

Appendix

A The CLAMP device

A.1 : Hardware details

The details of the sensors and peripherals onboard the CLAMP device are provided below:

Active and passive thermal sensing. The CLAMP device uses two 10 k Ω B57541G1103F NTC thermistors for active and passive thermal sensing (two different modalities). The active thermal sensor is maintained at a temperature of 55°C. The change in active thermal sensor readings over time, when the sensor comes in contact with an object, is indicative of the heat capacity of the object. The passive thermal sensor measures the surface temperature of the object in contact.

Force sensing. The CLAMP device uses an Interlink UX 402 force-sensing resistor (FSR) that can measure forces up to 150N. We calibrated the force-sensing resistor with an FX29 load cell. The resulting calibration curve is described by an exponential function of the voltage, achieving an $R^2 = 0.980$. While not as accurate as load cell sensors or MEMS force sensors over long-term cyclic use, force-sensing resistors offer significant advantages in terms of resilience and cost.

Vibration sensing. The CLAMP device uses a 20 mm diameter piezo disc, also known as a contact microphone, and a MAX4466 amplifier with a gain of 25 to measure audio signals resulting from contact.

Proprioceptive sensing. An important aspect of haptics is that sensing depends on action. The contact forces generated while grasping an object depend on the velocity and the angle at which the object is grasped. To address this, the CLAMP device has two 6-axis MPU6050 IMUs. The axes of the two IMUs are oriented such that the Y-axis of IMU-2 aligns with the Z-axis of IMU-1, while the x-axis of IMU-2 is transformed at an angle of -25° from that of IMU-1, about the Z-axis of IMU-1.

B The CLAMP dataset

B.1 Data processing

The raw haptic signals from the CLAMP device are processed in two stages: first, they are smoothed to produce visually interpretable data; then, features are extracted for model learning. We outline the exact details below:

1. We first synchronize all sensor data (from the active and passive thermal sensor, the force sensor, the contact microphone, and the two IMUs) based on their timestamps, to within 2 ms of each other.
2. We convert raw voltage readings from the active and passive thermal sensors to $^\circ\text{C}$ using the resistor values provided in the datasheet.¹
3. Data from some modalities are filtered to remove noise:
 - Active and passive thermal sensor data is passed through a Butterworth filter.
 - IMU data is passed through a causal moving average filter.

This results in haptic data that is visually interpretable.

4. We create the nine features from the smoothed sensor data. Specifically,
 - (a) For angular velocity feature, we find the gyroscope readings of IMU-2 relative to motion of IMU-1, in the frame of IMU-1. Hence, we consider the relative angular velocity of IMU-2 about the Z-axis of IMU-1 as a feature.

¹Despite access to a curve fit obtained from calibration, we do not convert raw voltage readings to readings in newtons for the FSR. We train our model on uncalibrated force values. We verified that training material recognition models using calibrated force values does not improve performance.

(b) For impedance, we use the following formula:

$$Z(t) = \begin{cases} \frac{F'(t)}{\omega(t)} & \text{if } \omega(t) \geq \delta \\ 0 & \text{if } \omega(t) < \delta \end{cases}$$

where $F'(t)$ denotes the force difference at time t , $\omega(t)$ denotes the angular velocity feature, and δ denotes a threshold for angular velocity, which we fix as $3^\circ/s$ for our experiments. While impedance is typically computed via linear velocity, we use angular velocity because our IMU-based proprioception is more accurate in this dimension. We fix a lower bound to exclude spurious values of high impedance that we observe at low angular velocities. This often happens when users attempt to change the grasp contact point, resulting in a sudden increase in force.

5. We apply a causal moving average filter on each feature except the contact microphone. The contact microphone readings are debiased and then downsampled.
6. Finally, the length of all features is set to 491 timesteps. Shorter features are padded with the last value or 0, depending on the feature.

B.2 Synchronizing contact

Contact is defined differently for sensors, depending on the suction cup they are located on. We synchronize contact for both cups by using a rule-based approach for sensors on each cup.

- Contact on the left cup is determined by the active thermal sensor readings. Contact is detected when the cup was not in contact on the previous timestep, and the active-thermal rate drops below $^\circ C/s$. Contact is released when the sensor was in contact on the previous timestep, the thermal rate becomes positive, and the temperature is below $53^\circ C$ — preventing the on-off controller’s response from registering as contact.
- Contact on the right cup is determined by the force sensor readings. Contact is detected when the cup was not in contact on the previous timestep, and the force signal exceeds a certain threshold. Contact is released when the sensor was in contact and the force falls below the same threshold. We set the threshold as 0.01V.

Once all contact onsets and releases in a trial have been detected, the trial data is segmented to include only the periods of contact. All features shorter than 491 timesteps are padded.

B.3 Filtering dataset for model learning

The CLAMP dataset contains many data points that are not directly usable for our task of material recognition. We exclude the following data points that are a part of our dataset, for model training:

- Contact instances with objects of material granite and dry wall, the two smallest material classes. This helps to alleviate the class imbalance problem in material recognition.
- Contact instances with non-zero force at the first timestep, which indicates that contact occurred before recording began.
- Contact instances where initial active temperature is less than $51^\circ C$. Prior work has shown that material recognition is challenging under varying initial conditions [?].
- Contact instances where one or more sensor modalities show erroneous readings.
- Contact instances with maximum angular velocity below a threshold.
- Contact instances with objects that have heterogenous surfaces.

79 B.4 Data Annotation

80 We provide the following prompt to GPT-4o, to generate ground truth annotations for material. The
81 system-level prompt is as follows:

```
82 You are an oracle that is part of the Haptic Dataset project. Your role is to  
83 inform us what object it is, and all the materials that the object is made of.  
84 A reacher-grabber will grasp this object from the left and the right.  
85
```

87 The prompt contains text along with more than 12 in-context examples. The prompt is as follows:

```
88 You will be provided with an image and a human-generated audio annotation  
89 converted to text as input. Respond in the following manner:  
90 1. Object:  
91 2. Materials:  
92 3. Heterogenous Surfaces:  
93 Here are some rules :  
94 - Under the 'Object' section, specify the object that is being referred to  
95 using just the image and the prompt.  
96 - Under the 'Materials' section, identify ONLY ONE material that the object is  
97 in contact with, directly or indirectly. Choose a material ONLY the following  
98 materials lists: [foam, aluminium, wood, steel, dry_wall, soft_plastic,  
99 hard_plastic, glass, paper, porcelain, granite, cardboard, rubber,  
100 vegetable_matter, fabric, brass].  
101 - If the material specified by text is 'Unknown', use your the image to  
102 recognize the materials.  
103 - Under the 'Heterogenous Surfaces' section, specify if the object is made up  
104 of different materials on the two opposing sides  
105
```

107 Each in-context example is provided in the following format (the horizontal line demarcates the two
108 inputs sent as one user prompt and the assistant prompt containing the example annotation

```
109 Input audio transcription : It's a painted stainless steel cup.  
110 Image : <encoded image>  
111 -----  
112 1. Object: cup 2. Materials: steel 3. Heterogenous Surfaces: No  
113
```

115 B.5 Dataset statistics

116 We demonstrate the diversity in the CLAMP dataset along three axes: distribution of object materi-
117 als, grasping forces, and grasping speeds. In Figure 1, we show the spread in values of maximum
118 force, maximum grasping speed, along with object material.

119 For the material distribution, we only consider objects that do not have heterogenous surfaces.
120 dataset contains 5357 objects, 680 ($\sim 13\%$) feature objects with heterogeneous surfaces. For the
121 remaining homogeneous objects, the material labels contain significant class imbalance, with the
122 largest class size (hard plastic) containing approximately 1000x as many samples as the smallest
123 class (dry wall).

124 Additionally, we plot the duration of contact instances as a histogram as well.

125 C The CLAMP model

126 C.1 Model training details

127 **Hyperparameters and code details.** The hyperparameters for the three experiments: haptic en-
128 coder training, CLAMP model pretraining, and CLAMP model finetuning on robot data, are pro-
129 vided in Table 1. The training code for the haptic encoder was written using the TSAI [?] API,
130 while that for the CLAMP model pretraining and finetuning was written using PyTorch API.

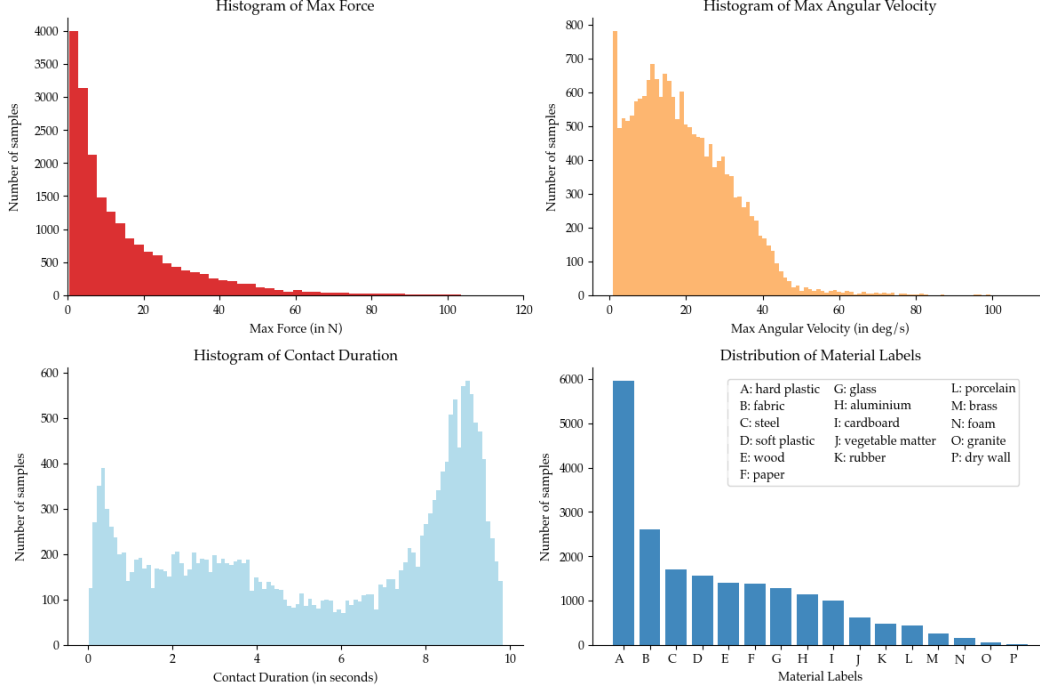


Figure 1: Visualizing the diversity in the CLAMP Dataset along various axes

Label weights. Experiments 1 and 2 use label weights from the CLAMP pretraining dataset. Since the robot-collected fine-tuning set has a different material distribution, Experiment 3 recalculates weights as the inverse class frequencies of the fine-tuning data instead of reusing the pretraining weights.

Layers used for learning. During pre-training on the CLAMP dataset, we freeze the haptic encoder (its weights remain fixed). During CLAMP model fine-tuning, we continue train the fusion MLP with the new label weights and unfreeze the haptic encoder to adapt it to haptic data from the robot embodiment.

Haptic encoder pre-training		CLAMP model pre-training		CLAMP model finetuning	
Hyperparameter	Value	Hyperparameter	Value	Hyperparameter	Value
Learning rate	1e-5	Learning rate	1e-5	Learning rate	1e-5
Weight decay	0	Weight decay	0	Weight decay	0
Filters in InceptionTime model	256	Filters in InceptionTime model	256	Filters in InceptionTime model	256
Batch size	64	Batch size	64	Batch size	64
Epochs	100	Epochs	120	Epochs	30
		λ_{kl}	0.1	λ_{kl}	0.1

Table 1: Hyperparameters for learning experiments

Seeds used for experiments. We perform all our experiments on the seed, chosen out of 3 random seeds, on which the haptic encoder shows the worst performance. We use this seed, 18, for all the experiments in this paper.

Unknown and Uncertain Predictions. For the CLAMP model, we find that $p_1 = 0.45$ and $p_2 = 0.25$ strikes a good balance between increase in performance (across accuracy and nMCC) and number of predictions filtered.

C.2 Extracting Visual Features from GPT-4o

We obtain log-probabilities from GPT-4o, renormalize them to redistribute probability mass across classes, and convert the results into logits for low-dimensional feature fusion.

148 To obtain a material prediction and log-probabilities, we employ a two-step prompting approach for
 149 better material classification. In-context examples are provided for only the second step. We initially
 150 feed in an image to identify the object in question. Then, we further prompt GPT with the prediction
 151 object and input image to generate a prediction for object material. For this step, we request the top
 152 14 logits. We filter out logits with tokens that do not belong to the list of materials, and assign a
 153 probability of 0 to those material classes for which a log-probability was not generated.

154 For the classes for which a log-probability was generated, we find the probability, take the fourth
 155 root, and then re-normalize the list of logits. This step ensures a wider spread of probabilities across
 156 material classes.

157 **D Real robot experiments**

158 For all real-robot experiments, we assume the grasp of each object. For the sorting and metallic ob-
 159 ject retrieval experiments, each object also has a pre-defined pose from where an image is captured.

160 **D.1 Sorting recyclable from non-recyclable items**

161 For this experiment, we used an unknown prediction threshold $p_1 = 0.18$ and an uncertain prediction
 162 threshold of $p_2 = 0.04$. The rules governing sorting for this experiment, based on object material,
 163 are:

Material	Trash/Recycle
Aluminium	Recycle
Brass	Recycle
Cardboard	Recycle
Fabric	Trash
Foam	Trash
Hard plastic	Trash
Paper	Recycle
Porcelain	Trash
Rubber	Recycle
Soft plastic	Trash
Steel	Recycle
Vegetable matter	Trash
Wood	Trash

Table 2: Rules for sorting recyclables

164 **D.2 Separating ripe from overripe bananas**

165 For this experiment, we classified bananas as ripe if the compliance prediction was hard, and over-
 166 ripe if the prediction was soft.