(t_m^i, t_e^i\right) = Q^i\left(t_m^i, t_e^i\right) + \alpha * \left(R^i + \gamma \max_{t_e'} Q^i\left(t_m^{i+1}, t_e'\right) - Q^i\left(t_m^i, t_e^i\right)\right) \quad \text{(Equation 21)}$$

In this equation, $Q^{i+1}(t^i_m,t^i_e)$ is the updated action value for trial *i+1* for a bin of (t^i_m,t^i_e), α is the learning rate (set to 0.01 for training), R^i is the reward given in trial *i*, γ is the discounting factor (set to 0.001 as t_s was randomly selected from the prior in RSG) for the maximum future value, $maxQ^i(t^{i+1}_m,t'_e)$. We discretized t_m and t_e using 40 and 50 bins, respectively (range: [341 1553] for t_m, [180 2285] for t_e; chosen from actual data of subjects). Note that the update occurred only locally (i.e., (t^i_m,t^i_e)) in the Q table. We trained this tabular Q-learning agent in the original prior-cost condition for 500,000 trials under complete exploration mode to sample reward in full space of t_m and t_e ($\boldsymbol{\varepsilon}$=1 in $\boldsymbol{\varepsilon}$-greedy exploration rule; when exploring, the action t_e from Equation 20 is perturbed by zero-mean Gaussian noise whose SD is w_m*t_e).

The second RL model that we implemented was Deep Q Network (DQN), which has achieved human-level performance in a variety of real-world games through end-to-end training (53). Crucially, DQN obviated the need for discretizing t_m by using a deep neural network to compute Q values with t_m and t_e as an input to the network. As a function approximator, the network also provides a natural means for generalizing over different t_m and t_e values, which could not be easily achieved in the tabular Q agent due to its local update. However, solving Equation 20 to select a reward-maximizing t_e still requires discretization of t_e (50 bins). For the network, we used two fully connected layers (64 units; interspersed with a rectified-linear-unit (ReLU) layer) for each input, t_m and t_e, which were then connected to the output layer ($Q(t_m,t_e)$) through an addition and ReLU layer. We used adam optimizer (3) to train DQN for 1,000,000 trials in the original prior-cost condition (mini-batch size (M in Equation 22 below): 256, learning rate: 0.001, discounting factor (γ): 0.0001, $\boldsymbol{\varepsilon}$=1 for exploration, experience replay buffer size: 100,000). During training, the network weights (8641 parameters in total) was updated to minimize the following mean-squared Bellman error, *L*:

$$L = \frac{1}{M}\sum_{i=1}^{M}\left(R^i + \gamma Q'\left(t_m^{i+1}, \arg\max_{t_e'} Q\left(t_m^{i+1}, t_e'\right)\right) - Q\left(t_m^i, t_e^i\right)\right)^2 \quad \text{(Equation 22)}$$

Where M is the mini-batch size, R^i is the reward given in trial *i*, γ is the discounting factor, *Q'* is action value from the target network (53) whose weights were cloned from the original Q network with smoothing (factor: 0.001), and arg*max* $Q(t^{i+1}_m,t'_e)$ is the best action chosen from the original Q network.

Finally, we also trained Deep Deterministic Policy Gradient (DDPG) network (54) that can work in both continuous spaces of t_m and t_e by using a separate actor network to implement action selection (the greedy policy in Equation 20). The only difference between DDPG and DQN is the actor network, which consists of an input layer for t_m, two fully connected layers (128 units) interspersed with two ReLU layers to generate t_e for output (33409 parameters in total). The actor network was trained to maximize Q value in Equation 20 using gradient ascent. The critic network of DDPG was identical to that of DQN except the fact that the number of units per layer increased to 128 (33665 parameters in total). The weight of critic network was updated similarly to minimize the mean-squared Bellman error (Equation 22) while the best future action (arg*max* $Q(t^{i+1}_m,t'_e)$ in DQN) is computed from the actor network in DDPG. We trained DDPG for 500,000 trials in the original prior-cost condition (mini-batch size: 256, learning rate: 0.001, discounting factor: 0.001, experience replay buffer size: 100,000, default Ornstein-Uhlenbeck noise in MATLAB for exploration).

After confirming that all three RL agents show the near-optimal policy after training for the original prior-cost condition (Supplementary figure 5), we simulated the RL agents for transition to the metamer condition. An important parameters that determine post-transition learning of the RL agents are the learning rate (α) and exploration rate ($\boldsymbol{\varepsilon}$ in $\boldsymbol{\varepsilon}$-greedy). We varied the two parameters in a reasonable range (α: 0.05, 0.005; $\boldsymbol{\varepsilon}$: initially 1, 0.5, 0 with decay rate of 0.05, 0.005) with a goal of matching learning dynamics in RL agents to actual data (Figure 3). Given the stochastic nature of sampling t_m across trials, we repeated the simulation 100 times for each agent with a particular combination of α and $\boldsymbol{\varepsilon}$, and averaged the results across the repetitions. For DQN and DDPG, we reset the experience replay buffer immediately after the transition and save all post-transition trials onward in the buffer. The mini-batch size for learning was set to 1 after the transition and then gradually increased to include all experienced post-transition trials. We tested whether including data from pre-transition trials (~150 trials) in the experience buffer can improve learning. However, DQN and DDPG still could not achieve the optimal level in their policy. All training and simulations were performed using a custom code and RL toolbox in MATLAB (Mathworks, Inc.).

Supplementary Figures

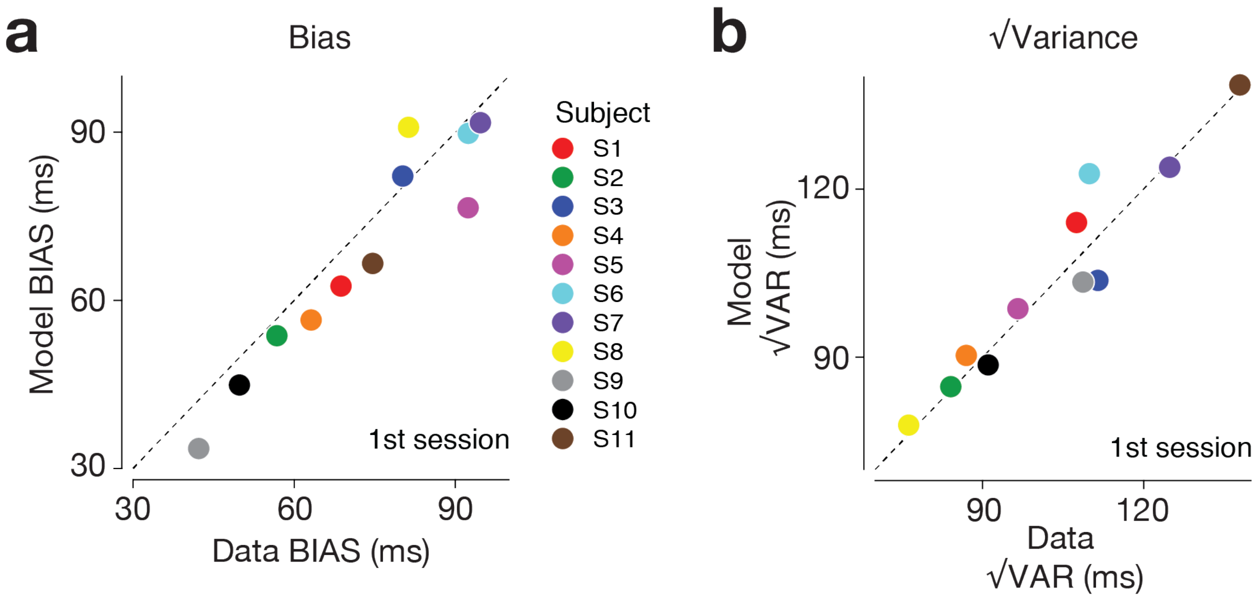

Figure S1. Comparison of ideal observer model to behavior in RSG task in the first session. We compared participants' behavior in RSG task to the prediction of the ideal observer model using two criteria, BIAS (**a**) and √VAR (**b**). These can be formally defined as follows:

$$\mathrm{BIAS} = \sqrt{E\left[bias^2\right]} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} bias_i^2}$$

$$\sqrt{VAR} = \sqrt{E[var]} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} var_i}$$

N is the number of bins that we used to group the continuously sampled t_s (approximately 14). We computed $bias_i$ and var_i (bias and variance of the *i*-th bin) as follows:

$$bias_i = E_i[t_p] - t_s^i$$

$$var_i = E[t_p - E_i[t_p]]$$

where $E_i[t_p]$ is average t_p for a given bin and t_s^i is the average t_s within that bin. We also measured the same statistics for the ideal observer associated with each participant. To do so, we simulated the generative model to perform the same exact experiment (same t_s values) using w_m and w_p fits to each participant. Participants' behavior was consistent with that of the ideal observer model.

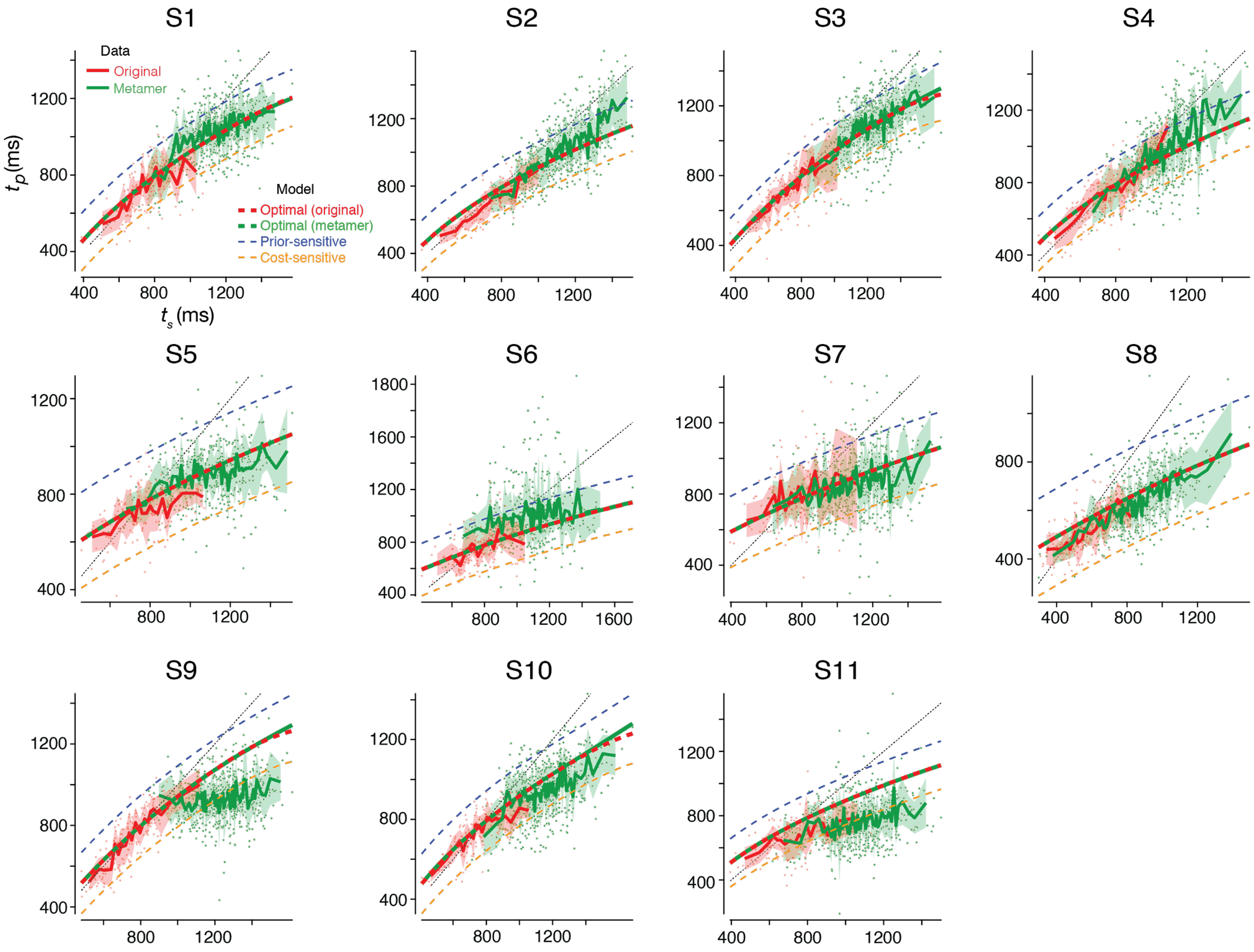

Figure S2. Data from all participants in the RSG task. Results are shown in the same format as in Figure 2d.

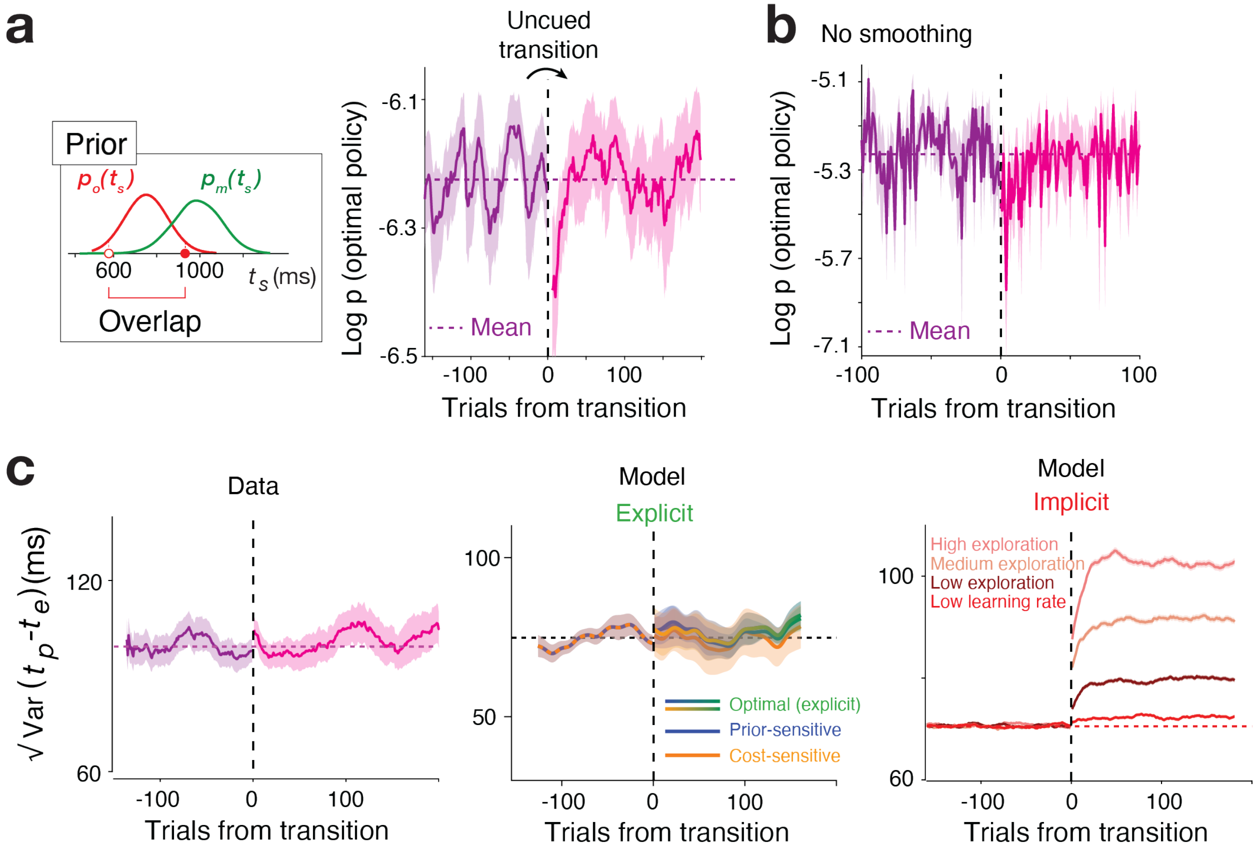

Figure S3. Control analyses for transient suboptimal policy in RSG task.

(**a**) The transient deviation from optimal policy for the subset of stimuli that overlap between the original and metamer priors. One potential reason behind the transient deviation from optimal policy in Figure 3a is that some of the t_s values in the metamer condition were not present in the original condition. To address this concern, we reanalyzed the probability of the optimal policy for the subset of trials whose t_s values overlapped with the range of t_s values before the transition. (left) For the transition from the original to the metamer (second and fourth sessions), we analyzed trials in which t_s was between the minimum (empty red circle) and maximum t_s (filled red circle) tested in the first session. For the transition from the metamer to the original (third and fifth sessions), we analyzed trials in which t_s was between the minimum and maximum t_s tested in the metamer condition of the second session. (right) Same as in Figure 3a for the overlapping values of t_s. Results indicate that participants' decision policy deviated from the optimality even for previously experienced t_s values validating our original conclusion from Figure 3a.

(**b**) Same as in Figure 3a without any smoothing (as in Figure 7a for VMR task).

(**c**) Left: in data, variability of responses (t_p) with respect to the optimal estimate (t_e, equation 3) show little change around the transition. This provides evidence against any implicit learning model (right panel; reinforcement learning model, Q agent; see Method) that increases the response variability to explore new decision policy after transition. Middle: explicit learning models do not show increased response variability after the transition.

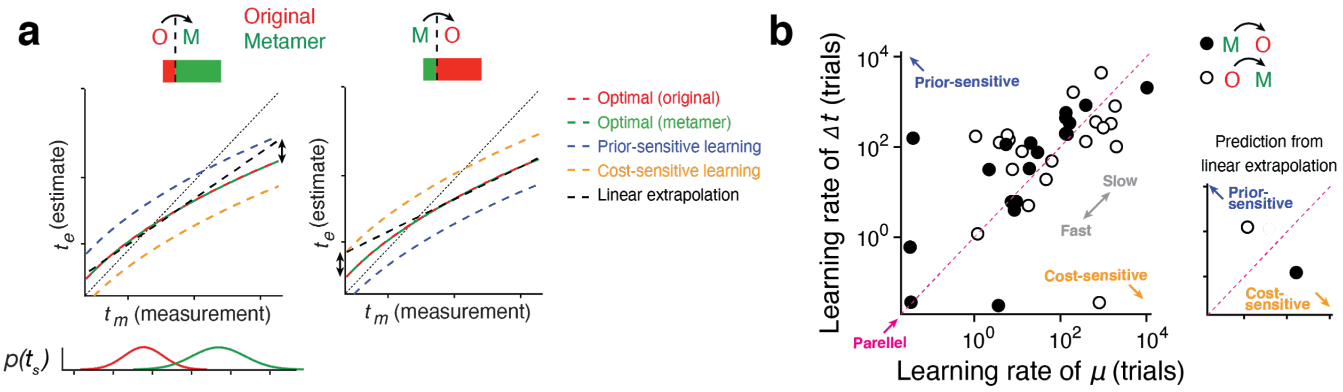

Figure S4. Test of an alternative model with linear extrapolation in RSG.

(**a**) Prediction of the linear-extrapolation model. In RSG, the optimal policy is nonlinear due to the scalar property (Figure 2) and when the prior and cost were switched, subjects might develop a new policy for new t_s values by linearly extrapolating from the previously learned optimal policy. This alternative model makes two specific predictions. First, given the concave shape of the optimal policy (Figure 2), in both original-to-metamer (left) and metamer-to-original (right) transitions, the linear extrapolation (black dashed lines) would lead to overestimation of time intervals. We did not observe the pattern of overestimation in data of individual subjects (Figure 5c-e). Second, the model further makes different predictions depending on the transition types. In the original-to-metamer transition, the linear extrapolation would generate more prior-sensitive-like behavior (blue and black lines in the left), while more cost-sensitive behavior is predicted in the metamer-to-original transition (orange and black lines in the right).

(**b**) Test of prediction from the linear-extrapolation model. We examined the specific prediction about the dependence on transition type using fitted learning rates (right inset). If subjects linearly extrapolated in the transition from the original prior-cost pair to its metamer, it would manifest as faster prior learning (upper left), and vice versa. We do not observe any difference in the learning rates between the two transition types (filled and empty circles; p(signrank)=0.42).

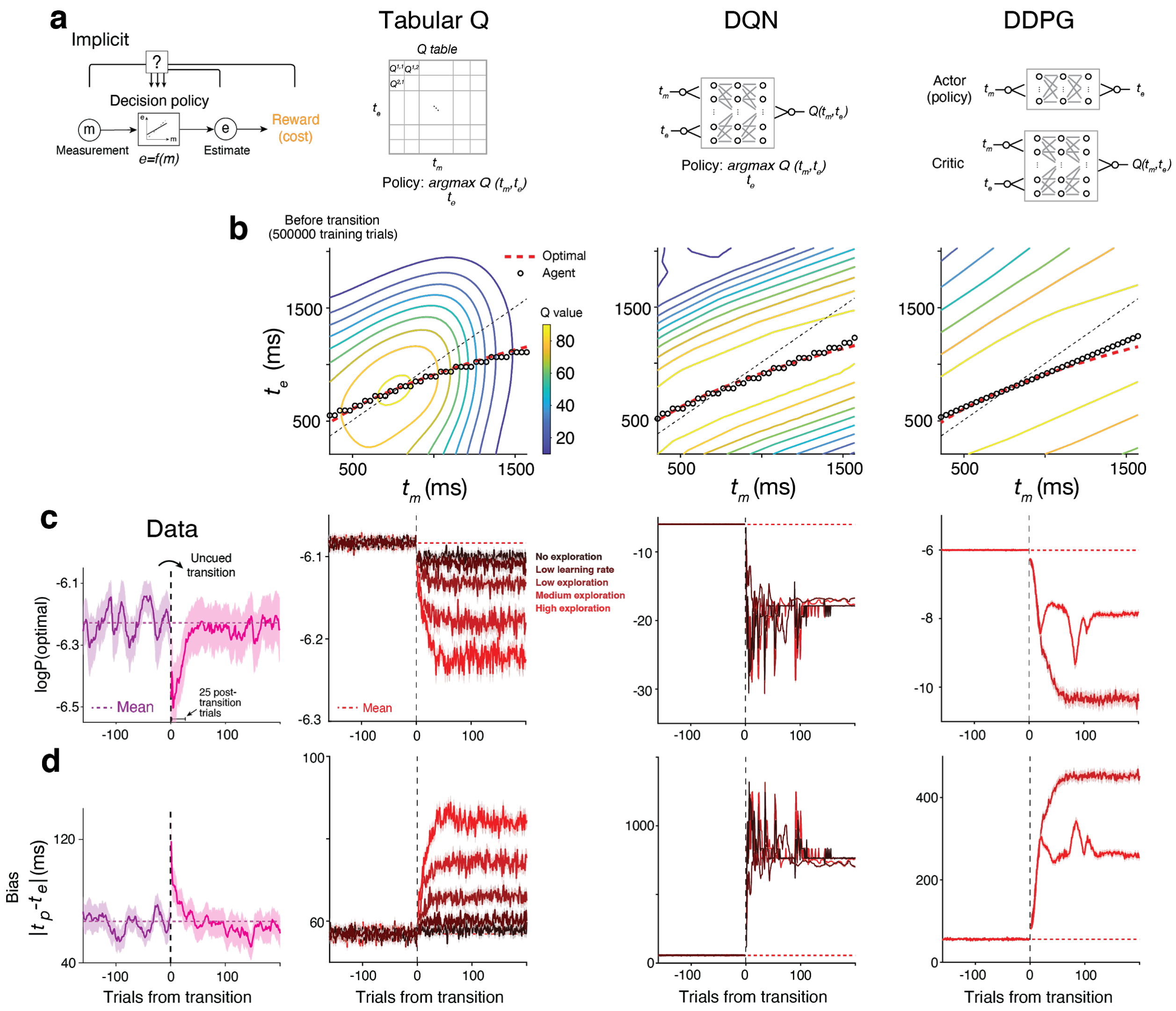

Figure S5. Test of reinforcement learning (RL) models in RSG.

(**a**) RL models as an instantiation of implicit learning strategy. In essence, the implicit models use trial-and-error experience of measurement (*m*), estimate (*e*), and associated reward to construct an optimal decision policy, *e*=f(*m*). In RL models considered here, agents build an internal representation of expected reward as a function of *m* and *e* (‘action value’ or Q(state,action) in more general RL terms) and then choose the best action that maximizes the expected reward given the state (greedy policy in Equation 20). We implemented three RL models that differ in how the continuous space for t_m and t_e in RSG was handled and how deep neural networks were used in the model (see Methods for details). Briefly, tabular Q agent (left) constructs a discrete look-up table for Q(t_m, t_e) that is updated by actual reward given in each trial. In DQN (middle), a multi-layer neural network serves as a function approximator for Q(t_m, t_e) and its weights were trained to minimize the prediction error between actual and expected reward. In DDPG (right), another network (actor or policy network) was used to accommodate the continuous nature of action space (t_e) and directly implemented the decision policy (t_m=f(t_e)).

(**b**) Decision policy of the RL models (left: Tabular Q, middle: DQN, right: DDPG) after training (500,000 trials for the original prior-cost pair) is consistent with the optimal policy (red lines). Corresponding Q values are also shown as color-coded contours.

(**c**) Log probability of the optimal policy, denoted ‘logP(optimal)’ as in Figure 3a (copied here on the left for comparison; left: Tabular Q, middle: DQN, right: DDPG). Different lines denote agents with different exploration rates and learning rates (no exploration: α=0.05, $\boldsymbol{\varepsilon}$=0; low learning rate: α=0.005, $\boldsymbol{\varepsilon}$(trial *i*)=1*(1-0.05)^{i}; low exploration: α=0.05, $\boldsymbol{\varepsilon}$(trial *i*)=0.5*(1-0.05)^{i}; medium exploration: α=0.05, $\boldsymbol{\varepsilon}$(trial *i*)=1*(1-0.05)^{i}; high exploration: α=0.05, $\boldsymbol{\varepsilon}$(trial *i*)=1*(1-0.005)^{i}; see Methods). In DDPG, exploration was not explicitly set up and two different learning rates (α=0.05, 0.005) were tested. Overall, RL models did show the transient suboptimality after transition but did not return to the optimality as fast as observed in human data.

(**d**) same as **c** for bias.

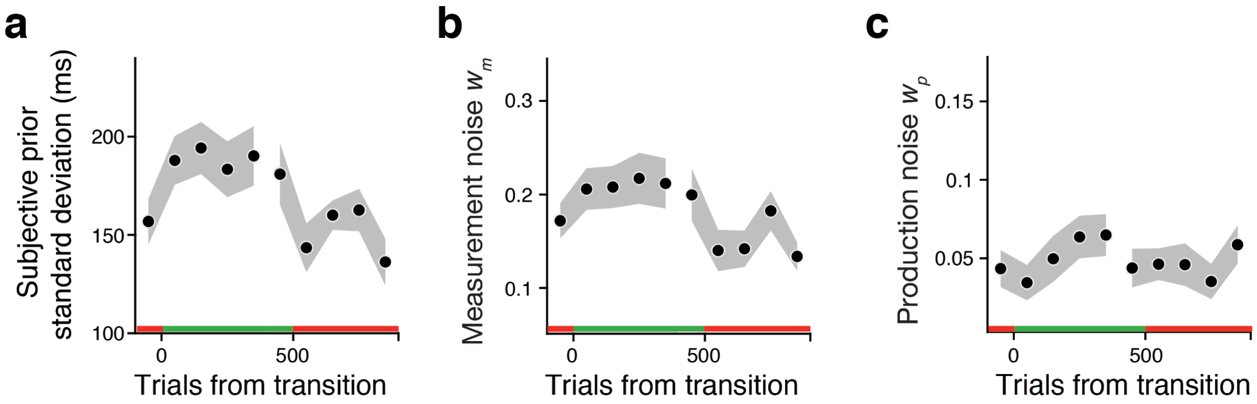

Figure S6. Time course of parameters in subjective prior-cost model. (**a**) Standard deviation of the subjective prior, (**b**), measurement Weber fraction (w_m), and (**c**) production Weber fraction (w_p) were relatively stable during transitions between the original and metamer conditions. Results are shown in the same format as Figure 4a (mean of the prior) and 4b (shift of the cost function). The original prior has SD of 144ms and the SD of the metamer (Gaussian mixture distribution) was 163.85 ± 19.25 (mean ± SD across participants).

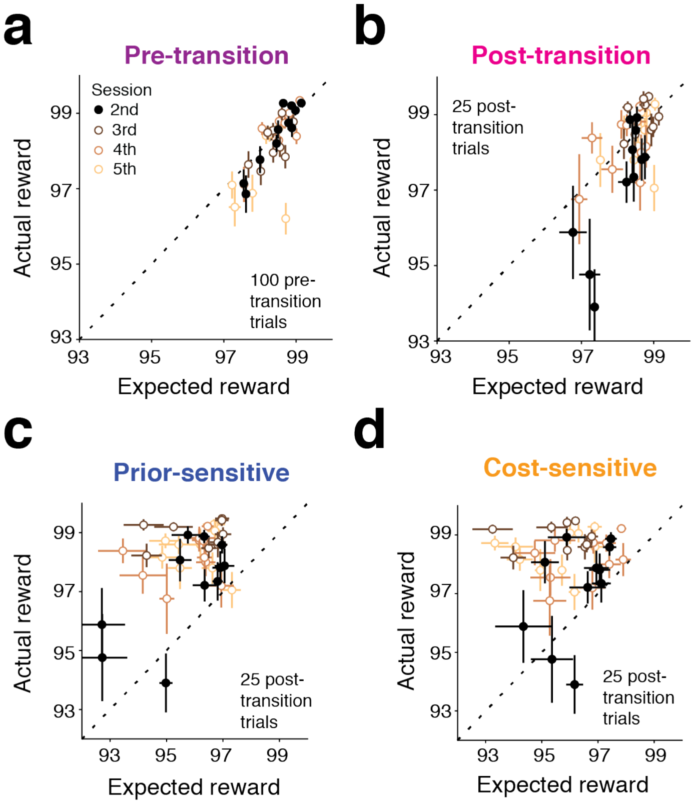

Figure S7. Comparison of observer models and participants for task performance in RSG. We used task performance (i.e., reward score) as another metric to compare participants' behavior to different models (optimal, prior-sensitive, and cost-sensitive). We compared the actual reward each participant received to the expected reward (Equation 19 in Method) under each model (Equation 12-15 in Method) using the Weber fractions derived from fits of the ideal observer to that participants' behavior in the 1st session. (**a**) Before the transition, actual reward was similar to the expected reward under the ideal observer model (circles: individual participants and sessions, error bar: SEM across 100 trials). (**b**) After the transition, the actual reward remained near the optimal level (p=0.14, one-tailed sign-rank test), although participants performed significantly worse than the ideal observer for the 2nd session (p=0.012, one-tailed sign-rank test; black circles), in which they experienced the transition for the first time. We chose 25 post-transition trials for this analysis based on the time course of p(optimal policy) in Figure 3a. (**c**,**d**) Subjects showed better performance than expected from either the prior-sensitive model (p<0.001, one-tailed sign-rank test) and the cost-sensitive model (p<0.001, one-tailed sign-rank test), consistent with findings in Figure 3b,c.

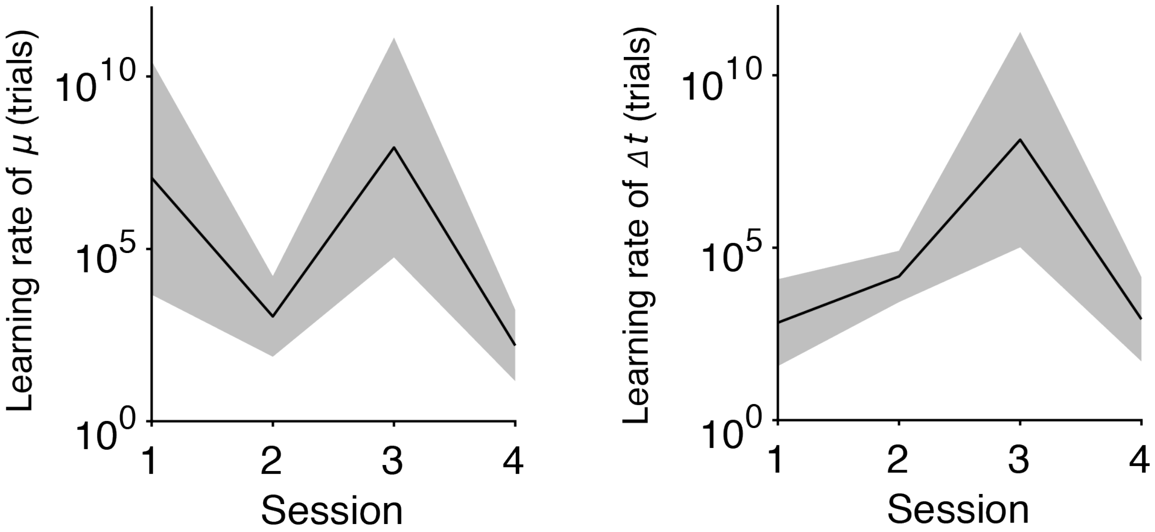

Figure S8. Test of meta-learning using fitted learning rates in RSG.

As our experiment involved repeated experience of the transition across 4 sessions, it is possible that participants either became aware of the transition or learned faster the new prior and cost function. This ‘meta-learning’ or ‘saving’ predicts that the learning rate we inferred from data would decrease as a function of sessions. We did not observe any systematic pattern for learning rates of the prior (left) and cost function (right).

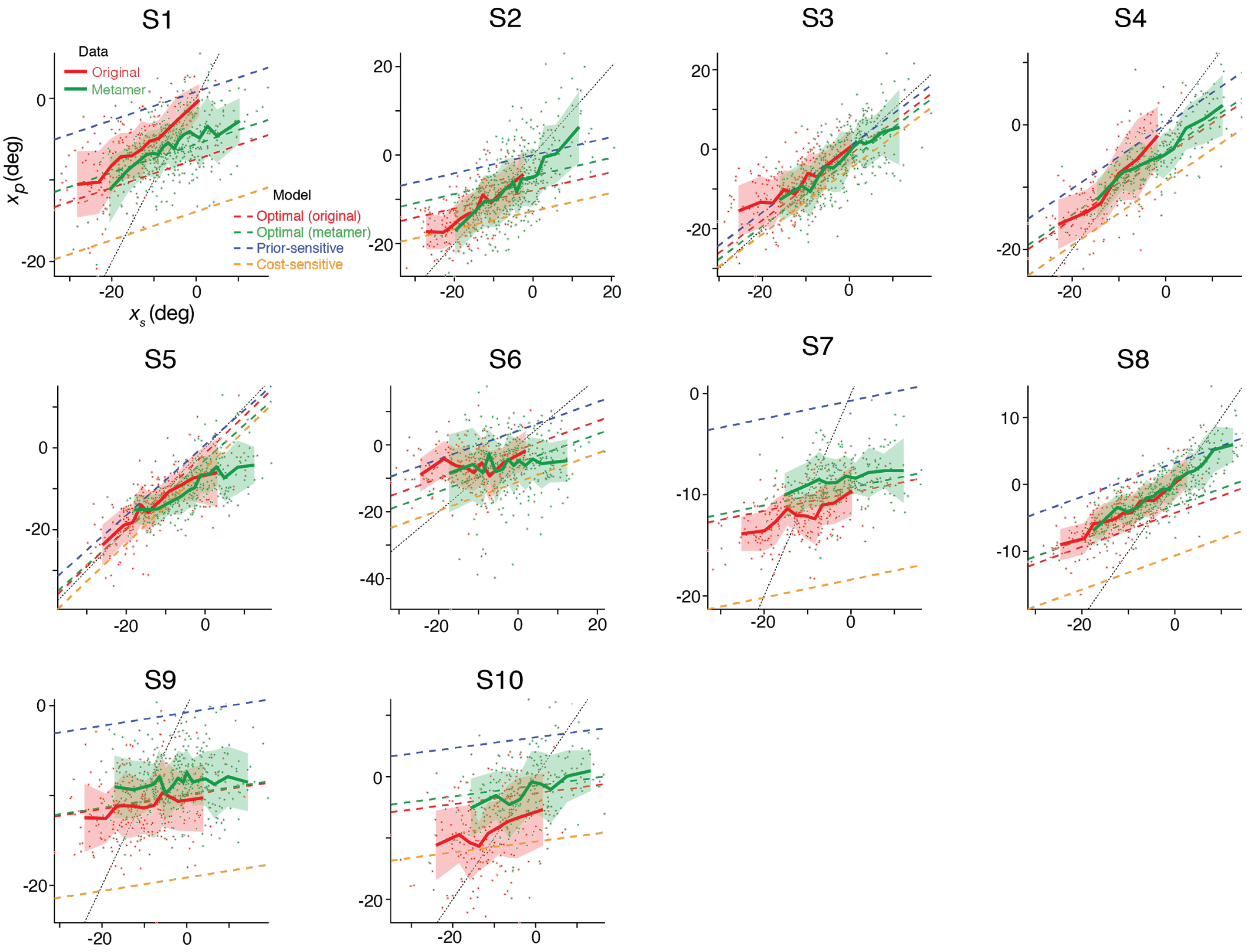

Figure S9. Data from all participants in the VMR task. Results are shown in the same format as in Figure 6d.

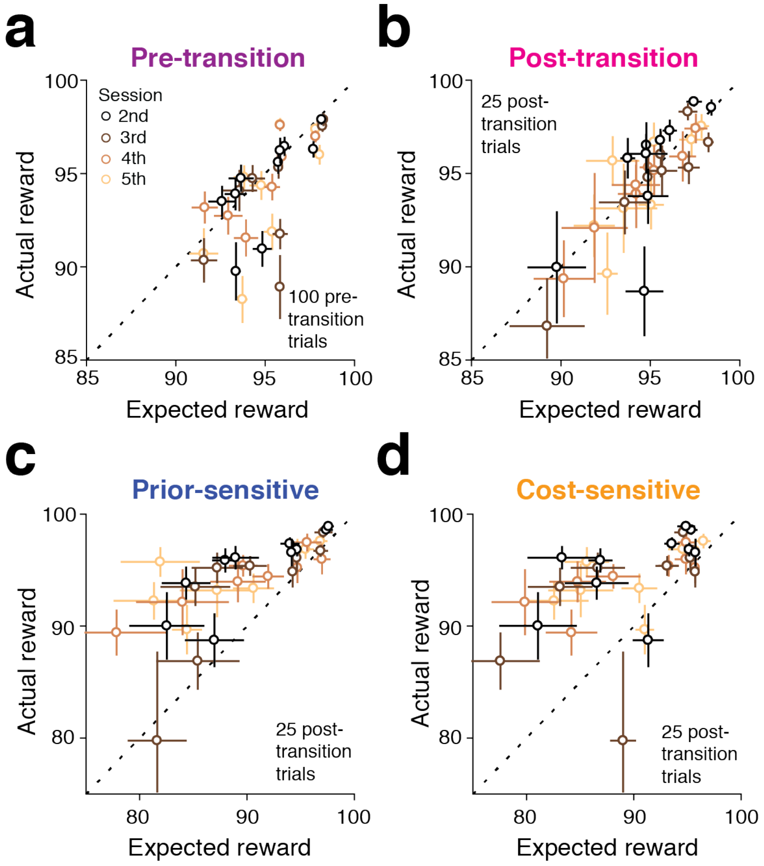

Figure S10. Comparison of observer models and participants for task performance in VMR in the same format as in Figure S5. We compared the actual reward each participant received to the expected reward (Equation 19 in Method) under each model (Equation 12-15 in Method) using the measurement and production noise parameters derived from the fits of the ideal observer to that participants' behavior in the 1st session. (**a**) Before the transition, actual reward was similar to the expected reward under the ideal observer model. (**b**) After the transition, task performance overall remained near the optimal level (p=0.061, one-tailed sign-rank test). Subjects showed better performance than expected from either the prior-sensitive model (p<0.001, one-tailed sign-rank test) and the cost-sensitive model (p<0.001, one-tailed sign-rank test), consistent with findings in Figure 7b,c.

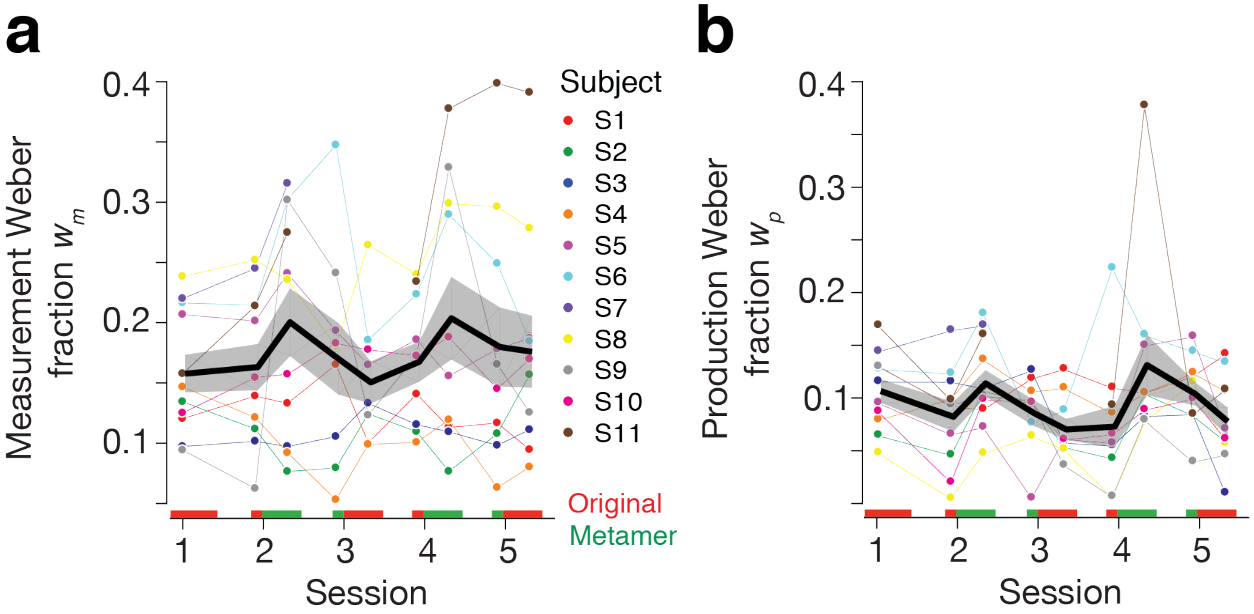

Figure S11. Time course of Weber fractions in RSG. Weber fractions were fitted to pre- and post-transition epochs (shown over the abscissa) separately.

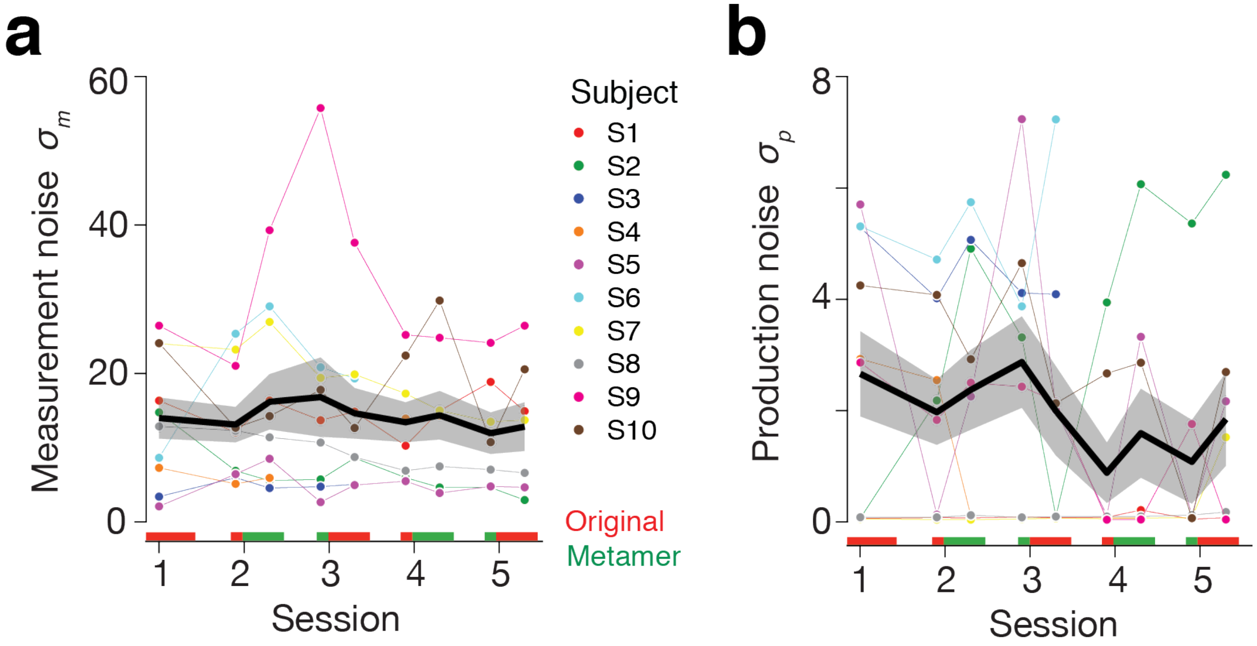

Figure S12. Time course of measurement and production noises in VMR. σ_m and σ_p were fitted to pre- and post-transition epochs (shown over the abscissa) separately.

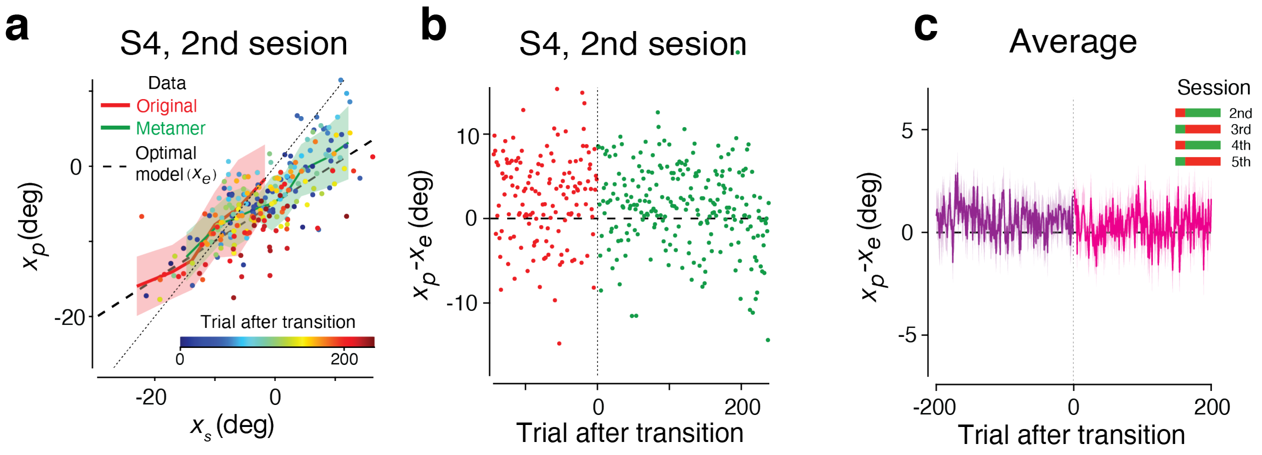

Figure S13. Raw data and its time course in VMR task.

(**a**) Representative data from a participant. Dots show individual trials after the transition (color-coded). Average across trials is also shown (lines for the mean, shades for SEM; red for original, green for metamer), together with the optimal model (dashed line).

(**b**) Time course of the relative bias for the same data set in **a**.

(**c**) same as **b** for average across all participants and sessions (N=36).

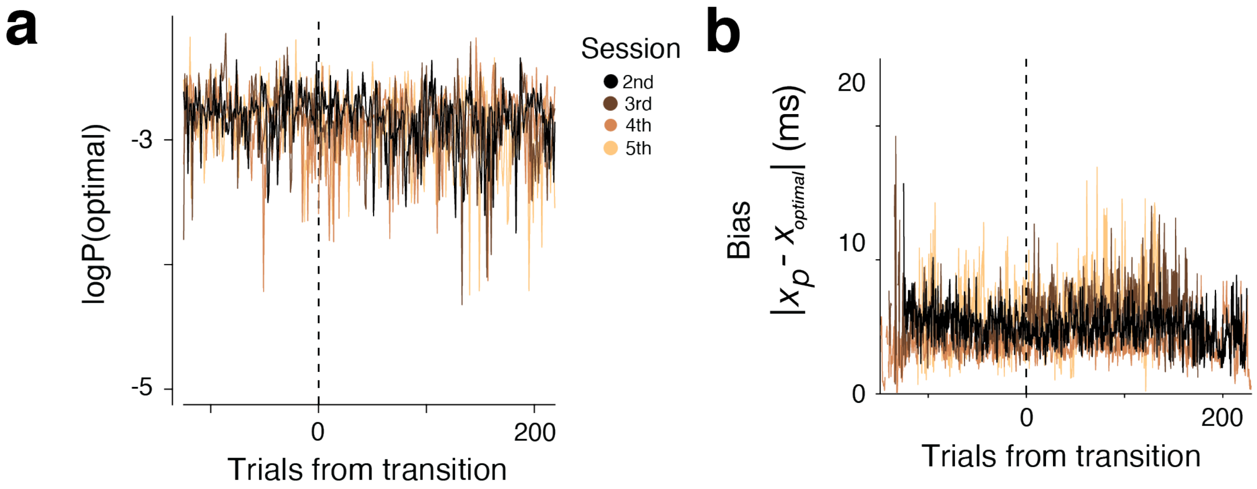

Figure S14. Time course of optimality and bias for individual sessions in VMR as in Figure 7a.

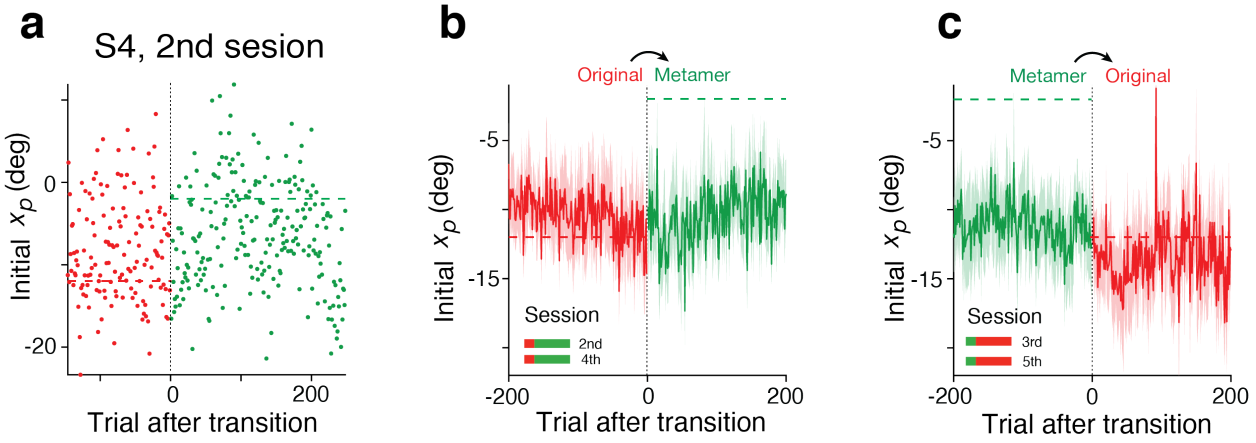

Figure S15. Analysis of initial hand angle in VMR.

(**a**) Representative data from a participant. Red and green dots denote individual trials from the original and metamer condition, respectively. Horizontal dashed lines indicate the mean of the prior distribution for x_s (red: original, green: metamer).

(**b**) Average across all participants for original-to-metamer transition (horizontal dashed lines for the mean of the priors). No smoothing was applied.

(**c**) same as **b** for the metamer-to-original transition.

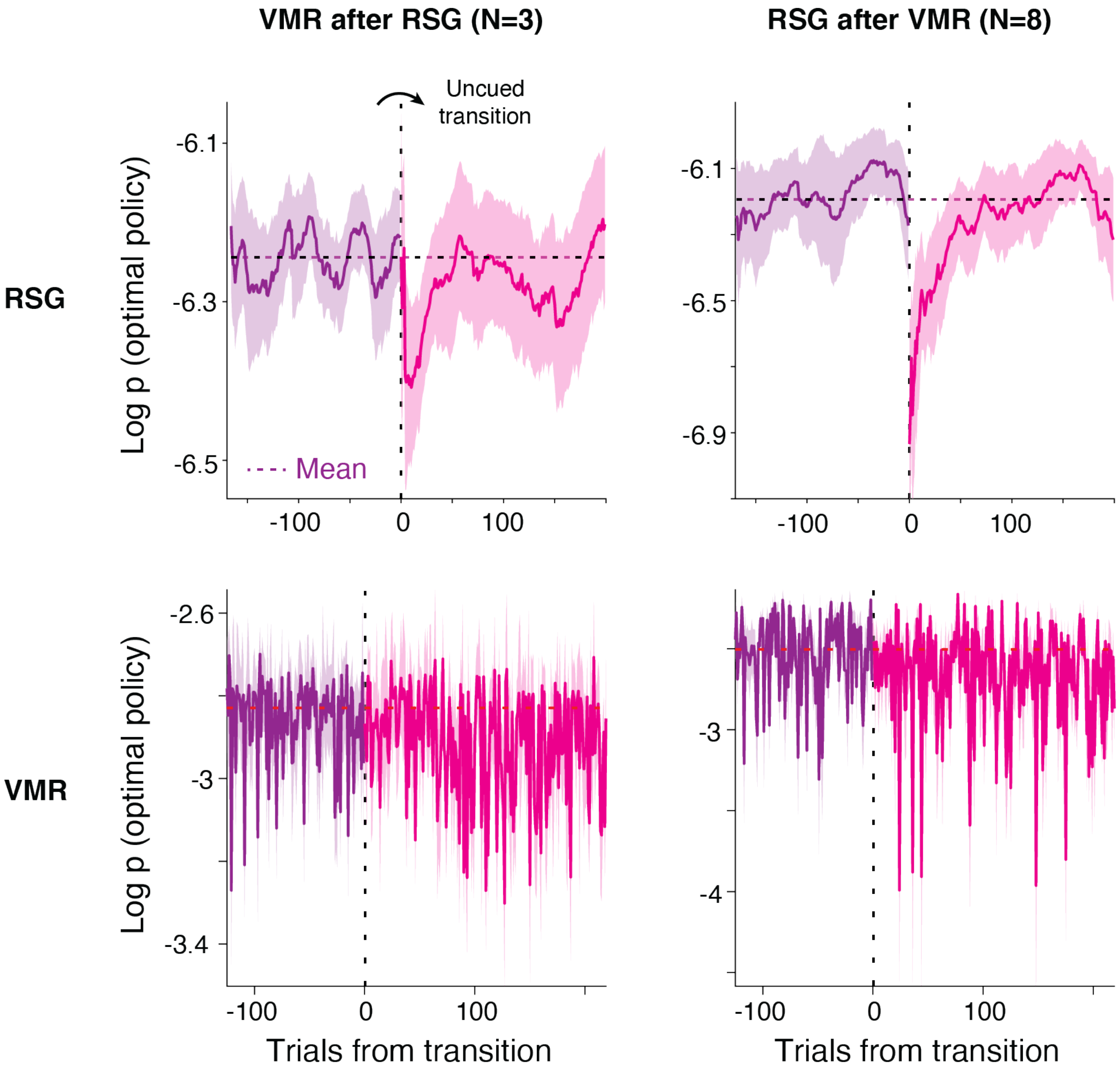

Figure S16. Comparison of subjects with different task sequences in terms of transient suboptimal policy.

We tested whether the sequence of RSG and VMR tasks participants performed affected the transient deviation from optimality, potentially by making participants aware of the transition. In our experiment, we attempted to counterbalance the sequence across subjects: Three subjects performed the RSG task first (left column) and the remaining 8 subjects performed the VMR task first (right column; the proportion was not even due to experimental schedules). Regardless of the task sequence, we still observe the transient deviation from optimality in RSG, but not in VMR.

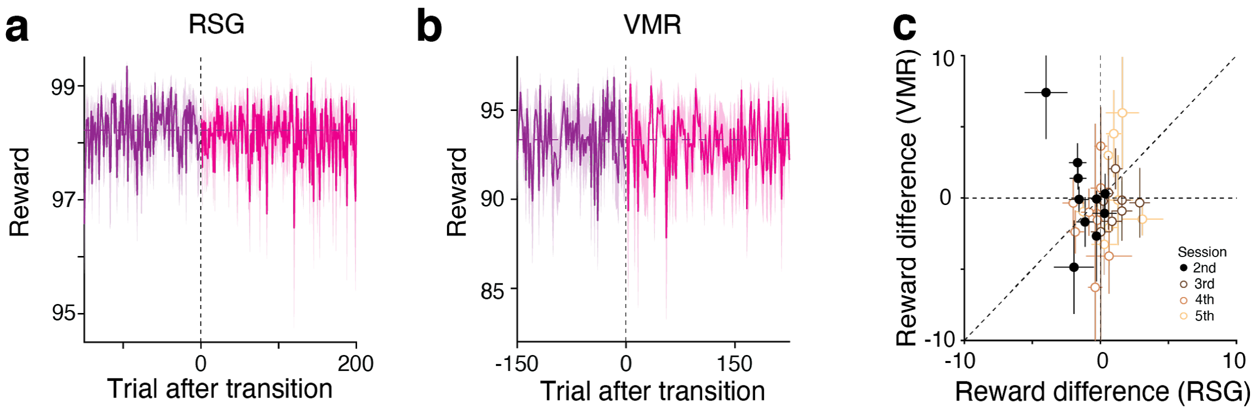

Figure S17. Time course of reward in both RSG and VMR tasks.

Any noticeable drop in reward can initiate exploratory behavior that can be misinterpreted as the model-based explicit learning. Although we matched overall expected reward between the original and metamer conditions by carefully designing the cost function (see Methods), the actual reward subjects experienced can be different. However, we did not observe a noticeable drop in actual rewards across the transition in both RSG (**a**) and VMR (**b**) tasks. Furthermore, when we computed the reward difference between 25 pre- and post-transition trials (**c**), the amount of reward difference was not significantly different between RSG and VMR (p(signed rank)=0.334). This result indicates that the reward statistics cannot contribute to the different results across RSG and VMR (i.e., transient sub-optimality in RSG, not in VMR).

References

1. C. Malapani, S. Fairhurst, Scalar Timing in Animals and Humans. *Learn. Motiv.* **33**, 156–176 (2002). 22

2. M. Jazayeri, M. N. Shadlen, Temporal context calibrates interval timing. *Nat. Neurosci.* **13**, 1020–1026 (2010).

3. D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2014).