Data-driven sparse skin stimulation can convey social touch information to humans

Supplementary Information

Mike Salvato, Sophia R. Williams, Cara M. Nunez, Xin Zhu, Ali Israr, Frances Lau, Keith Klumb, Freddy Abnousi, Allison M. Okamura, Heather Culbertson

1 Social Touch Dataset

1.1 Scenario Prompts

The scenario prompts that the toucher and receiver are given are shown below. The amusement prompt was used to acclimate participants with the procedure. The toucher audios were used in the scenario classification experiment. See Dataset SD1 for recordings of these prompts.

Amusement (Used for training only)

Toucher: Imagine this: You're spending time with the person sitting next to you. You're talking about small things here and there, but there are also some long pauses where neither of you has much to say. Then, they make a face and say something really, really funny. Suddenly everything feels lighter. You reach out and touch them to show your delighted amusement.

Receiver: Imagine this: You're spending time with the person sitting next to you. You're talking about small things here and there, but there are also some long pauses where neither of you has much to say. Sure, it's a little awkward. Silence is fine, but today you're feeling chatty. The person sitting next to you seems receptive, so you crack a joke. It must have worked, because suddenly everything feels lighter. When they touch you, expressing their amusement, you feel like maybe you saved the day, or at least the next ten minutes.

Attention

Toucher: Imagine this: You're at a crowded party with the person sitting next to you, and they've drifted into a conversation with someone else. You don't want to be rude and interrupt them, but you really need to ask them an important question. Turn back toward them, and touch them in a way that gets their attention.

Receiver: Imagine this: You're at a crowded party with the person sitting next to you, but you've drifted off into a side conversation with other people. The person you're talking to is telling you a fascinating story, and you're completely rapt. It's like no one else is in the room. You're not purposefully ignoring the person you came to the party with, but you're really focused on hearing all the details. Then, the person you came to the party with touches you in a way that signals they need your attention.

Gratitude

Toucher: Imagine this: You and the person sitting next to you are at a dinner party with a group of friends you've known for a long time. You start to sense the conversation getting dangerously close to a topic that could put you in a really uncomfortable position if aired in this group. The person sitting next to you picks up on your uneasiness and deftly takes the conversation in a different direction. You feel a deep sense of relief, and you want them to know you are grateful. Reach out and touch this person to express your gratitude.

Receiver: Imagine this: You and the person sitting next to you are at a dinner party with a group of friends you've known for a long time. You have a knack for being able to spot a train wreck before it happens, so when you sense the conversation is getting dangerously close to a topic that could damage the reputation of the person sitting next to you, you steer it back on track. You don't expect any credit for such agile social maneuvers, but when they reach out and telegraph "thank you" with their touch, you instantly know you've done good.

Happiness

Toucher: Imagine this: Today is the perfect day. Like magic, everything is going right, and everyone around you seems to be in a good mood. You're walking down the street in the sunshine, listening to your favorite song. Feel that little bounce in your step? You catch a glimpse of yourself in a store window, and dang – you look good! On a whim, you decide to pop in and buy a lottery ticket. Why not? Scratch, scratch, scratch...you win $50! This day just couldn't get any better. Now take this feeling of happiness and touch the person next to you to express it.

Receiver: Imagine this: When the person walking toward you just can't stop smiling, you know something is going really right. They're beaming, and you can just tell they're having the best day ever, almost walking on clouds. They bound over to you and reach out to touch you, and it's like an electric bolt of pure joy flows through you.

Calming

Toucher: Imagine this: It's 7:30 pm on a miserable, rainy Thursday, and you're waiting for the person next to you to meet you for dinner. It's been one hell of a week for them. Everything that could go wrong for them has gone wrong. Finally, you see them walk through the door, and they're completely frazzled. You can practically feel the stressed out energy radiating off their body. Go to this person and touch them in a way that feels calming.

Receiver: Imagine this: What a crappy week. You're stressed out, and things just keep piling on. You really aren't in the mood to meet the person sitting next to you for dinner, but you can't back out now. You walk through the door in a state, and that must show because they reach out and touch you in the most compassionate and tender way. You feel instantly understood, and it brings a wave of calm. Your blood pressure feels like it just dropped 20 points, in a good way.

Love

Toucher: Imagine this: You and the person sitting next to you are spending the afternoon together. You're walking to get a bite, the weather is amazing, and you're catching up on everything in the way that friends do. You look at them, and it suddenly strikes you how much this friendship means to you, that life is so much easier and better with them around. Reach out and touch this person to express your love for them.

Receiver: Imagine this: You and the person sitting next to you are spending the afternoon together. You're walking to get a bite, the weather is amazing, and you're catching up on everything in the way that friends do. You look at them, and it suddenly strikes you how much this friendship means to you, that life is so much easier and better with them around. They reach out to express their love for you.

Sadness

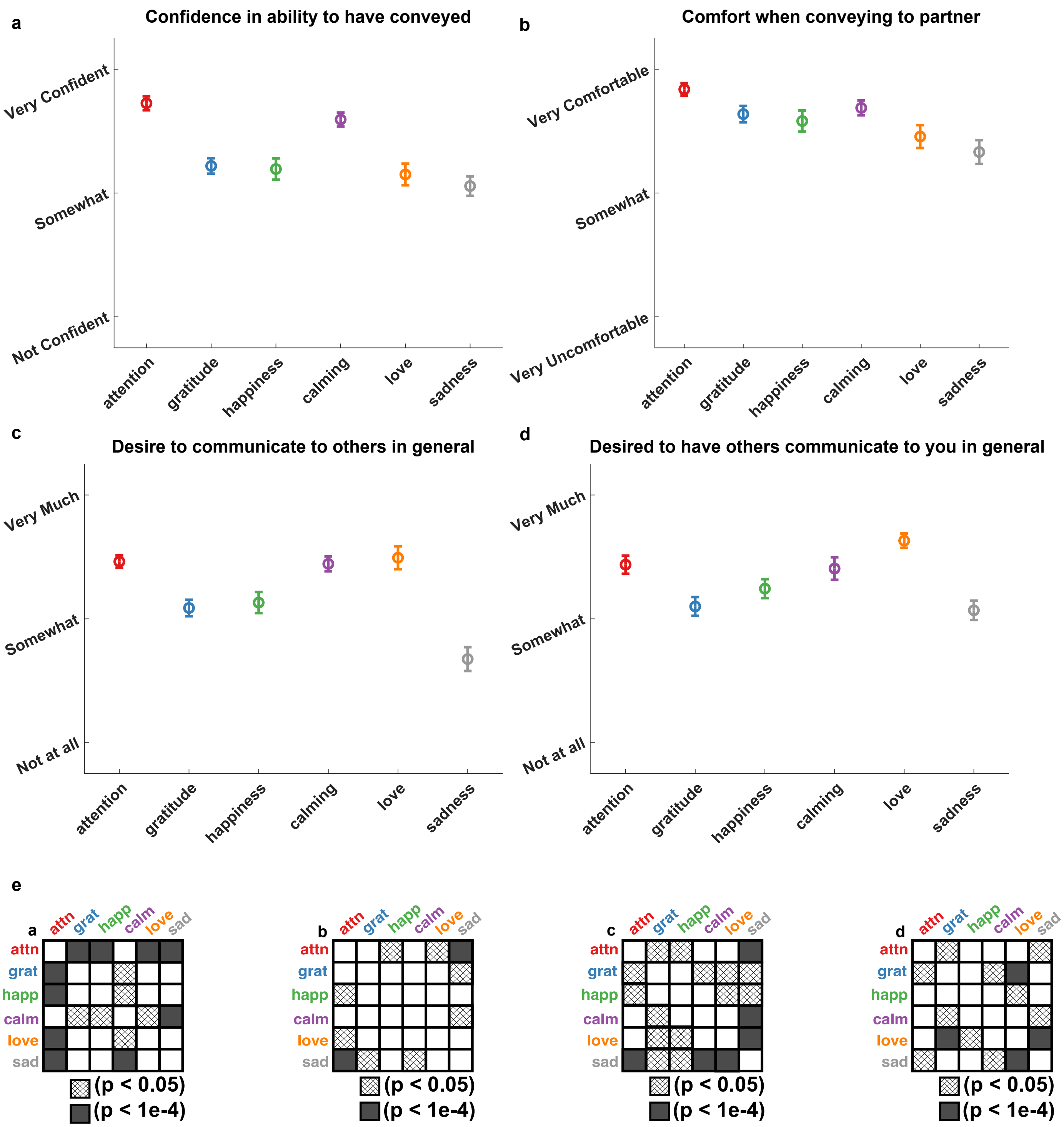
Toucher: Take a moment to think about someone you have lost – could be the death of someone close, or a breakup that tore you apart. Sit with that feeling for a bit... locate it in your body. Maybe it feels heavy, or achy. Get in touch with the sadness you feel about this loss, and touch the person next to you in a way that expresses that sadness.

Receiver: Even if it doesn't say anything out loud, a heavy heart is a loud presence. It's almost like another person in the room. Or maybe instead of a presence what you are feeling is an absence. The absence of joy. It seems like the person next to you is in mourning, like they've lost something that was important to them. That makes you instantly want to fill up the space with something like compassion or help or just being there.

1.2 Social Touch Survey

We provided a survey in which we asked participants to rate their thoughts on conveying each meaning through touch during the data collection process and in general along four scales. We used the Friedman test to check the effect of touch meaning on response, as our data was non-normal and not independent. We provide the exact prompt for the ”Attention” version of each question:

- *Rate your confidence in your ability to have conveyed **Attention** through touch alone.* (Fig. S1a) The Friedman test shows that touch meaning has a significant effect on participants' confidence in their ability to convey that meaning ($\chi^2(5) = 58.74$, $p = 2.2\text{e-}11$). The mean values for attention and gratitude were higher that the other touch meanings.
- *Rate your comfort level when conveying **Attention** to your partner in this study.* (Fig. S1b) The Friedman test also shows that touch meaning has a significant effect on participants' comfort when conveying an meaning to a partner ($\chi^2(5) = 38.55$, $p = 2.9\text{e-}4$). The highest average comfort level was reported for attention, whose distribution was statistically significantly different from the distributions for love and sadness.
- *In general, how much do you want to communicate **Attention** to other people through touch?* (Fig. S1c) The Friedman test shows that touch meaning has a significant effect on participants' desire to communicate that meaning to others ($\chi^2(5) =71.48$, $p = 5.0\text{e-}14$). Participants have the highest desire to communicate attention, calming, happiness, and love.
- *In general, how much do you want others to communicate **Attention** to you through touch?* (Fig. S1d) The Friedman test shows that the touch meaning has a significant effect on participants' desire to have others communicate that meaning to them ($\chi^2(5) = 59.74$, $p = 1.4\text{e-}11$). The mean values indicate that participants most desire others to communicate love, attention, and calming to them.



a	attn	grat	happ	calm	love	sad
attn		■	■		■	■
grat	■			▨		
happ	■			▨		
calm		▨	▨		▨	■
love	■			▨		
sad	■			■		

▨ (p < 0.05)
■ (p < 1e-4)

b	attn	grat	happ	calm	love	sad
attn			▨		▨	■
grat						▨
happ	▨					
calm						▨
love	▨					
sad	■	▨		▨		

▨ (p < 0.05)
■ (p < 1e-4)

c	attn	grat	happ	calm	love	sad
attn		▨	▨			■
grat	▨			▨	▨	▨
happ	▨				▨	▨
calm		▨				■
love		▨	▨			■
sad	■	▨	▨	■	■	

▨ (p < 0.05)
■ (p < 1e-4)

d	attn	grat	happ	calm	love	sad
attn		▨				▨
grat	▨			▨	■	
happ					▨	
calm		▨				▨
love		■	▨			■
sad	▨			▨	■	

▨ (p < 0.05)
■ (p < 1e-4)

Fig. S1. Social touch recording survey results. (**a**-**d**) The survey results for each question, based on a 7-point Likert scale. Error bars indicate standard error. (**e**) Associated significance matrix for each question, using the Dunn-Sidak post-hoc test for Friedman. This test was chosen as our data was non-normal and not independent.

2 Mapping Algorithm Formalization

Here we provide a full formalization of our algorithm, which that maps from data recorded on a 2D sensor array to an array of actuators. Our use case is to map from a pressure sensor array recording of human touch to a device consisting of a series of actuators worn by a human. The algorithm tracks trajectories of high pressure datapoints whose pressures vary smoothly over space and time, and chooses which of those trajectories to map onto the actuators. For the trajectories selected, our algorithm chooses an accurate rendering given the the actuator workspace limitations. The code is provided in Code SC1.

2.1 Function Input and Output

The algorithm provides a mapping

$$g : f_1 ... \times ... f_p \times a_1 \times ... \times a_q \rightarrow \mathbb{R}^{3\times p\times q}$$

where p is the number of data frames, q is the number of actuators. $f_1, ..., f_p \in F$ are a series of data frames (which are not constrained to be rectangular) and $a_1 ... a_q \in A$ are 3D geometric shapes.

Each frame $f_t \in F$ contains a series of coordinates $\mathbf{x} \in \mathbb{R}^2$ and associated intensities $I_t(\mathbf{x}) \in \mathbb{R}$ at time index t. These coordinates must be in metric coordinates (e.g. millimeters), and these frames define the "coordinate space." The intensities provide a measure of the probability of detection at that coordinate.

Each shape $a_i \in A$ represents the workspace for actuator i. The workspace is defined by the actuator's available motions (e.g. up and down, side to side, etc.) and the extents of its motion in any direction. It is represented in the same units as the coordinate space. The algorithm tracks only the center of each actuator, so the shape should represent the bound of how far the actuator center can move.

For each frame for each actuator, a 3D value for the location of each actuator is output. Let $L_i(t) \in \mathbb{R}^3$ be the location of actuator i at time t in coordinate space. The first 2 dimensions of $L_i(t)$ are coordinate distances, and last dimension is the measurement value.

To map values from the algorithm to hardware, a function

$$h : \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

maps coordinate space actuator location values $L_i(t)$ to desired 3D locations for each actuator. Typically, the first two dimensions would be mapped by the identity function, representing no scaling between coordinate space and the actuator output in those dimensions. Because the intensity values of sensor frames F are not necessarily directly mappable to a location in space, the third dimension will be determined based on the use case.

Step 1: Trajectory generation using multi-object tracking

In the first phase of the algorithm, we leverage multi-object tracking algorithms [1] to find the optimal contiguous paths tracked in the sensor. Such algorithms are typically used in computer vision to extend object detectors from single images to continuously tracked objects in video. They rely on the object detector providing a probability of a correct detection to the tracking algorithm. They then take all detections across all images, and provide a consistent set of tracked object trajectories through the video. We use this method to locate areas of contiguous, high pressure readings of sensor, which we assume represent interaction of a participants hand on the sensor. Our method for trajectory generation is an adaption of [1] for our sensor frames.

If we consider our sensor frames as video, we assume the pressure at each pixel location is monotonic with the probability that it represents the center of mass of pressure being applied by some object. This is a statement that higher pressure indicates a higher likelihood of meaningful contact with the sensor. Thus we could consider each pixel as a detection, with probability as a function of pressure. Let $\mathbf{g}$ be a pixel, where $\mathcal{G} = \{\mathbf{g}_1, ...\mathbf{g}_n\} = \{(t_1, \mathbf{x}_1), ..., (t_n, \mathbf{x}_n)\}$ is the set of all pixels, consisting of coordinates $\mathbf{x}$ over all times t, and let $V(\mathbf{g})$ be the intensity of a pixel. $V(\mathbf{g}) = V(t, \mathbf{x}) = I_t(\mathbf{x})$. Let $\bar{V}$ be the mean intensity over all pixel values in the data and σ_V be the standard deviation of those values. We use the following equations to calculate the functional probability of a detection:

$$z(\mathbf{g}_i) = \frac{V(\mathbf{g_i}) - \bar{V}}{\sigma_V}$$

$$P(\mathbf{g}_i) = \Phi(z(\mathbf{g}_i) - \sigma_k) * m$$

where $\Phi()$ is the cumulative distribution function (CDF) of the standard normal distribution. The above equation uses the z-score of a pixel value compared to all pixels in the entire sequence and determines its probability value by taking the standard normal CDF. σ_k is used to determine the number of standard deviations from the mean that a pixel must be for a value to be considered to have 0.5 likelihood of being a detection – in our case we used $\sigma_k = 1.25$. The constant $m < 1$ is used to prevent undefined values in later steps. In our work we used $m = 0.98$.

For computational efficiency we consider only pixels which are a local maxima. We define $\mathcal{G}' \subseteq \mathcal{G}$:

$$\mathcal{G}' = \{\mathbf{g}_i \mid V(\mathbf{g}_i) \geq V(\mathbf{g}_j) \text{ for } |||\mathbf{x}_i - \mathbf{x}_j|| \leq \sqrt{2}\}$$

In addition, we need a measure of the transition probability between detections $P_{\text{link}}(\mathbf{g}_i|\mathbf{g}_j)$, where $t_i = t_j + 1$

$$P_{\text{link}}(\mathbf{g}_i|\mathbf{g}_j) = \begin{cases} 1 - \frac{||\mathbf{x}_i - \mathbf{x}_j||}{k_d} & \text{if } ||\mathbf{x}_i - \mathbf{x}_j|| \leq k_d \\ 0 & \text{otherwise} \end{cases}$$

where k_d is a problem specific parameter. For our use case $k_d = 50$.

The goal is to find the set of trajectories $\mathcal{J}$ that best explains $\mathcal{G}'$ [1], with trajectories $J_k \in \mathcal{J}$, $J_k = \{\mathbf{g}_{k0}, ..., \mathbf{g}_{kl}\} \subset \mathcal{G}'$. This is formulated as a maximum a posteriori probability (MAP) estimate:

$$\mathcal{J}^* = \underset{\mathcal{J}}{\text{argmax}}\; P(\mathcal{J} \mid \mathcal{G}')$$

$$= \underset{\mathcal{J}}{\text{argmax}} \prod_i P(\mathbf{g}_i \mid \mathcal{J}) \prod_{J_k \in \mathcal{J}} P(J_k)$$

$$J_k \cap J_w = \emptyset, k \neq w$$

$$P(\mathbf{g}_i|\mathcal{J}) = \begin{cases} P(\mathbf{g}_i) & \mathbf{g}_i \in J_k \in \mathcal{J} \\ 1 - P(\mathbf{g}_i) & \text{otherwise} \end{cases}$$

$$P(J_k) = P_{\text{entr}}(\mathbf{g}_{k0}) P_{\text{link}}(\mathbf{g}_{k1}|\mathbf{g}_{k0}) ... P_{\text{link}}(\mathbf{g}_{kl}|\mathbf{g}_{kl-1}) P_{\text{exit}}(\mathbf{g}_{kl})$$

P_{entr} is the probability that a point starts a trajectory, P_{exit} is the probability a point ends a trajectory. $P_{\text{entr}} = P_{\text{exit}}$ is a tunable constant. We use e^{-8}, indicative of source/sink costs of 8 in the formulation by [1]. Increasing this value increases the number of trajectories that the algorithm is likely to find.

The constraint $J_k \cap J_w = \emptyset, k \neq w$ indicates that two trajectories cannot share an element.

A min-cost flow framework to solve for $\mathcal{J}^*$ is presented in [1].

Step 2: Actuator workspace restriction

The previous section gave us $\mathcal{J}^*$ which represents a set of trajectories from the sensor data. We will now find a mapping from the trajectories to our actuators.

Assume we have a set of 3D geometric shapes A which represent the bounds of motion for the center of an actuator in trajectory space. Each vertex should be mapped into the coordinate space of our sensor frames F. By positioning the shapes onto a particular location on the frame, they represent some area of the sensor space that we wish to actuate. By trying many transforms of the set of shapes, we can find the optimal portion of the sensor space to render based on which trajectories pass through that area. Let M represent the set of transforms we wish to try (e.g. translations and scalings). Let A_m represent the set of actuator workspaces transformed by transform $m \in M$.

We then find the set of trajectories to render given a workspace restriction. We assume that a given actuator can only render one trajectory at a time, so we wish to find a set of trajectories that would not require any actuator to render more than one trajectory at any time $\tau \in T$, where $T = 1, ..., p$ all times in which we have data frames. This implies a matching between two sets of data – the set of trajectories at time τ, and the set of actuator workspaces with any trajectories in them at time τ. In order for a set of trajectories to be valid and maximal, there must then be a one-to-one correspondence between the two sets – a bipartite perfect matching. Let

$$\mathcal{J}^{*\tau} = \{J \in \mathcal{J}^* | \exists \mathbf{g}_i \in J, t_i = \tau\}$$

Consider any $\Theta^\tau \subseteq \mathcal{J}^{*\tau}$. Let

$$\begin{aligned}\Xi^\tau = \{a \in m(A) | \exists J \in \Theta^\tau \text{ s.t. } \exists \mathbf{g}_i \in J, t_i = \tau,& \\ \mathbf{x}_i \text{ within } a \in m(A)\}&\end{aligned}$$

i.e. the actuator workspaces a where there exists a pixel $\mathbf{g}$ in a trajectory $J \in \Theta^\tau$, with location $\mathbf{x}$ within a at time τ. Consider the bipartite graph with vertex sets Θ^τ (trajectories) and Ξ^τ (actuator workspace) and edges between them. In order for all J s.t. $J \in \Theta^\tau$ to be compatible, the edges must be a bipartite perfect matching in this graph, otherwise two trajectories are in the same actuator workspace at the same time. By checking if such a bipartite perfect matching exists for each Θ^τ and associated Ξ^τ for all $\tau \in T$, we can determine which subsets of $\mathcal{J}^*$ can be rendered. For every timestep τ, we consider the bipartite graphs for all possible all $\Theta^\tau \subseteq J^{*\tau}$ and associated Ξ^τ. If a bipartite perfect matching is not possible for Θ^τ we know the set of trajectories J s.t. $J \in \Theta^\tau$ is invalid. Thus we can calculate the set of all sets of trajectories which are invalid $\Theta^\vee$. We use Hall's theorem to check the bipartite perfect matchings [2].

We wish to find the best set of compatible trajectories for our frame sequence and actuator workspace constraints. To do this we create a convex measure of trajectory quality:

$$R(J, m(A)) = \sum_{\mathbf{g}_i \in J} \log\left(\frac{P(\mathbf{g}_i)}{1 - P(\mathbf{g}_i)} * D(\mathbf{g}_i, m(A)\right)$$

$D()$ is a small factor that weights trajectory elements lower as they move further from the center of the actuator bounding shape they are within at a given time. In this work we set

$$D(\mathbf{g}_i, m(A)) = 1.02\left(1.04 - \frac{0.04||\mathbf{g}_i - c_i||_2^2}{c_r^2}\right)$$

where c_i is the center of the actuator workspace that $\mathbf{g}_i$ is inside of, and c_r is the radius of the workspace. This factor assumes that it is more preferable to render the center of a workspace than the edges. Let $\Theta^\vee$ be the set of all sets of trajectories that are incompatible as above, and $\mathcal{J}^{*'}$ be the power set of $\mathcal{J}^*$. Let $k(J, \mathcal{J}) = 1$ iff $J \in \mathcal{J}$.

We can then find the optimal trajectories by the following convex optimization problem:

$$\operatorname*{argmax}_{m \in M} \operatorname*{argmax}_{\mathcal{J} \in \mathcal{J}^{*'}} \sum_{J \in \mathcal{J}} R(J, m(A))$$

$$\text{subject to: } \left[\sum_{J \in \mathcal{J}, J \in \Theta} k(J, \mathcal{J})\right] < |\Theta|, \forall \Theta \in \Theta^\vee$$

The above can be solved with an off-the-shelf convex optimization solver such as CVX [3], [4] for the inner optimization, and by iterating over all $m \in M$ for the outer optimization.

We could have chosen to merge Step 1 and Step 2 into a single optimization where the trajectories are selected with information about the actuator workspaces provided. However, we chose to separate these procedures so that Step 1 represents the trajectories strictly as a function of the data, so they are a more accurate representation of the data itself than if a single optimization were performed.

3 Scenario Classification Experiment

3.1 Non-forced-choice Simulation

Our experiment was a forced-choice task. In Fig. S2 we simulate the notion that a user did not believe any response is valid. We assume that if a user assigns high probabilities to few options, they are less likely to have selected no scenario. Thus we plot the top-choice accuracy as a function of different probability cutoffs for the maximum probability scenario assignments. Top-n indicates the sum of the probability of the n highest probability scenario assignments is calculated. Specifically, consider $[p(s_1), ..., p(s_6)]$ to be the sorted probability assignments for the 6 scenarios in some instance, where $p(s_1) \geq ... \geq p(s_6)$. The summed probability for that instance in the top-n case is $\sum_{i=1}^{n} p(s_i)$. We calculate accuracy as in Fig. 2, except even if the top-choice response was correct, it is marked incorrect if the probability sum is less than the cutoff. For example, if a user rated "attention" and "gratitude" highly, it is possible they would have selected one of those 2 rather than a "no response" option. The top-2 graph can be used to analyze such behaviors. By providing this analysis participants no longer need to decide at what certainty level they would select no scenario, and instead a more complete profile of user perception can be analyzed.

3.2 First Round Classification Results

For top choice classification, Fig. S3 compares the first round classification results with the overall result. We see 42% accuracy for the first decoding compared to 45% accuracy overall, and with similar areas of confusion.

3.3 Self-Assessment Manikin clustering

We provide clustering analysis for the Self-Assessment Manikin ratings of the displayed haptic signals (Fig. S5). Using the Calinski-Harabasz [5] criterion we find only 5 clusters, unlike the 7 obtained when clustering on the user-assigned probabilities (Fig. 8). We see two clusters similar to those obtained via the user-assigned probabilities. Cluster 4 here is similar to Cluster 7 from the user-assigned probabilities, a cluster which primarily contains points from the sadness scenario. Cluster 5 here is similar to Cluster 6 for the user-assigned probabilities, a cluster which has many points from calming, love, and sadness. While the user-assigned probabilities had clusters which seemingly exemplified each of happiness, gratitude, and attention, no such clusters exist in the Self-Assessment Manikin clustering. This may be evidence that the 2-dimensional Self-Assessment Manikin may be insufficient to fully capture common modes of interpretation by the subjects.

3.4 Actuator Signal Measurement

We recorded the output of the each voice coil actuator laid flat (Fig. S4) using the Micron Tracker Sx60 (ClaroNav, Toronto, ON, CA), a vision tracker with sub-millimeter resolution. We recorded six trials for each scenario and report the means and standard deviations of the actuator displacements (Figs. S6-S7). We see that the actuators do not directly match the commanded signals, due to differing response characteristics for each actuator. This is possibly due to the use of voice coils that were originally designed to produce sound, where accuracy of frequency response is more important than amplitude. However, the signals sent to the actuators are repeatable across trials, as demonstrated by the small standard deviations. Integrating closed-loop control on the actuator forces or displacements using sensors is left for future work.

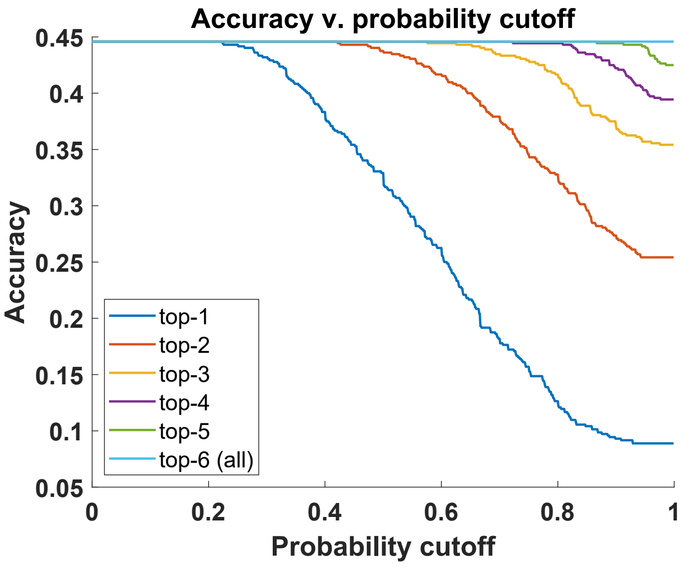
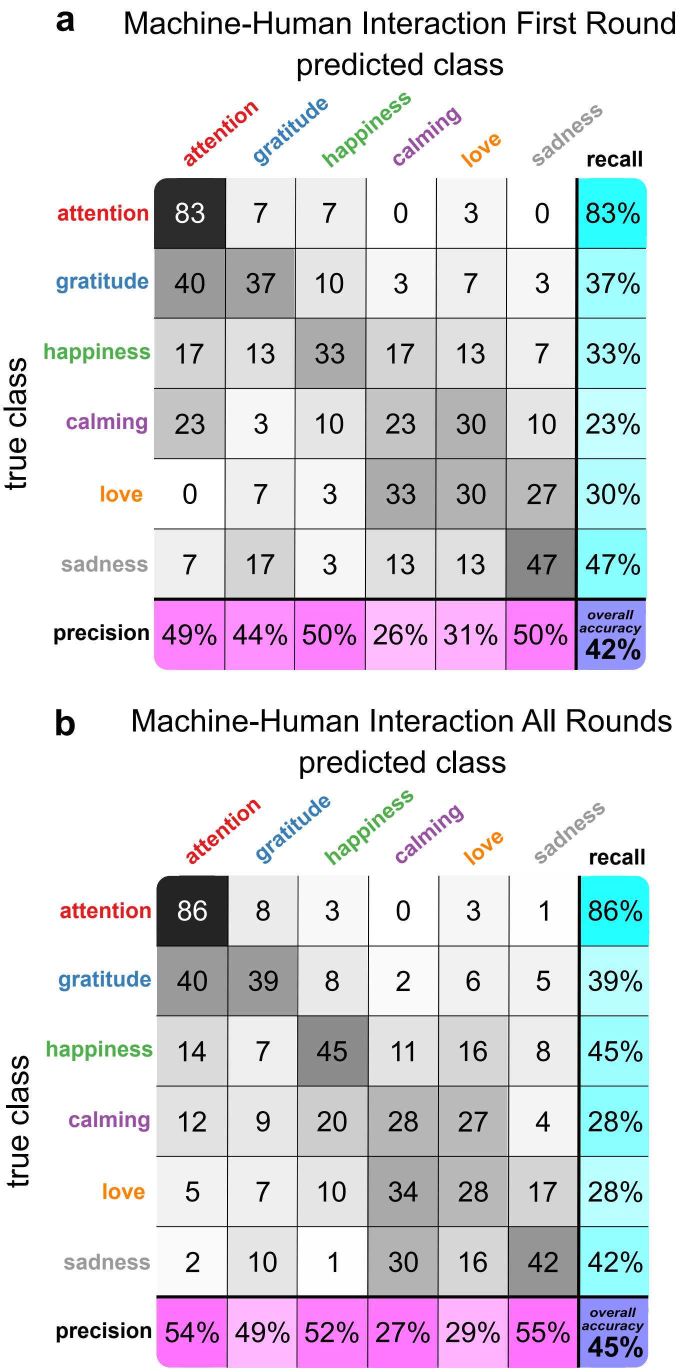


Fig. S2. Top-choice accuracy as a function of different probability cutoffs for the maximum probability scenario assignments. Top-n indicates we sum the probability of the highest n scenario probability assignments. Even if the top-choice response was correct, it is marked incorrect if the probability sum is less than the cutoff. For example in the top-2 line consider an instance where the summed weight of the top 2 choices is 0.3. For probability cutoff of at least 0.3 it will be marked incorrect, regardless of whether the top choice selection was correct.



a Machine-Human Interaction First Round

predicted class → true class ↓	attention	gratitude	happiness	calming	love	sadness	recall
attention	83	7	7	0	3	0	83%
gratitude	40	37	10	3	7	3	37%
happiness	17	13	33	17	13	7	33%
calming	23	3	10	23	30	10	23%
love	0	7	3	33	30	27	30%
sadness	7	17	3	13	13	47	47%
precision	49%	44%	50%	26%	31%	50%	overall accuracy 42%

b Machine-Human Interaction All Rounds

predicted class → true class ↓	attention	gratitude	happiness	calming	love	sadness	recall
attention	86	8	3	0	3	1	86%
gratitude	40	39	8	2	6	5	39%
happiness	14	7	45	11	16	8	45%
calming	12	9	20	28	27	4	28%
love	5	7	10	34	28	17	28%
sadness	2	10	1	30	16	42	42%
precision	54%	49%	52%	27%	29%	55%	overall accuracy 45%

Fig. S3. Comparing first round top choice classification with all rounds top choice classification. Each figure is a confusion matrix for the scenario that participants thought was most likely displayed by the rendered signal. They provide information on the participants' overall accuracy, how participants confused signals, and participants' precision and recall for their top scenario choices. (**a**) First of the four decoding rounds. Rows are normalized to 100, with 30 samples per row in raw data. (**b**) All four decoding rounds. Rows are normalized to 100, with 120 samples per row in raw data.

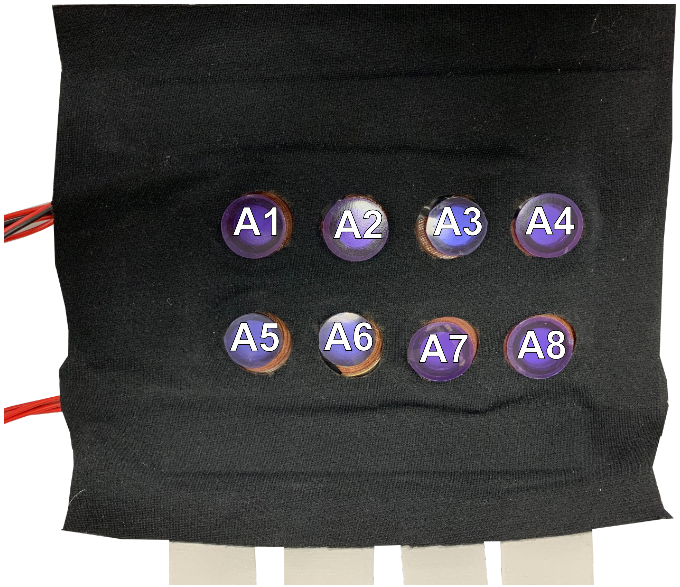


Fig. S4. Voice coil actuator sleeve laid flat. Each actuator has a thin plastic covering to increase surface area. Signals are measured while sleeve is laid flat.

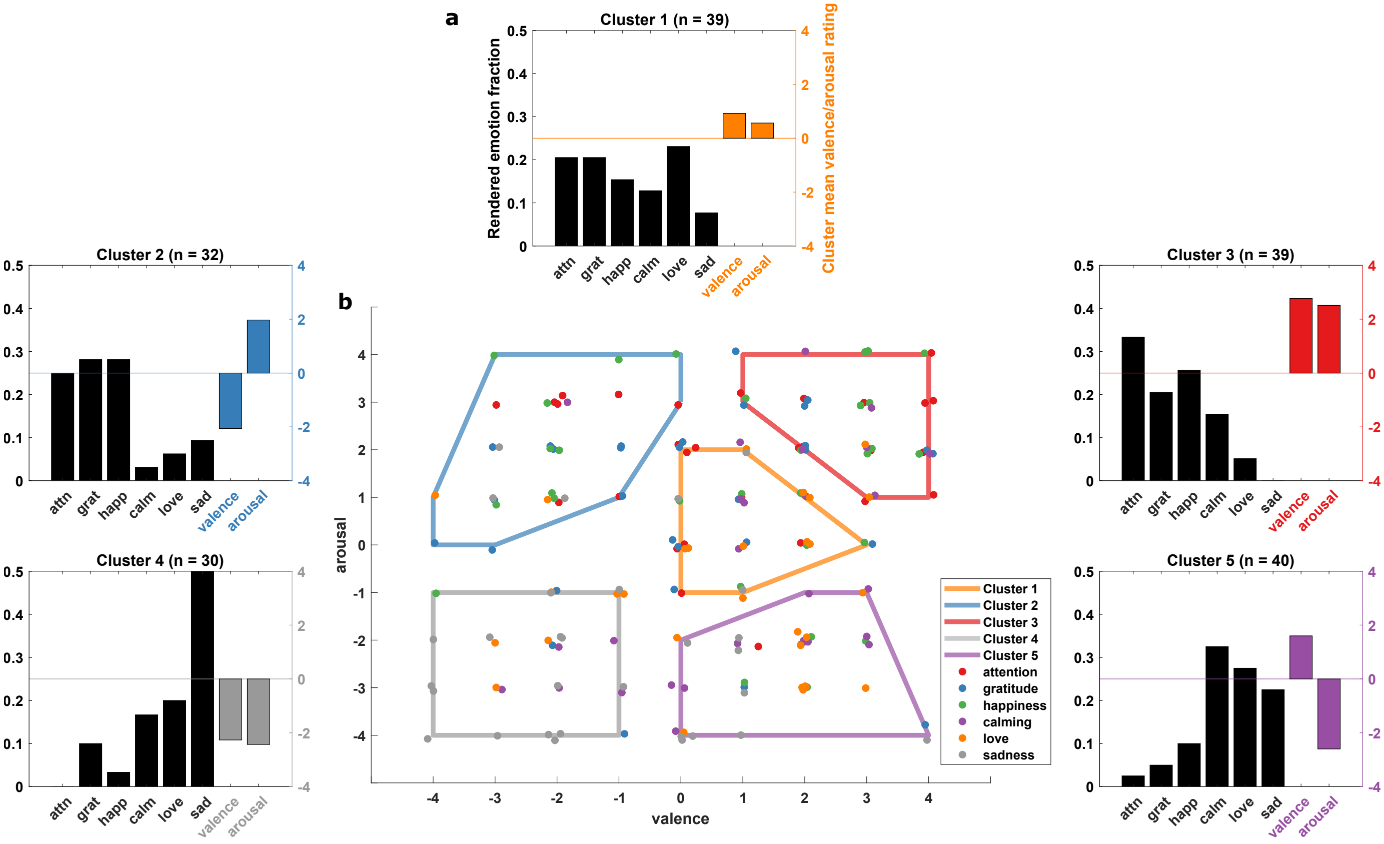


Fig. S5. Self-Assessment Manikin cluster analysis. Each subject provided valence and arousal ratings for each signal, providing a 2-dimensional vector describing their score. (**a**) The outer figures show the results of a K-means analysis. The points were clustered to find consistent responses across participants. The black bars show the fraction of data points in the cluster where the signal played was from that scenario. The colored bars show the vector value of the mean of the cluster. This can be interpreted as the black bar showing the presented signal, and colored bar showing the mean valence and arousal participants indicated for signals in that cluster. (**b**) The central figure provides a cluster visualization. Each participant's valence and arousal scores are plotted in 2D. The point color represents the presented signal. Because the data is discrete, a small amount of noise is added to each point so they do not directly overlap. We then drew the convex hull of the points in each cluster from (a). This is a two-dimensional visualization of cluster locations, in order to provide intuition for what the graphs in (a) represent.

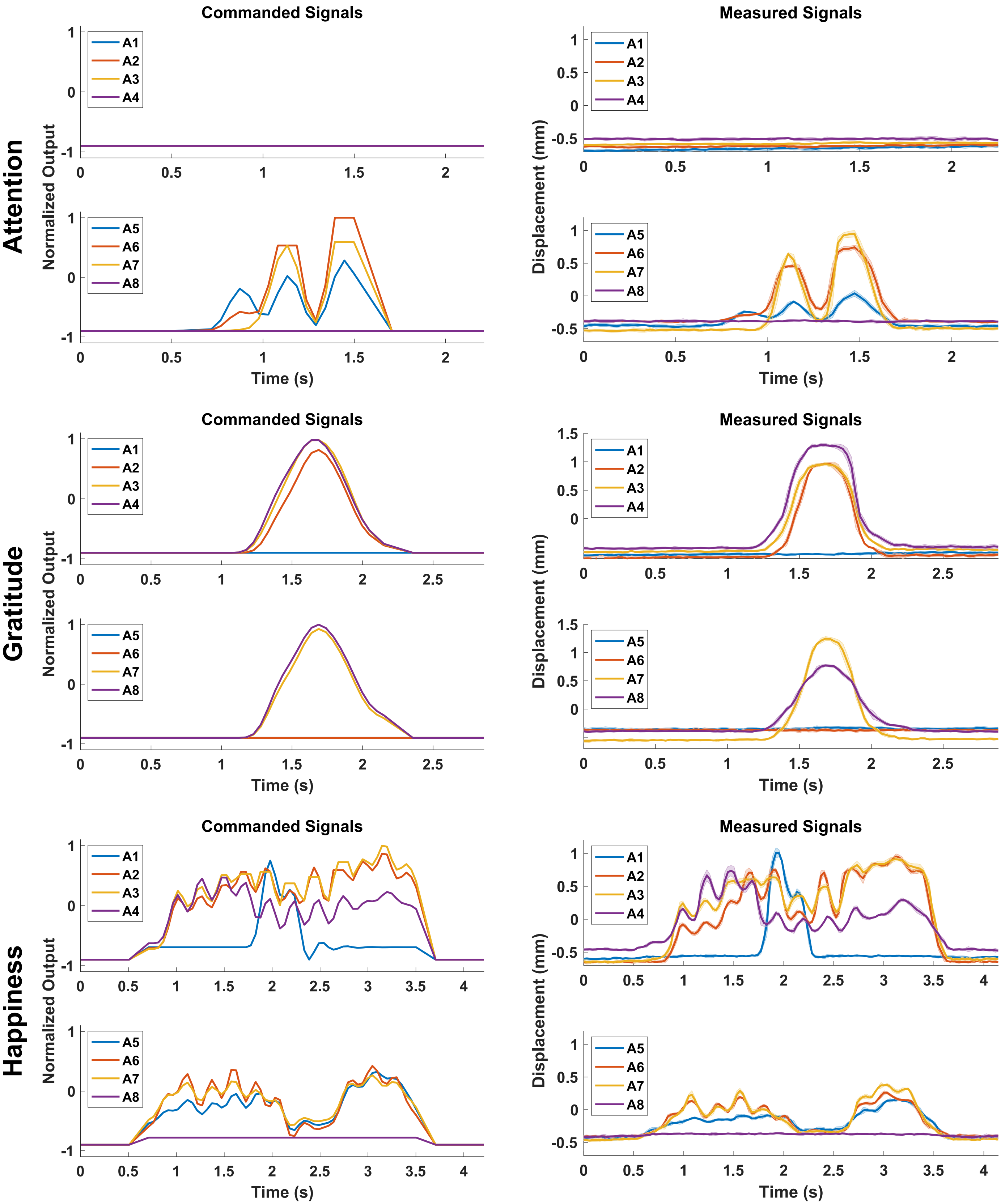


Fig. S6. Commanded and measured social touch signals. The commanded (left) and measured (right) signals sent to the voice coil actuators. The shaded error bars of the measured signals represent the standard deviation of six repetitions.

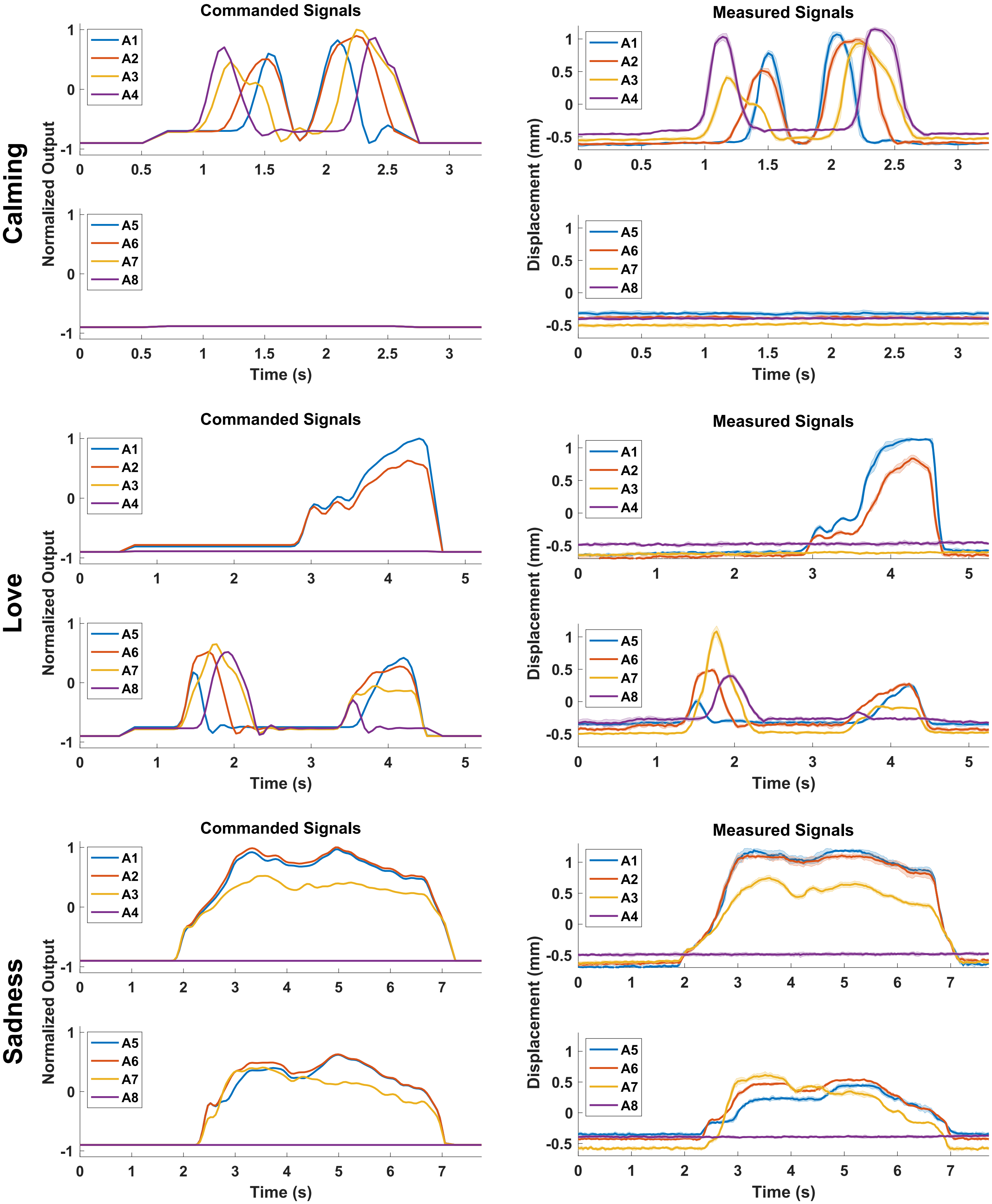


Fig. S7. Commanded and measured social touch signals. The commanded (left) and measured (right) signals sent to the voice coil actuators. The shaded error bars of the measured signals represent the standard deviation of signal repetitions.

4 Supplementary Media, Code, and Data

Movie SM1: Social Touch Recording Demonstration

This is a video showing an example interaction from our social touch dataset collection. The video shown here is a demo where two authors on the paper interact, and this data is not included in the social touch dataset. The audio is toucher version of the "love" scenario.

Movie SM2: Voice Coil Actuation

This is a video of the voice coil actuators displaying our "love" signal. Fig. S7 shows the commanded and recorded signal. We see there is a stroking motion followed by a squeezing motion.

Code SC1: Mapping Algorithm Code

https://github.com/charm-lab/social_multiobject_tracking

Dataset SD1: Audio for the Scenario Prompts

https://stanford.box.com/v/sparse-social-touch Folder: audio_prompts. (text available in section SI Social Touch Dataset: Scenario Prompts).

Dataset SD2: Public Social Touch Dataset

We provide the recorded data for public use. The dataset consists of PPS sensor data for each touch stored as 3D Python numpy arrays. We also provide annotations for which gesture was being used during this touch. If more than one clear gesture was made, the sections with each gesture are annotated. https://stanford.box.com/v/sparse-social-touch Folder: pressure_data.

Dataset SD3: Classification Experiment Results

In the scenario classification experiment we asked each participant to indicate the probability that a displayed signal was drawn from each scenario. This was repeated three times for each signal and recorded. In addition, for each signal we asked each participant to rate the valence and arousal of that signal, the results of which are available here. https://stanford.box.com/v/sparse-social-touch Folder: classification_results.

References

[1] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[2] S. Chawla, "Lecture notes in advanced algorithms – lecture 5: Applications of network flow," The University of Madison Wisconsin http://pages.cs.wisc.edu/~shuchi/courses/787-F09/scribe-notes/lec5.pdf, 2009.

[3] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[4] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[5] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics–Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.