Supplementary Information for "Audio cues enhance mirroring of arm motion when visual cues are scarce"

Edward D. Lee, Edward Esposito, Itai Cohen
Department of Physics, 142 Sciences Dr, Cornell University, Ithaca NY 14853

Appendix S1: Experimental protocol

All subjects were informed about the purpose and goal of the study at the beginning of the experiment and gave consent. After a preliminary survey about experience in sports or performing arts and questions about any conditions that would exclude them from the study (including vision, hearing, and arm motion problems and history of poor experience with virtual reality headsets), they were shown how to use the motion capture suit and virtual reality headset comfortably. The subject was familiarized with the mirror game outside of the virtual reality environment through two quick practice rounds (one hand at a time) with the researcher. Subjects were then instructed to "mirror [simultaneously] the motion, or velocity, of the avatar" where the word "simultaneously" was included in the training conditions because it was unclear if all subjects understood what was implied by mirroring in the untrained condition. When audio cues were used, they were also told, "Try to use the sound to predict the motion of the avatar's hand." Immediately previous to the start of the mirroring task, they were reminded visually by a floating script to "Mirror the hand." Periodically throughout the trial, the comfort of subjects in the virtual environment was assessed verbally. At the end of the experiment, all subjects filled out a post-experiment survey to assess the comfort of the suit and virtual headset, importance of fatigue, clarity of instructions, and to check if they had been following instructions.

A sequence of trials for a single hand consisted of 16 different 30 s trials where the first and last trials were always a fully visible condition. During the experiments, the task was paused every 2–3 minutes to assess the subject for any poor reactions to the virtual environment and to ask explicitly about fatigue. If the subject expressed any sign of fatigue, a rest of time of at least 15 s was taken.

We tested four different experimental conditions including no training and no audio (Visual Only), no training with audio (Audio), training without audio (Train), and training with audio (Train+Audio) each with subject sample size N and unique subject and hand combinations M: $(N = 10, M = 17)$, $(N = 10, M = 10)$, $(N = 7, M = 13)$ and $(N = 8, M = 15)$, respectively. Nearly all subjects participated in two experiments, one with each hand and the first hand chosen randomly. The exceptions were when coding bugs prevented us from continuing the experiment.

For the Train and Train+Audio conditions, subjects were told that the first 5 minutes of the experiment would consist of a practice round with a single break in the middle. During the break, subjects were asked if they had any questions about their performance. When audio cues were used, the experimenter emphasized the instruction to use the audio cue and asked the subjects to explain how they were using the audio cues. If they made incorrect inferences about how the audio corresponded to the motion—for example, one subject thought the volume of the audio changed with the location of the avatar's hand—the experimenter explained to them how they were incorrect. To all subjects, the experimenter explained that the audio cue had pitch proportional to the speed of the avatar and became higher in pitch when the avatar was moving faster and lower when the avatar was slowing down or changing directions.

We collected data from 35 participants, but one subject was excluded from the analysis because of professed disinterest in the experiment and cursory completion of the post-experimental survey that included answering an inapplicable question without any mention or question to the experimentalist. All subjects were assigned to one experimental condition per visit. Subjects ranged in ages from 18–42 with varying levels of experience in physical activities requiring coordination with others. Experimental protocol was approved by the institution's IRB and the HRPO at the DoD.

The motion of the avatar was generated by the experimenter with the goal of keeping it aperiodic and within velocity bounds that would be well tracked by the PN motion capture suit. In Fig. S1, we show the autocorrelation function (ACF) of the avatar's motion and the distribution of velocities. The autocorrelation function

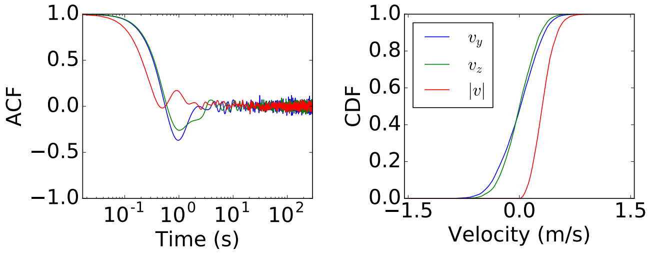

FIG. S1. Statistics on the avatar's motion. (left) Autocorrelation function (ACF) of the velocity along the y and z axes on which we assessed performance and the norm velocity $|v|$. There is little structure in the velocities after the 1 s time scale because the motion is aperiodic. (right) Cumulative distribution function (CDF) of the velocities. The velocities are relatively slow and nearly all within a speed of 1 m/s.

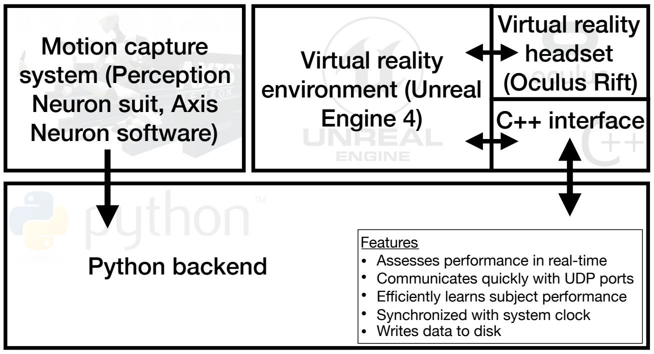
Motion capture system (Perception Neuron suit, Axis Neuron software)
Virtual reality environment (Unreal Engine 4)
Virtual reality headset (Oculus Rift)
C++ interface
Python backend
Features
• Assesses performance in real-time
• Communicates quickly with UDP ports
• Efficiently learns subject performance
• Synchronized with system clock
• Writes data to disk

FIG. S2. Architecture of the experimental apparatus. The Python backend collects data from the Perception Neuron (PN) motion capture system and compares it with prerecorded motion of the avatar that is displayed in the Oculus Rift virtual reality headset in the environment designed using Unreal Engine 4 (UE4). Subject performance is then assessed and the result used to train the learning algorithm that determines the next set of visibility parameters. Those parameters are communicated back to UE4 for the next 30 s trial.

shows small periodicities at the 1 Hz time scale but otherwise little other periodicity at longer time scales. The CDFs show that the velocity of the avatar was limited to a small regime bounded by 1 m/s.

Appendix S2: Experimental apparatus architecture

To run the experiment, we combine several commercially available or open source platforms to run the virtual environment, capture the motion of the subject, and train the online learning algorithm. We discuss how these are combined in an overview and then discuss details of the platforms in more detail.

The main components of the system are detailed in Fig. S2. The apparatus involves running a virtual reality environment on Unreal Engine 4 (UE4), a game development engine. Subjects are immersed in the environment with the Oculus Rift virtual reality headset. We capture their motion using the Perception Neuron motion capture suit. We compare the subject's motion with the prerecorded avatar's on a Python backend and learn the subject's performance landscape the results of which are sent to UE4 to determine the course of the experiment.

UE4 is a standard game development engine used to develop applications for virtual reality environments built on a C++ backend [1]. Since it is widely used, many plugins and features are ready to use, and the Oculus Rift requires no further programming to interact with the three-dimensional environment that we build. We use the environment to display visual instructions to the subjects, manipulate the visual appearance of the avatar, play the audio cue, and provide feedback to the subjects on their performance in the form a green "health bar" above the avatar's head. This environment is displayed in the Oculus Rift virtual reality headset that was originally designed for gaming and is available on the consumer market. It provides a 3-dimensional perspective through two lenses that refresh the visual field at 90 Hz. Although each eye has a high definition 1080p view of the world, the width of the field of view means that pixelation is visible, if not conspicuous.

To check that this environment was adequate for controlling the visual appearance of the avatar during our experiments, we verified that the internal loop controlling whether or not the avatar was visible was accurate to the tens of milliseconds level. We did this by recording the system clock time every iteration of the loop found it to be accurate within tens of milliseconds. Instead, the limiting factor in how low we can reduce the shortest visual gap or visual appearance of the avatar is the refresh rate of the headset. This pins us at a lower limit of about 0.1 s which is close to the minimum for human reaction time.

Perception Neuron (PN) is a motion capture suit developed by Noitom. Instead of relying on optical marker tracking, PN is based on a network of inertial measurement units (IMUs) that measure local acceleration and angular velocities. This is a relatively new technology because drift error can become a serious problem for systems not tethered to a fixed coordinate system.[1] Nevertheless, it is the case that in recent years IMU-based systems have made notable advances and easily portable, energy efficient, and significantly cheaper than most optical marker tracking systems.

PN comes with software that rapidly (within a delay of 15 ms) computes and transmits via port the motion of the subject including position, velocity, acceleration, and rotation angles [2]. However, these measurements are processed by a custom algorithm based on proprietary technology and the raw acceleration and orientation data from the suit sensors are inaccessible. Although we cannot inspect the algorithm in detail, we note that the widely-used algorithm used to calculate lower order moments of motion (velocity and position) are almost always variations on the Kalman filter [3]. Typically, results are more biased by particular assumptions of the algorithm the higher the order of the integral or derivative one takes from the data, so we focus on measuring the velocities of the subjects and do not consider positions or orientations of the body.

For our analysis, it is important that the total latency in our system be below 100 ms which is the lower limit to human reaction time. Across a few tests, we find that the PN system compares remarkably well to other well-tested equipment systems and latency errors are easily below 100 ms.

[1] Drift error refers to the fact that the measurement components cannot measure directly the position or velocity of the IMUs directly, but that they must be calculated from integration of noisy measurements.

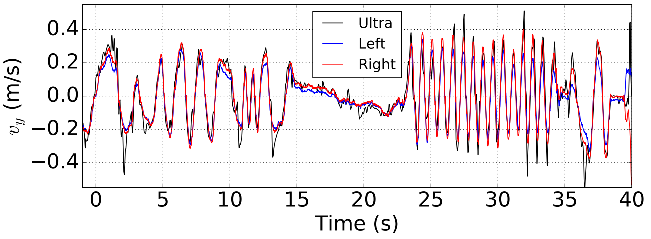

FIG. S3. Comparison of velocity with an ultrasound distance meter. A box was held between the two hands of the experiment and moved towards and away from the detector. We show the resulting estimates of the velocities from the detector (black), the Perception Neuron left hand (blue), and the right hand (red). Zero crossings in the velocities agree within 50 ms, below the lower limit of our analysis in performance (Fig. 2).

First, we compare the PN suit with a known standard and well-tested Vicon optical marker tracking system in a local facility. This system provides a different way of measuring the motion of the subject because it tracks the location of each of the markers which can be used to calculate the velocities instead of the acceleration. When properly calibrated, the Vicon system can measure the position of its markers down to millimeter precision and with a latency of single milliseconds. We find that on a computer system with sufficient processing power (otherwise a significant time varying delay is incurred) and when the PN suit is physically connected to the computer, latency is well below 50 ms as advertised. The values of the velocities do not agree with those estimated from the Vicon system but they constitute roughly a scaled transformation such that velocities < 1 m/s like those encountered in the avatar's motion do not incur more than 10% error in the conditions we explored. Reassurringly, the zero velocity crossings match almost exactly in the two systems. Given the high accuracy and precision in timing of direction changes but relatively significant errors in the magnitudes, we do not consider directly the magnitudes of the velocities in our analysis.[2]

We furthermore compare the timing of the suit with an ultrasound Vernier Motion Detector. The ultrasound distance meter is reportedly accurate to a single millimeter with a maximum recording frequency of 30 Hz [4]. To compare the PN suit with the detector, we held a box between both hands and moved it towards and away from the detector and measured the velocities along this axis of motion as shown in Fig. S3. By interpolating the measured velocities to estimate the zero crossings, we find that disagreements were below 50 ms. Again, we found nothing to suggest that the latency of the PN suit was large enough to affect our results when estimating the velocities. Indeed, we find close agreement between the two systems and the timing of direction changes is precise within few tens of milliseconds.

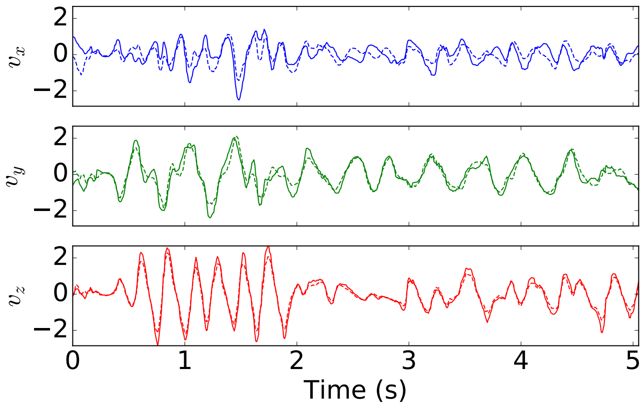

FIG. S4. Example velocity trajectories as measured using the Perception Neuron suit when the subjects hands are clasped together. Errors in the inferred orientation of the hands lead to larger relative errors in the x direction which corresponds to the axis pointing from the subject to the avatar (Fig. 1A). We ignore this axis for our analysis.

Finally, we tested the suits by fixing the hands together checking for consistency between the two velocity trajectories, an example of which is shown in Fig. S4. Here, we found that rotational errors in the orientation of the arms would leads to differences in timing and velocity along the x axis (pointing from the subject to the avatar). The other axes y and z seemed to be less affected by this problem. For our analysis, we do not consider the x direction.

Appendix S3: Dynamic time warping (DTW)

Although spectral techniques provide one way to compare motion in coordination tasks (including variants of cross correlation, wavelet analysis, and recurrence plots), we use DTW to align the velocity trajectories of the subject with that of the avatar [5–7]. One major issue with using spectral techniques to identify temporal delays in aperiodic motion is ambiguity in deciding which local peak in the time-lagged cross-correlation corresponds to the time delay especially when individuals are failing to mirror the partner well. DTW, on the other hand, finds the globally optimal alignment and thus can use global information to resolve these ambiguities in local alignment. Overall, DTW is a computationally efficient way of accounting for strong local nonlinearities in multiple dimensions when comparing motion trajectories.

The goal of DTW is to align two curves by allowing lo-

[2] For the alignment of the subject's motion with the avatar's, we use the magnitude of the measured velocities but we design our cost function to rely on linear differences between the velocities to minimize the effects of scale. In principle, we could also account for such scaling errors by introducing a scaling error parameter, but we did not find this necessary.

cal temporal stretching. This is accomplished by a combinatorial algorithm that involves finding the optimal path in the matrix defined by Eq S3.1

$$D_{\mathrm{ij}} = |\vec{v}_{\mathrm{s}}(t_{\mathrm{i}}) - \vec{v}_{\mathrm{a}}(t_{\mathrm{j}})| + \lambda g(\mathrm{i},\mathrm{j}) \tag{S3.1}$$

The corresponding path minimizes the total accumulated distance between the warped curves with some extra cost g and strength of regularization λ for disfavoring unrealistic trajectories. The resulting path defines a warped time $\tilde{t}_{\mathrm{i}}$ that gives a measure of the local time delay (or anticipation) that the subject shows while tracking the avatar.

The first term in Eq S3.1, what we call the cost, is often quadratic in the distance. For our system, the linear distance between the two velocities is essential because of both human motion and limitations of the experimental apparatus that we are using to capture motion. Some subjects change directions very rapidly to correct for errors in direction and this results in large velocities with temporal profiles that are almost correct while velocity magnitude deviations can be large. With a cost function that grows superlinearly with the velocity, error peaks would be aligned even at the expense of many features smaller in magnitude but indicative of mirroring. Furthermore, we have found that the motion capture system can overestimate absolute velocities especially when the acceleration is large. Thus, peaks in velocities are especially prone to systematic error. In both these cases, a superlinear distance measure between the velocities would favor weight large peak matching instead of aligning the many smaller features of trajectories, and so we rely on a cost linear with distance between the trajectories.

As for second term in Eq S3.1, the regularization, we design g to avoid situations in which the subject is impossibly anticipating the motion of the avatar (as can happen when motion seems briefly periodic) and when the inferred delay is so large that subjects would have to remember far into the past while memorizing new motion simultaneously. To design a sensible regularization function in Eq S3.1, we find that when $\lambda = 0$ DTW will find some trajectories where the subject is leading the avatar by seconds or is behind the avatar by seconds. These trajectories tend to appear in cases where the subject is doing very poorly and so it is difficult to find a temporally local trajectory that resembles the avatar's motion. They also occur where brief periodicities mean a phase shift of 2π overlays the trajectories. Noting that when the avatar is fully visible, subjects infrequently venture outside a time delay of $3/2\,\mathrm{s}$ or are ahead by more than $1/2\,\mathrm{s}$, we define g to be zero within the interval $\Delta t \in [-1/2\,\mathrm{s}, 3/2\,\mathrm{s}]$ and then sharply increasing outside of that range.

$$g(\mathrm{i},\mathrm{j}) = \left\{ \begin{array}{cc} 0, & |t_{\mathrm{i}} - t_{\mathrm{j}} + 1/2| < 1 \\ |t_{\mathrm{i}} - t_{\mathrm{j}} + 1/2|^6, & |t_{\mathrm{i}} - t_{\mathrm{j}} + 1/2| \geq 1 \end{array} \right. \tag{S3.2}$$

with $\lambda = 10^{-3}$ controlling the strength of regularization.

To calculate alignment, we first use FastDTW which can calculate the time warp in nearly linear time instead of quadratic time [8]. If the found trajectory ventures outside of the bounding interval $\Delta t \in [-1/2\,\mathrm{s}, 3/2\,\mathrm{s}]$, we then solve the problem using our own (slower) implementation including the regularization. We find that about 60% of the untrained trials were regularized whereas only 35% of the trained trials were. We might expect this difference because untrained individuals typically do not replicate the trajectory of the avatar as well and the algorithm is more prone to misaligning stretches of motion.

Appendix S4: Velocity error thresholds

In the main text, we only consider the temporal delays ϵ^* to characterize the performance of the subjects. Here, we explore the effect of a threshold in the alignment of the velocities ϵ_v^*. In agreement with our results when only considering the time delay threshold as shown in Fig. 2, we find that Visual Only performance is much worse when compared to the other conditions, there is a range of timescales from about 200–800 ms where the largest variation in performance between conditions appear, and that beyond those limits performance variation is small. We also find that audio cues have a larger effect on performance for the shortest time scales, in contrast with training where performance is worse at faster time scales. Overall, inclusion of the velocity error threshold reaffirms our results about the change in mean performance in the main text where we only consider the time delay threshold.

To measure velocity error, we focus only on normalized velocities. We ignore the speed because the size of the avatar does not scale with the size of subject and because we find that the PN suit system is prone to scaling errors with velocity estimation (See SI Section S2). To compare the velocity directions, we define the error to be

$$\epsilon_v\left(\tilde{t}\right) = \frac{1}{2} - \frac{1}{2}\frac{\vec{v}_{\mathrm{a}}\left(\tilde{t}\right)\cdot\vec{v}_{\mathrm{s}}\left(\tilde{t}\right)}{\left|\vec{v}_{\mathrm{a}}\left(\tilde{t}\right)\right|\left|\vec{v}_{\mathrm{s}}\left(\tilde{t}\right)\right|} \tag{S4.1}$$

that is 1 when the velocities are anti-aligned, 1/2 when they are orthogonal, and 0 when they are exactly parallel. As with the timing delays, we choose a threshold ϵ_v^* and measure when subjects are below the fixed threshold. Now instead of a single threshold, we have two thresholds in both velocity ϵ_v^* and timing ϵ^*,

$$\hat{\pi}(\tau, f) = \frac{1}{\tilde{T}+2}\left(1 + \sum_{\tilde{t}}^{\tilde{T}} \Theta\left[\epsilon^* - \left|\epsilon\left(\tilde{t}\right)\right|\right] \times \Theta\left[\epsilon_v^* - \left|\epsilon_v\left(\tilde{t}\right)\right|\right]\right). \tag{S4.2}$$

We calculate the mean performance $\langle\pi_{\mathrm{i}}\rangle_{\epsilon^*}$ over the average per subject as we change the thresholds. To do this, we infer the entire predicted landscape given the data points from Eq S4.2 for every combination of ϵ^* and ϵ_v^* of interest. We summarize the results of the predicted

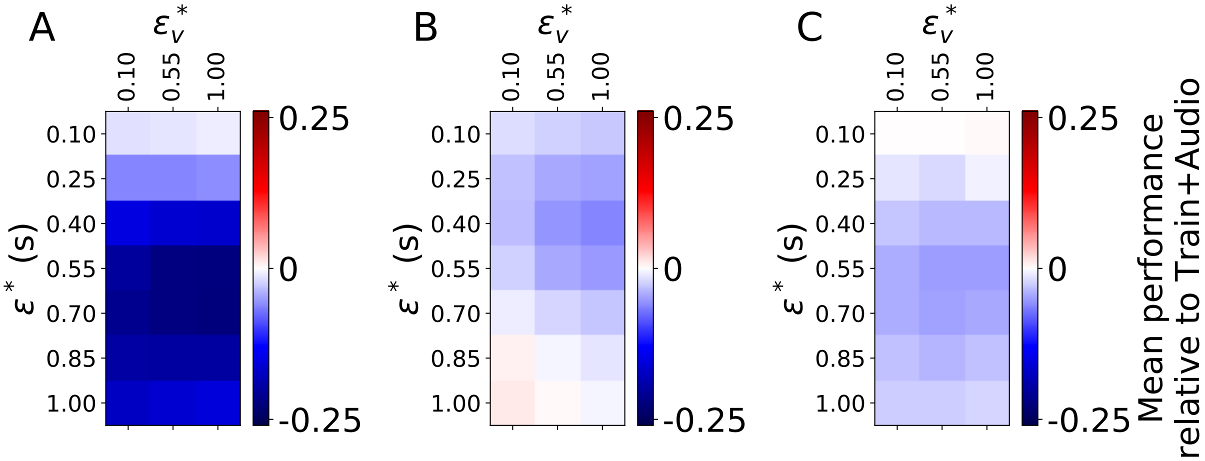

FIG. S5. (A) Difference in average performance of Visual Only relative to Train+Audio for different combinations of the time delay ϵ^* and velocity direction ϵ_v^* threshold. Rightmost column corresponds to the difference between the mean performance values shown in the bottom of Fig. 2. (B) Comparison of Train with Train+Audio. (C) Comparison of Audio with Train+Audio. Negative values (blue) indicate that performance is worse relative to Train+Audio.

landscapes in Fig. S5 where we show the change in average performance from (A) Visual Only to Train+Audio, (B) from Train to Train+Audio, then (C) from Audio to Train+Audio. The deepness of the blue indicates how much worse average performance in the shown condition is relative to Train+Audio, whereas red indicates relatively better performance. Consistent with results in the main text, subjects do much better with either training or audio than in the Visual Only condition as indicated by the blue-dominated leftmost graph. The rightmost column of these graphs corresponds to the differences in the mean performance values shown in Fig. 2. In the Train and Audio conditions in Figs. S5B and C, the enhancement for Train+Audio is concentrated at $0.2\,\mathrm{s} < \epsilon^* < 0.8\,\mathrm{s}$ across all ϵ_v^*. At the smallest shown $\epsilon_v^* = 0.1$, we again are at the limit where subjects all perform poorly because the threshold for error is so low, and so we find, as expected, a narrowing the range of performance across all conditions.

Appendix S5: Learning the performance landscape

Mapping the topology of the performance landscape by measuring every combination of parameters (τ, f) is infeasible. If, however, we assume that that the average performance landscape changes smoothly as we change the visual appearance of the avatar with parameters τ and f, we can measure a few key points and interpolate the missing ones. We model any particular measurement of the subject i's performance as a stochastic variable.

$$\pi_{\mathrm{i}}^*(\tau, f) = p_{\mathrm{i}}(\tau, f) + \eta_{\mathrm{i}}, \tag{S5.1}$$

where $\pi \in [0,1]$ has been mapped to the real line with the inverse logistic transform $\pi^* = -\log\left[1/\pi - 1\right]$ such that $0 \to -\infty$ and $1 \to \infty$. The first term in Eq S5.1 refers to the variation inherent to the subject, embodying how fluctuations in the performance landscape are correlated across different τ and f. It has mean $\langle p_{\mathrm{i}} \rangle = \mu_{\mathrm{i}}$. The second term in Eq S5.1 refers to an independent source of statistical noise η_{i} with mean $\langle \eta_{\mathrm{i}} \rangle = 0$ and width $\langle \eta_{\mathrm{i}}^2 \rangle = \alpha_{\mathrm{i}}^2$. The expected covariance between any two measurements is then

$$\langle \pi_{\mathrm{i}}(\tau, f)\pi_{\mathrm{i}}(\tau', f') \rangle = \langle \left[p_{\mathrm{i}}(\tau, f) - \mu_{\mathrm{i}}\right]\left[p_{\mathrm{i}}(\tau', f') - \mu_{\mathrm{i}}\right] \rangle + \delta_{\tau,\tau'}\delta_{f,f'}\alpha_{\mathrm{i}}^2 \tag{S5.2}$$

with delta function $\delta_{x,x'} = 1$ only if $x = x'$ and 0 otherwise.

We model the distribution characterized by the covariance in Eq S5.2 using Gaussian process regression (GPR). This technique is equivalent to a multivariate normal distribution of the observed data points with covariance that typically decays with increasing distance between two parameter sets (τ, f) and (τ', f') [9, 10], where the decay length determines the typical size of local features in the performance landscape.

When modeling the covariance function during the course of an experiment, we used different formulations for running the experiments including radial basis kernel G or an exponential kernel K, which are both common parameterizations of the kernel function. They are, respectively,

$$K_{\mathrm{i}}(d) = \theta_{\mathrm{i}} \exp\left(-d^2/2\sigma_{\mathrm{i}}^2\right) \tag{S5.3}$$

$$G_{\mathrm{i}}(d) = \phi_{\mathrm{i}} \exp\left(-d/\lambda_{\mathrm{i}}\right) \tag{S5.4}$$

with coefficients ϕ_{i} and θ_{i} and scale parameters λ_{i} and σ_{i}. Typically, the diagonal terms representing the noise are considered separate from the kernel function such that the covariance is the sum of the two:

$$\langle \pi_{\mathrm{i}}(\tau, f)\pi_{\mathrm{i}}(\tau', f') \rangle = K_{\mathrm{i}}(d) + \alpha_{\mathrm{i}}^2\delta_{\tau,\tau'}\delta_{f,f'} \tag{S5.5}$$

In addition to the kernel, we must also decide on a geometry for the performance landscape that determines the distance d in Eq S5.5. We use the geodesic distance on a hemisphere to observe the singularities at $f = 1$ and $f = 0$ corresponding to the north and south poles, where the longitude lines of performance at different τ all converge [11].

Combining these elements, the log-likelihood of the set of observed data points for subject i is given by the multivariate normal distribution

$$\log L_{\mathrm{i}} \propto -\sum_{\substack{x=(\tau,f)\\ x'=(\tau',f')}} \pi_{\mathrm{i}}(x) K_{x,x'} \pi_{\mathrm{i}}(x') - \frac{1}{2}\log|K| \tag{S5.6}$$

If the hyperparameters are not optimized at every step, the parameter combination (τ, f) of maximal predicted uncertainty is deterministic after every measurement because the log-likelihood does not depend on the value of performance measured at that point. If the hyperparameters are optimized, then the parameter combination with maximal uncertainty can change, but the computational cost of the calculation can be much higher.

We used different formulations of GPR depending on the experimental condition. For the untrained conditions, we used a radial basis kernel function without hyperparameter optimization at every trial. Thus for all the untrained trials, all the same points were measured on the performance landscape in the same order, though ensuring that the parameter combination with maximum uncertainty was selected next. For the trained conditions, we used an exponential kernel with online hyperparameter optimization. The difference in procedures means that the sets of points collected for the untrained trials are fixed throughout all subjects whereas for the trained subjects the measured points are scattered differently throughout the parameter space. When we model the aggregate landscapes at the end, however, we model all performance landscapes in the same way.

Appendix S6: Aggregate performance landscape

We combine the measured values across all subjects for a given experimental condition to construct an aggregate performance landscape that captures subject-specific fluctuations and structure common across subjects. We add onto Eq S5.1 a common landscape P across all subjects and noise H that is iid for every observation made with $\langle H \rangle = 0$ and $\langle H^2 \rangle = \beta$.

$$\pi_{\mathrm{i}}(\tau, f) = p_{\mathrm{i}}(\tau, f) + \eta_{\mathrm{i}} + P(\tau, f) + H \tag{S6.1}$$

The terms P and H do not have indices because they are shared across all subjects.

Then, we assume the corresponding covariance matrix has the block form

$$\begin{aligned} K_{\mathrm{ij}}(d) = a\, K_{\mathrm{co}}(d) + \delta_{\mathrm{ij}} \left(b_{\mathrm{i}}\, K_{\mathrm{i}}(d) + \delta_{\tau,\tau'}\delta_{f,f'}\alpha_{\mathrm{i}}^2\right) + \\ \delta_{\mathrm{ij}}\delta_{\tau,\tau'}\delta_{f,f'}\delta_{\mathrm{n},\mathrm{n}'}\beta^2 \end{aligned} \tag{S6.2}$$

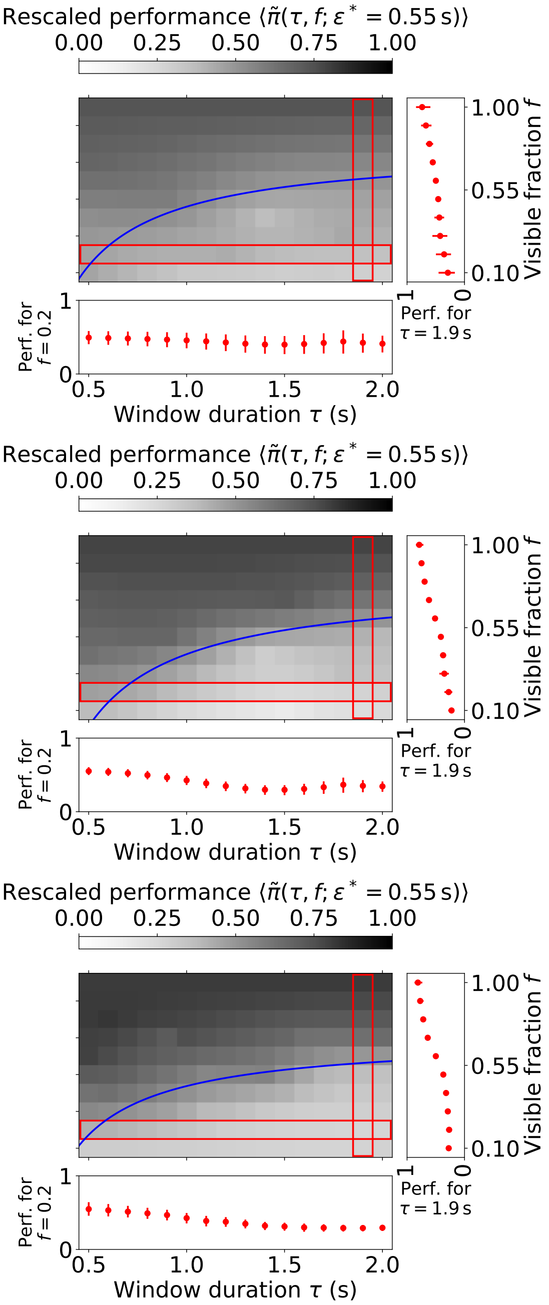

FIG. S6. Aggregated performance landscapes for the Visual Only, Audio, and Train conditions from top to bottom as predicted using Gaussian process regression. Blue lines are best fits to level contours of the rescaled performance $\langle \tilde{\pi} \rangle = 1/2$ of the form given by Eq 1.

with common kernel K_{co}, weight coefficients a and b, and each data point has a unique index n. Here, we use the more flexible Màtern kernel,

$$K(d) = \Theta \frac{2^{1-\nu}}{\Gamma(\nu)} (d/\lambda)^{\nu} \kappa_{\nu}(d/\lambda) \tag{S6.3}$$

where κ is the modified Bessel function of the second kind [12]. The Màtern kernel has a smoothness parameter ν

such that when $\nu = 1/2$ (with $0 \leq \nu \leq 1/2$ on a spherical surface) it is the exponential kernel from Eq S5.4.

Given the large number of parameters, we regularize the problem by imposing a sparseness constraint on the coefficients of the subjects when maximizing likelihood

$$f(\{\theta_\mathrm{i}\}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} |\theta_\mathrm{i}| \tag{S6.4}$$

such that subject specific terms in Eq S6.2 are driven to zero when the common landscape is sufficient to describe the subject's behavior. When the coefficient θ_i is driven to 0, the other parameters for that subject's kernel become irrelevant so this is an efficient way of reducing the dimensionality of the parameter space. We also find that the noise terms are often driven to 0 if they are not constrained even though fluctuations in the data seems to be strongly important even when comparing the same subject's performance for $f = 1$. Therefore, we add a weak regularization for the noise terms

$$\frac{1}{10^3\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} |\alpha_\mathrm{i} - 1/2| \tag{S6.5}$$

As validation of our choice of the structure of the covariance matrix, we find that the ratio of coefficients $a/\langle b\rangle_\mathrm{i}$ is not driven to 0, but varies from 0.2 to 4.8 indicating that shared structure in the performance landscape is important.

The maximum likelihood parameters we find describe a model that agrees well with the measured data points across all 24 combinations of the 4 experimental conditions and 6 values of ϵ^* with correlation coefficient of $\rho = 0.98$ when comparing π with $\hat{\pi}$. We also perform a cross-validation test by leaving one data point out of the data (such that the covariance matrix is one row and one column smaller) and comparing the prediction with the measured data point and find still yet $\rho = 0.95$ [13]. For each single landscape (for a fixed ϵ^* and experimental condition), we check directly the prediction error of this cross-validation procedure and find that the average norm error per landscape is less than 0.01. These statistics show that we have found a good fit to the performance landscape.

As we describe in the main text, we must aggregate over rescaled performance landscapes to show the transition curve shown in Fig. 1C. First, we rescale them such that they all reach the value of $\tilde{\pi} = 1/2$ at $(\tau, f) = (2.0, 0.6)$. Then, we set $\tilde{\pi}(f = 1) \approx 1$ and $\tilde{\pi}(f = 0) \approx 0$. The precise values for this last step in rescaling are chosen for maximum contrast, but the shape of the transition region does not depend on the upper and lower limits of the rescaled performance landscape. The result of this aggregation for the Train+Audio condition is in Fig. 1C and the other conditions are shown in Fig. S6.

Appendix S7: Modeling the transition

In the main text, we fit the level curve of performance in the region between high and low performance using a linear relation between the visible duration τ_vis and the invisible duration τ_inv parameterized by the two constants a and b in Eq 1. We find this curve by minimizing the total value of the boxes that the contour passes through

$$C = \sum_{\mathrm{k}=0}^{\mathrm{K}} \pi\left(\tau_k, f = \frac{1}{1+b} - \frac{a}{(1+b)\tau}\right)^2 \tag{S7.1}$$

where $\tau_\mathrm{K} = 2\,\mathrm{s}$. Since we restrict our contour to the limits of the parameter space we explore in this work, however, Eq S7.1 can be minimized by simply reducing the length of the contour.

In order to ensure that the contour passes through the grid and does not minimize length at the expense of fitting the level curve, we normalize by the length of the contour on the grid. This length is

$$L = \int_{\tau=\tau_0}^{\tau=2} \sqrt{1 + \frac{df}{d\tau}^2}\, d\tau \tag{S7.2}$$

where τ_0 corresponds to value of τ where the contour crosses the bottom limit of the measured performance landscape where $f = 0.1$,

$$\tau_0 = \frac{a}{-0.1b - 0.9} \tag{S7.3}$$

Combining Eqs S7.1 and S7.2, we have the objective function that we use to find the parameters in Eq 1

$$\min_{a,b} \mathcal{C}(a, b) = C(a, b)/L(a, b). \tag{S7.4}$$

The discreteness of the landscape means that a gradient-based minimization routine will fail. Instead, we evaluate the function across a grid of a and b to find the optimal solution. For all the experimental conditions, we find the values of a and b to yield very similar transition curves. The values are shown in Table II.

Appendix S8: Decay distributions

When we inspect the durations of time t that subjects are tracking the avatar closely, we find that the distributions $p(t)$ can be described by three main classes characterized by either an exponential tail, a gamma-like function with a dearth of the shortest decay times, or a heavier-tailed distribution. Although the exponential decay is a signature of a memoryless process, the remaining two distributions indicate that the dynamics of how subjects are tracking the motion of the avatar are not generated from an time-independent process. Here, we provide more detail on a few possible explanations for the form and origin of these distributions.

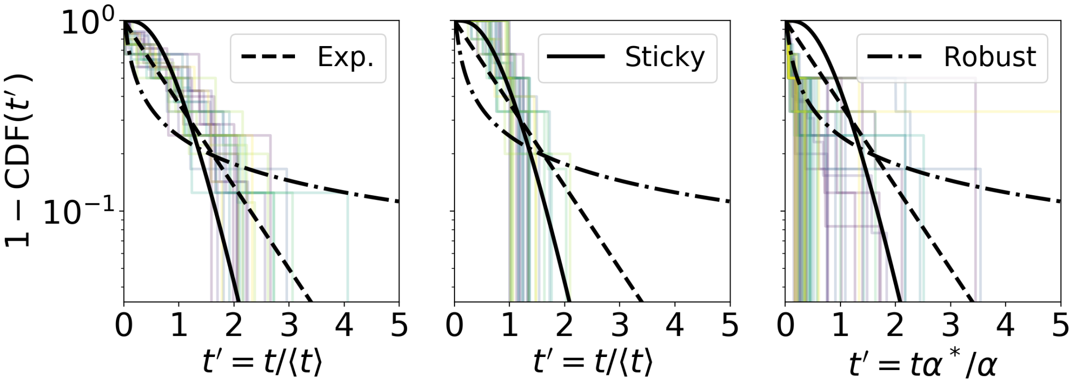

FIG. S7. Distribution of decay times for stable runs as indicated by shaded regions in Fig. 1B. (left) The most frequently occurring distribution is close to an exponential decay. (middle) "Sticky" distributions show a dearth of very short decay times. (right) "Robust" distributions show a power-law like decay with a heavy tail.

TABLE I. *
Transition contour parameters

Condition	a	b
Visual Only	0.41	0.28
Train	0.38	0.42
Audio	0.49	0.25
Train+Audio	0.34	0.50

TABLE II. Parameters found for Eq 1 using objective function in Eq S7.4. Compared to the other conditions, Train+Audio has a flatter transition zone showing that the transition to poor performance varies less with τ.

The predominant class of distributions are exponential. In the language of control theory, exponential decays are considered a "first to failure" process in a multicomponent system where failure manifests when the first component fails. This intuition suggests that while the average decay rate of the subject might depend on subject handedness, fatigue, difficulty of the task, or other factors, much of the variation around the average can be explained by a memoryless process as if the subject is susceptible to random fluctuations that lead to failure.

In contrast, we find another class of decay distributions that show a characteristic depletion of short decay times as indicated by a flat region of the CDF at small durations. If it were the case that subjects were not to decay straight from success (S) to failure (F) at mirroring but first had to decay to intermediary states behaviorally indistinguishable from S, we say subjects show "sticky" mirroring:

$$\mathrm{S}_0 \xrightarrow{k_0} \mathrm{S}_1^* \xrightarrow{k_1} \cdots \xrightarrow{k_{\mathrm{N}-1}} \mathrm{S}_\mathrm{N}^* \xrightarrow{k_\mathrm{N}} \mathrm{F} \tag{S8.1}$$

with decay rate constants k_i. For $\mathrm{N} > 1$, we would expect a flat region in the complementary CDF near $t = 0$ whose extent depends on the number of intermediary states before decay. Since the average time to decay is only determined by the sum of the rate constants $K = \sum_{\mathrm{i}=0}^{\mathrm{N}} k_\mathrm{i}$, we write the complementary CDF of decay times, otherwise known as the survival function, as a function of a single rate constant

$$1 - \mathrm{CDF}(t') = e^{-Kt'} \sum_{\mathrm{n}=0}^{\mathrm{N}} \frac{K^\mathrm{n} t'^\mathrm{n}}{\mathrm{n}!} \tag{S8.2}$$

where the distributions have been rescaled such that $K = 1$ in Fig. S7. In the limit of $\mathrm{N} \to \infty$, we recover the gamma distribution. We find that the measured values of N as calculated with maximum likelihood are concentrated at smaller values. Over 50% of the observed values smaller than or equal to 5 when $\epsilon^* = 1/2\,\mathrm{s}$, suggesting that enhanced dynamical stability corresponding to the "sticky" distribution is slight.

In the remaining trials, we find distributions dominated by very short decay times but with a heavy tail of "robust" long runs. With the exponential and "sticky" distributions, we observe dynamics that are consistent with subjects occupying success or failure states separated by "energy barriers." When the dynamics are dominated by the time it takes to escape a local energy minimum, we may expect an exponential distribution for decay times. When one energy minimum becomes strongly dominant such that there is little switching, however, the dynamics will instead be dominated by the width of the local energy basin. Using this intuition, we might expect that the first passage time for simple diffusion as shown in Fig. S7 model the data better than the other distributions,

$$1 - \mathrm{CDF}\,(t') = 1 - \sqrt{\frac{\alpha}{\pi}} \int_{1/30}^{t'=t\alpha^*/\alpha} t^{-3/2} e^{-\alpha/t}\, dt. \tag{S8.3}$$

Here, the lower limit is important and is given by our interpolation of the velocity trajectories at 30 Hz. We cannot get a scaling collapse by rescaling by the mean.

Instead, we rescale by the parameter α using a constant $\alpha^* = 0.05$ as we show in Fig. S7. For robust mirroring, it is as if the subject is trapped in some wide region characterized by successful mirroring.

[1] Epic Games. Performance and profiling, 2017.

[2] Vicon. Personal Communication. July 2016.

[3] G Welch and G Bishop. An introduction to the Kalman filter. Technical report, 2006.

[4] Motion Detector User Manual, 2018.

[5] F Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Trans Acoust, Speech, Signal Process*, 23(1):67–72, February 1975.

[6] Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about Dynamic Time Warping is Wrong. In *KDD/TDM 2004*, pages 1–11, Seattle, August 2004.

[7] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[8] Stan Salvador and Philip Chan. Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intell Data Anal*, 11(5):561–580, 2007.

[9] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, Singapore, August 2006.

[10] C E Rasmussen and C K I Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, 2006.

[11] Tilmann Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4):1327–1349, September 2013.

[12] Charles F F Karney. Algorithms for geodesics. *J Geod*, 87(1):43–55, 2013.

[13] David M Allen. Mean Square Error of Prediction as a Criterion for Selecting Variables. *Technometrics*, 13(3):469–475, August 1971.