CLAMP: Crowdsourcing a LArge-scale in-the-wild haptic dataset with an open-source device for Multimodal robot Perception

Pranav N. Thakkar[†1], **Shubhangi Sinha**[†1], **Karan Baijal**[1], **Yuhan (Anjelica) Bian**[1],
Leah Lackey[1], **Ben Dodson**[1], **Heisen Kong**[1], **Jueun Kwon**[1], **Amber Li**[1], **Yifei Hu**[1],
Alexios Rekoutis[2], **Tom Silver**[1], **Tapomayukh Bhattacharjee**[1]

[†]Equal Contribution

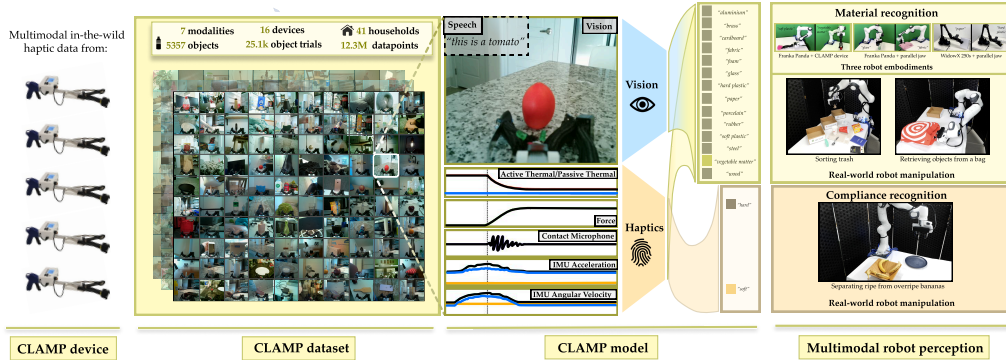[1]Cornell University, [2]Horace Mann School

Figure 1: We present the CLAMP device for collecting multimodal, in-the-wild haptic data. Using data from 16 devices, we build the CLAMP dataset and train models for material and compliance recognition. Our models generalize to three different robot embodiments and enable real-world robot manipulation.

Abstract: Robust robot manipulation in unstructured environments often requires understanding object properties that extend beyond geometry, such as material or compliance—properties that can be challenging to infer using vision alone. Multimodal haptic sensing provides a promising avenue for inferring such properties, yet progress has been constrained by the lack of large, diverse, and realistic haptic datasets. In this work, we introduce the CLAMP device, a low-cost ($<$\$200) sensorized reacher-grabber designed to collect large-scale, multimodal, in-the-wild haptic data from non-expert users in everyday settings. We aggregate data from 16 CLAMP devices deployed to 41 users to create the CLAMP dataset, the largest open-source multimodal haptic dataset to date, comprising 12.3 million data points across 5357 household objects. Using this dataset, we train a haptic encoder that can infer object material and compliance from multimodal haptic data. We leverage this encoder to create the CLAMP model, a visuo-haptic perception model for material recognition that generalizes to novel objects and three robot embodiments with minimal finetuning. We also demonstrate the effectiveness of our model in three real-world robot manipulation tasks: sorting recyclable waste from trash, retrieving objects from a cluttered bag, and distinguishing overripe from ripe bananas. Our results show that large-scale, multimodal, in-the-wild haptic data collection can unlock new capabilities for generalizable robot manipulation. Device assembly guide, dataset, and code can be found here: https://emprise.cs.cornell.edu/clamp/

Keywords: Multimodal haptic perception, Data acquisition device, Datasets, Robot manipulation

1 Introduction

Robots benefit from knowledge of object properties beyond geometry, such as material and compliance, to manipulate objects in real-world scenarios such as sorting waste or grasping deformable objects [1, 2, 3]. However, it is challenging to reliably determine object material or compliance using vision alone [4, 5]. Haptic sensing helps in recognizing these object properties, both by itself [6, 7, 8, 9] and coupled with vision [10, 11], but it remains less utilized in robotics than vision or language. This is because state-of-the-art haptic datasets [12, 13, 14, 15] lack one or more of the following attributes:

1. **Multimodality**: Most state-of-the-art haptic datasets capture only contact force [12, 16, 17]. However, haptic sensing is inherently multimodal, including temperature [18, 19], force [20, 21], vibration [22, 23], and proprioception [24]. Prior work has shown that using more haptic modalities improve recognition of object properties such as instance [25] and compliance [15].
2. **Scalability**: Haptic data requires physical interaction, making data collection resource-intensive. Collecting haptic data from multiple users can enable large-scale datasets, but requires standardized sensing hardware that is easy to build and use.
3. **Diversity**: Most haptic datasets are collected in controlled environments with a curated list of objects [22, 14]. These datasets do not capture real-world haptic interactions during manipulation. Moreover, these datasets do not capture interactions with heterogeneous household objects, i.e., objects made of multiple materials (e.g. mobile phones, shoes, books with covers).
4. **Data from Grasping Interaction**: Haptic sensing is action-conditioned, i.e., it depends on the actions an agent takes while interacting with an object. Most datasets focus on non-prehensile interaction with one virtual finger [7, 26, 15]. As a result, their data differ from data generated during real-world manipulation involving grasping.

In this work, we **c**rowdsource a **la**rge, multimodal, in-the-wild haptic dataset for **m**ultimodal robot **p**erception. Our key insight is that *large-scale, multimodal, in-the-wild haptic data can help to create perception models that enable real-world robot manipulation for different embodiments.* Our contributions are as follows:

1. The **CLAMP device**, a low-cost, sensorized reacher-grabber that captures five haptic modalities: active thermal, passive thermal, force, vibration, and proprioceptive sensing. The device can store data, record image and speech annotations, and is designed for long-term deployment with non-expert users: it is lightweight (0.59 kg), portable, and contains a display with a GUI that allows users to collect data easily with clear instructions.
2. The **CLAMP dataset**, the largest open-source multimodal haptic dataset in the robotics literature — containing 12.3 million data samples from 5357 in-situ household objects gathered by 41 non-expert users with a total of 16 CLAMP devices. The CLAMP dataset contains data from a wide range of household objects with diversity in grasp orientation, speed, and force.
3. The **CLAMP model**, a visuo-haptic model for material classification that generalizes to novel objects and data from multiple robots and gripper types with minimal finetuning. Additionally, the haptic encoder in the CLAMP model can be transferred to classify objects based on compliance with no finetuning.
4. **Real robot experiments** that demonstrate a finetuned CLAMP model with a 7-DoF Franka Panda in three real-world robot manipulation tasks: sorting waste into trash and recycling, where the robot reasons about object material in cluttered scenes, retrieving objects from a cluttered bag, where it reasons about object material under occlusion and visual ambiguity, and separating ripe from overripe bananas, where it reasons about object compliance among objects of the same instance.

2 Related Work

Data acquisition in haptics. Prior works in object instance recognition and material recognition have collected haptic data from robots [7, 14, 17, 28, 29] or human experts [12, 14, 26], however

Dataset	Object inst.	Touches	Data Samples	Source	Modalities
Penn Haptic Texture Toolkit [22]	100	200	*	Human (Author)	3
The Feeling of Success [17]	106	9.3k	*	Robot	2
TVL [16]	*	*	44k	Human (Author)	3
Open Access Haptic Database [15]	47	1340	*	Human (Author)	5
Touch100k [13]	*	*	100k	Human (Derived)	3
Penn Haptic Adjective Toolkit [7]	60	600	*	Robot	5
SSVTP [14]	*	*	4500	Robot	2
Proton [27]	357	1.1k	*	Human (Author)	3
Touch and Go [12]	3971	13.9k	*	Human (Author)	2
CLAMP Dataset (Ours)	**5357**	**25.1k**	**12.3M**	Human (Crowdsourced)	**7**

Table 1: Comparison of the CLAMP dataset with various existing datasets and their characteristics. The CLAMP dataset is the largest open-source haptic dataset to date in terms of data samples and modalities (5 haptic + vision and language) (* indicates unreported figure in source).

this approach is constrained by the availability of robots and human experts. While haptic sensing rigs [30, 31, 16] provide a pathway to scalable data collection, existing rigs are not optimized for size [30], user-friendliness [31], or cost [16], and also lack streamlined workflows for data collection. In contrast, the CLAMP device is designed for non-experts to use in their homes.

Haptic datasets. State-of-the-art haptic datasets [12, 13, 14, 15, 7, 32, 33, 34, 25, 17, 28] contain fewer data samples than those in vision [35], language [36] or robot actions [37]. Previously, Yang et al. [12] created the largest haptic dataset by number of objects, consisting of 3971 objects, while Cheng et al. [13] aggregated the largest dataset by data points with 100k object touches. In comparison, the CLAMP dataset contains 5357 household objects and 12.3 million data samples, representing a significant increase in data scale.

Most datasets are collected using a curated set of materials [18, 38, 19] or objects [15, 33, 17], typically by robots or human experts who ensure uniform interaction and consistent contact duration [15, 12, 33, 17, 39, 7]. In comparison, our data collection allows users to collect data from any graspable household object, enabling diverse, in-situ haptic data. Finally, prior datasets capture some haptic modalities among force [20, 40, 12, 26, 16], vibration [23], thermal [18, 41] sensing, or a combination of the above [15, 7, 32, 33, 34]. In comparison, our dataset captures a total of 7 modalities, including 5 haptic modalities along with vision and structured language.

Object property recognition. Real-world robot manipulation can benefit from recognizing object properties in the wild [12]. Recognizing object material from vision alone is challenging without more information [42, 4]. Recent works that ground vision–language models in object properties [43, 44] also highlight multimodal perception as the next step. While many works have used haptic recognition models to identify various object properties [10, 7, 21], these models are trained on data from controlled settings. Yang et al. [12] train a material classification model on unimodal, in-the-wild haptic data, specifically high-resolution force data, collected from non-prehensile actions. In comparison, we train material and compliance classification models on multimodal, in-the-wild haptic data collected from grasping actions, enabling recognition for real-world robot manipulation.

Haptic data is *action-conditioned*, which makes transferring haptic recognition models across robots challenging. As a result, most work on object property recognition from visuo-haptic sensing [10, 25] pretrain or finetune their models on data from only one robot. Yu et al. [26] show that their tactile model, pretrained on human-collected haptic data, transfers to classification tasks on robot data, but does not show generalization to multiple robots. Tatiya et al. [45] learn a common latent space to transfer haptic knowledge between robots but validate their methods on simulated robots only. In comparison, we show that our visuo-haptic model, pretrained on human-collected data, transfers to three robot embodiments with varying gripper embodiments.

3 The CLAMP Device

Sensing. The CLAMP device features a sensorized reacher-grabber that captures a total of five modalities upon contact with an object. The left sensor bed includes a passive unheated thermal sensor that records the ambient temperature of the object, and an active thermal sensor heated to 55

°C, whose response over time during contact reflects the object's heat capacity. The suction cup on the right end-effector houses a contact microphone that measures vibrations, a force-sensing resistor that measures normal force, and a six-axis IMU. These sensors are embedded in 3D printed sensor beds as shown in Figure 2. The device also includes a second IMU, mounted on the device body frame, which helps to isolate the motion of the end-effector.

Compute and storage. Our device integrates all sensors with a Raspberry Pi Pico microcontroller (referred to as Pico) and a Raspberry Pi Zero 2W (referred to as SBC, or single-board computer). The Pico is mounted on a custom printed circuit board (PCB) and samples at a rate of 50 Hz for all sensors except the contact microphone, which is sampled at 100 Hz. Data is transferred to the SBC via a Universal Asynchronous Receiver–Transmitter (UART) interface. All data is stored on-board and retrieved when the device is returned after deployment.

User interface. The SBC is connected to an Adafruit 2.2" PiTFT Display. The SBC takes input from four buttons on the PiTFT and powers the display using PyGame. On booting, the computer loads a graphical user interface (GUI) that allows the user to interact with the device and guides them through the data collection process. Two LEDs, one for each suction cup, provide users with real-time feedback about contact between object surface and each cup, based on sensor readings from the active thermal sensor (right cup) and the force sensor (left cup).

Design. The device's long handle allows users with limited reach to grasp objects in-situ. We mount the lighter Pico frame along the device length and the heavier SBC enclosure close to the handle, to keep the device ergonomic. As a result, users noted that they held the device in one hand as they repositioned objects with their free hand.

Design principles. We designed the device to be:

- "Easy to Build": The CLAMP device can be assembled from scratch within 5 hours, which is on par with other open-source robotics hardware [46, 47]. The CLAMP device uses readily available sensors sold with datasheets. We also release all CAD files and assembly instructions.
- "Easy to Carry": The CLAMP device is lightweight (0.59 kg), allowing users to take the device home. We achieved this by selecting the lightweight Pi Zero 2W as our onboard computer and using compact end-effectors that keep the center of gravity of the device close to the handle.
- "Easy to Use": Users grasp objects with the reacher-grabber, receive contact feedback via LEDs, and follow on-screen prompts through the GUI to collect data and annotations. Buttons on the PiTFT allow users to track progress and power down the device when not in use.
- "Easy to Scale": One CLAMP device costs $< \$200$ to build, making it more affordable than existing haptic sensors [48, 49, 50] and sensing suites [51]. The sensorized suction cups can be mounted on robot end-effectors, allowing roboticists to collect haptic data with their own robots.

4 The CLAMP Dataset

The **CLAMP dataset** is the largest open-source multimodal haptic dataset to date. The dataset consists of 5357 object instances, 25.1k object touches, and 12.3M individual data samples. We gathered this data from 16 CLAMP devices deployed to 41 users.

Data collection process. Each user is given a short tutorial and a manual on how to use the device. The GUI guides users through three steps of data collection for an object: capturing an image of the object, recording a spoken annotation, and collecting haptic data by grasping the object for up to 10 seconds per trial, for five trials. The GUI then returns to the start screen. At the end of the deployment, the device is returned for maintenance and data backup.

Data post-processing and featurization. We collect data from six haptic sensors during each trial: active and passive thermal sensors, a force sensor, a contact microphone, and two IMUs. After synchronizing all sensor data streams, we apply smoothing filters and extract features from each sensor, resulting in a total of nine features. Refer to Appendix B.1 for more details. We synchronize contacts for each suction cup by detecting contact onset and loss for the left and right cups separately.
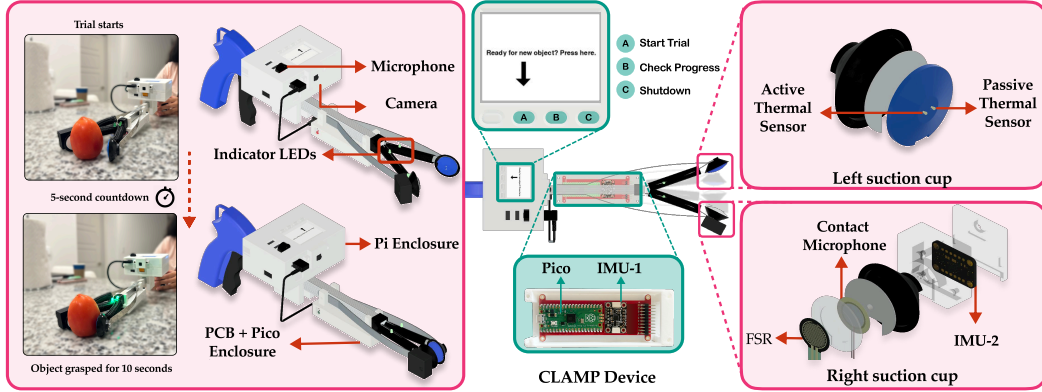
Figure 2: **Device overview:** The CLAMP device features a modified reacher-grabber equipped with sensors that capture five haptic modalities. Our device enables non-expert users to collect haptic data in their homes.

We elaborate on the rule-based approach used to detect contact onset and release for each cup, in Appendix B.2. Finally, we pad the features to ensure all features are of the same length.

Object property annotations. We generate annotations for object material and compliance to enable model learning with the CLAMP dataset. In particular, each object is labeled with:

1. One of 16 **material labels**: {aluminium, brass, cardboard, dry wall, fabric, foam, glass, granite, hard plastic, paper, porcelain, rubber, soft plastic, steel, vegetable matter, wood}. We create this list from 300 material subcategories from the CES Edupack [52] database that appear in our dataset, which we group by similarity towards downstream manipulation. Our material list covers 93% of objects in our dataset.
2. If the object has different materials on opposing surfaces, i.e., **heterogeneous surfaces**: {yes, no}. Note the difference between these objects (e.g. mobile phones) and *heterogeneous objects* (multiple materials overall, e.g. furniture).
3. An adjective label describing **object compliance**: {soft, hard}.

To generate these labels, we first transcribe the audio annotation using Whisper [53]. We then provide the audio transcription along with the image annotation for the object to GPT-4o [54], along with more than 12 in-context examples. We detail the exact prompts used in Appendix B.4. Finally, expert annotators verify the audio transcription and material labels and re-run the annotation pipeline for any corrections made.

Dataset statistics. The CLAMP dataset contains diversity in objects and in grasping actions. In the appendix, we quantify diversity across three axes: materials, grasping forces, and grasping speeds. Refer to Appendix B.5 for more details.

5   The CLAMP Model

The **CLAMP model** is a multimodal perception model that recognizes object material from visual and multimodal in-the-wild haptic data. The model fuses a haptic encoder trained on the CLAMP dataset with a pretrained visual encoder.

Haptic Encoder. Our haptic encoder is a six-block InceptionTime model [55], followed by a global average pooling layer, and finally a three-layer MLP as the classification head. We split the CLAMP dataset sample-wise, train the encoder over three random seeds, and use the worst-performing model for comparisons. See Appendix C.1 for more details.

Visual Encoder. We use GPT-4o [54] as a low-dimensional visual encoder. Given an input image, audio transcription, and in-context examples, it outputs per-token probabilities, which we filter and normalize into a categorical distribution over material classes. See Appendix C.2 for more details.
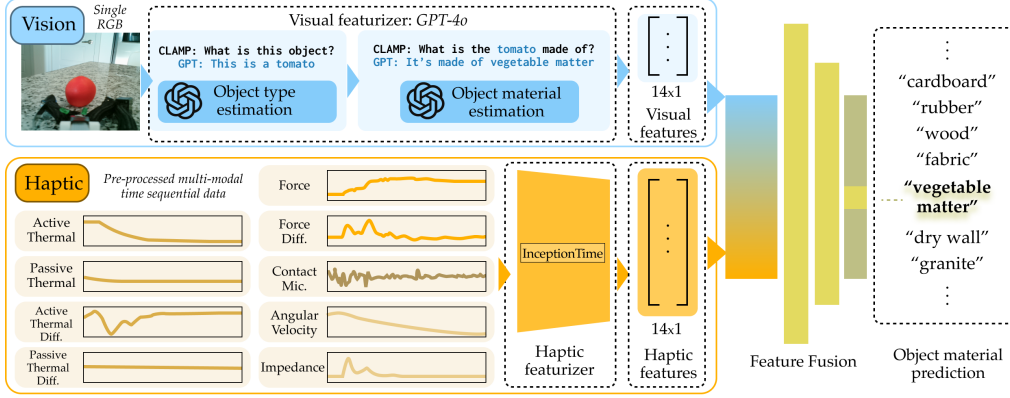
Figure 3: **Model overview:** We propose the CLAMP model, a visuo-haptic model that fuses outputs from a GPT-4o [54] visual encoder and a pretrained InceptionTime-based [55] haptic encoder.

The CLAMP Model: Combining Haptics and Vision. The CLAMP model uses a 2-layer MLP to fuse the outputs of the haptic and visual encoders. To train the model, we freeze the pretrained haptic encoder and minimize a loss function: $\mathcal{L} = \mathcal{L}_{WCE} + \lambda_{KL} \cdot \mathcal{L}_{KL}(\mathcal{V}||\mathcal{P})$, where $\mathcal{L}_{WCE}$ is a weighted cross-entropy loss, $\mathcal{L}_{KL}(\mathcal{V}||\mathcal{P})$ is a KL-divergence loss term with the vision probabilities $\mathcal{V}$ and the visuo-haptic model probabilities $\mathcal{P}$, and λ_{KL} is the weight for KL loss (Appendix C.1).

Comparison to Pretrained Vision Foundation Models. We compare the CLAMP model on material classification performance to GPT-4o [54], a grounded vision-language model (PG-VLM [43]), and a CLIP [56] model finetuned on our dataset. For each model, we consider three inputs: raw images, images cropped to the object instance using Grounding DINO [57], and images segmented by Grounded SAM [58]. We compare the CLAMP model with the three vision-only models, across all inputs, in Table 2, which reports test-set accuracy and normalized Matthews correlation coefficient (nMCC). The CLAMP model achieves the best performance, suggesting that the background knowledge of the pretrained models cannot compensate for their lack of haptic input.

Haptic Modality Ablation Studies. We compare classification performance of the haptic encoder to ablated encoders trained on all but one haptic modality. In Table 2, we see that the encoder trained on all modalities performs best, suggesting the importance of each modality. All five modalities provide complementary cues about object properties. Force conditioned on proprioception distinguishes soft objects from hard ones [15], vibration sensing captures high-frequency contact cues in hard objects, and active/passive thermal sensing separates objects of similar compliance by thermal conductivity and temperature [41]. Finally, we see that the CLAMP model performs better than the haptic encoder alone, confirming that both vision and haptics are necessary for material recognition.

Alternative Haptic Encoders. We consider alternative network architectures for the haptic encoder. We first compare our InceptionTime encoder to a Random Forest encoder used in previous work on haptic classification [59], finding InceptionTime to obtain better performance (Table 2).

Visuo-Haptic Fusion Strategy. We explore finetuning CLIP and fusing it with the haptic encoder. While CLIP underperforms in comparison with GPT-4o in Table 2, it enables learning a visuo-haptic representation with high-dimensional embeddings. We compare the CLAMP model to a setup where a finetuned CLIP encoder is fused with the haptic encoder, using high-dimensional embeddings from both. The CLAMP model outperforms the CLIP-based fusion, suggesting that fusing higher-dimensional features does not necessarily lead to better performance.

Testing on Held-out Objects. We train a new CLAMP model on a dataset split into training, validation, and test sets by objects rather than by contact instances. The CLAMP model outperforms the vision-only baseline on the new test set, suggesting that our model can infer object material when tested on novel objects. The CLAMP model's 3% accuracy improvement over the vision-only baseline is statistically significant with $p < 0.05$ (using the Wilcoxon-Signed Rank test).

Haptic Models

Method	Acc.	nMCC
InceptionTime	**0.59 ± 0.01**	**0.76 ± 0.006**
Random Forest	0.52 ± 0.001	0.71 ± 0.001

Modality-specific Ablation

Absent Modality	Acc.	nMCC
None	**0.58**	**0.76**
Active Therm.	0.49	0.7
Passive Therm.	0.46	0.68
Force	0.22	0.57
Vibration	0.57	0.75
Proprioception	0.54	0.74

Vision Models

Method	Acc.	nMCC
GPT-4o (raw)	**0.65**	**0.80**
GPT-4o (segmented)	0.59	0.77
GPT-4o (cropped)	0.64	0.80
CLIP (raw)	**0.58**	**0.75**
CLIP (segmented)	0.56	0.74
CLIP (cropped)	0.57	0.74
PG-VLM (raw)	0.39	0.67
PG-VLM (segmented)	0.44	0.69
PG-VLM (cropped)	**0.58**	**0.76**

Visuo-haptic models

Method	Acc.	nMCC
CLAMP model	**0.87**	**0.93**
CLIP + Haptic encoder	0.79	0.89

Evaluation on held-out objects

Method	Acc.	nMCC
CLAMP model	**0.73**	**0.85**
GPT-only	0.70	0.83

Table 2: We report the performance of material recognition models on the CLAMP dataset. Note that for "Haptic Models", metrics are reported as mean ± stddev.

Transferring the Haptic Encoder to Compliance Recognition. We show that our haptic encoder, which was trained for material classification, can be transferred to recognize object compliance. We train a new classification head on the frozen haptic encoder using object compliance recognition labels (soft and hard) from the CLAMP dataset. The haptic encoder achieves an accuracy of 0.88 on the test set and an nMCC score of 0.869. This result suggests that our haptic encoder learns representations that are applicable beyond material recognition.

Analyzing CLAMP Model Uncertainty. Uncertainty quantification is important for robot manipulation tasks to reduce failures [60]. To assess whether CLAMP model predictions carry uncertainty signals, we mark a prediction as *uncertain* if the corresponding softmax value is less than a threshold p_1, or if the difference between the two largest softmax values is less than a threshold p_2. Filtering predictions increases the model test accuracy from 0.73 to 0.82, and the nMCC from 0.85 to 0.90, while retaining 75% of predictions. Refer to Appendix C.1 for values of p_1 and p_2.
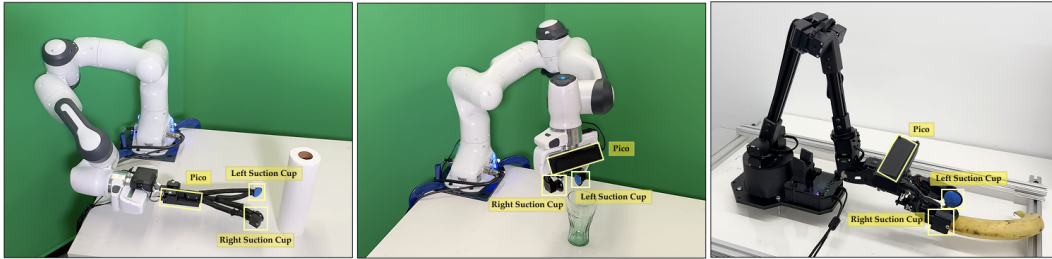
6 Demonstrating the CLAMP Model on Real Robots

Figure 4: We mount haptic sensors from the CLAMP device on three robot embodiments and collect haptic data. **Left.** Franka with CLAMP device. **Middle.** Franka with Parallel Jaw. **Right.** WidowX with Parallel Jaw.

In this section, we demonstrate that the CLAMP model enables multimodal robot perception for multiple embodiments and thus improves robot manipulation capabilities in real-world scenarios.

Finetuning the CLAMP model on robot data. We collect haptic data from three different robot embodiments, and show that a pretrained CLAMP model requires little finetuning data to exceed the performance of vision-only methods towards material recognition. We consider three embodiments:

1. Franka with CLAMP Device: We mount a shortened CLAMP device on the end effector of the Franka Panda. The device is actuated by tensioning its central steel strips, which are connected to a pulley system driven by a pair of interlocked Dynamixel XC330-M288-T motors. A 3D printed housing contains the motors and the Pico.
2. Franka with Parallel Jaw: We directly attach the suction cups of the CLAMP device to the fingers of the Franka arm using 3D printed mounts. An enclosure on the Franka Hand houses the Pico.

3. <u>WidowX with Parallel Jaw</u>: We similarly directly attach the suction cups of the CLAMP device to the fingers of a WidowX 250s arm using 3D printed mounts. An enclosure houses the Pico.

The narrow grasping width for the two parallel jaw embodiments prevents grasps for several household objects, resulting in a smaller dataset for these embodiments compared to that for the first one. We post-process and featurize the collected haptic data following the steps described in Section 4. However, for embodiments with a parallel jaw gripper, we modify the impedance feature calculation to use linear acceleration instead of angular velocity. Refer to Appendix D.1 for details.

We finetune a pretrained CLAMP model with varying amounts (7%, 15%, and 30% data) of embodiment-specific haptic data and offline images collected with a CLAMP device. Table 3 shows that using more finetuning data improves model performance, and that finetuned models for all three embodiments perform better than vision-based material predictions with 15% of finetuning data.

Franka with CLAMP device			Franka with Parallel Jaw			WidowX with Parallel Jaw		
Finetuning Data (%)	Acc.	nMCC	Finetuning Data (%)	Acc.	nMCC	Finetuning Data (%)	Acc.	nMCC
Vision-only	0.63	0.80	*Vision-only*	0.77	0.87	*Vision-only*	0.68	0.83
Zero-shot	0.55	0.76	7%	0.75	0.86	7%	0.71	0.83
7%	0.73	0.85	15%	0.93	0.96	15%	0.71	0.84
15%	0.77	0.88	30%	0.95	0.97	30%	0.81	0.89
30%	0.82	0.90						

Table 3: We report the performance of the CLAMP model finetuned on robot-collected data for three robot embodiments, for varying amounts of finetuning data.

Leveraging multimodal perception for real-world manipulation. We finetune a CLAMP model with 85% of the finetuning dataset for the "Franka with Parallel Jaw" embodiment equipped with a wrist camera, and deploy it in three real-world scenarios.
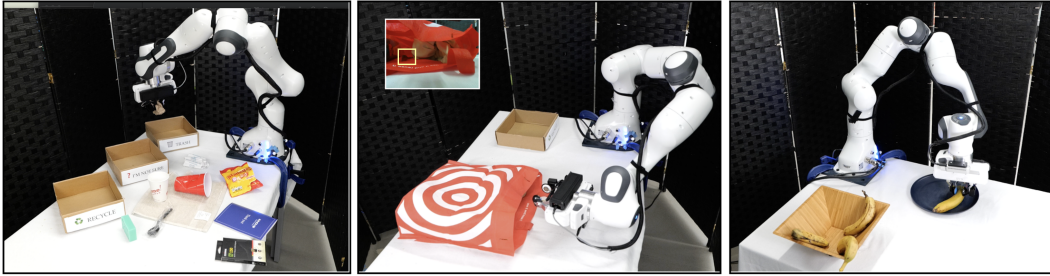

Figure 5: We demonstrate the CLAMP model in three real-world robot manipulation scenarios. **Left.** The robot sorts recyclables from non-recyclables. **Middle.** The robot retrieves a metallic object (in yellow square) from a bag with multiple objects. **Right.** The robot distinguishes overripe from ripe bananas.

- <u>Sorting recyclable waste from trash</u>: In this setting, the robot disposes of 10 objects not seen in the finetuning dataset, into "Trash" or "Recycle" bins based on material predictions from the CLAMP model (sorting rules detailed in Appendix D.3), with an "I'm not sure" box for uncertain predictions. The robot disposes of objects in the correct bin 9/10 times per trial, while generating unknown predictions 0.66/10 times per trial, over three trials. This shows that our model can accurately recognize material for unseen objects in cluttered environments.
- <u>Retrieving objects from a cluttered bag</u>: In this setting, the robot interacts with objects inside a cloth bag, and must retrieve the object only if it is metallic. The bag contains three unseen objects, one of which is metallic. We also log predictions from the vision-only model for comparison. The finetuned CLAMP model retrieved the metallic object in 6 out of 13 trials, while the vision model does not find a metallic object in any of the 13 trials. We thus show that haptics provides key cues in visually challenging situations. See Appendix D.4 for more details.
- <u>Sorting ripe from overripe bananas</u>: In this setting, the robot interacts with four bananas (two of which are overripe), predicts its compliance, and retrieves only ripe bananas. Our haptic encoder transferred to compliance prediction and finetuned on robot data achieves an accuracy of 0.833 and an nMCC of 0.86. This shows that our model can differentiate between visually ambiguous objects of the same instance, based on object compliance.

7 Limitations and Future Work

In this work, we present the CLAMP device, dataset, and model, which collectively represent a significant step towards scaling visuo-haptic perception for robotics. We conclude with a discussion of limitations in the current framework and opportunities for future work.

CLAMP Device Design Constraints. The CLAMP device was optimized for weight, cost and portability. This lightweight design came with trade-offs. For example, we opted against using a more powerful microprocessor like the Raspberry Pi 4, which would have enabled higher bandwidth data collection such as video streams, high-frequency contact microphone data, and vision-based tactile sensing. In addition, the current suction cup design allows for agile handling, but its low surface area and convex sensor bed can hinder consistent contact with objects with curved surfaces.

CLAMP Dataset Limitations. While the CLAMP dataset is the largest haptic dataset to date, it is still far smaller than counterparts in vision and language. It remains to be seen how much data is required to train large visuo-haptic foundation models that have emergent generalization capabilities analogous to large language and vision-language models. Furthermore, while the CLAMP dataset features diversity along multiple axes, it does not have diversity in terms of sensors used for data collection. Future work involves combining our dataset with data from other sensors, such as vision-based tactile sensors [48].

CLAMP Model Limitations. The CLAMP model performs better than vision-only and haptic-only perception models, but there is much room for future work on learning better visuo-haptic models with the CLAMP dataset. A direction to explore in future work is training larger models that operate directly on the raw visuo-haptic inputs. The structure we imposed by featurizing the haptic data and prompting a VLM for visual encoding was helpful for the current CLAMP model, but this may change as data and compute continue to scale.

References

[1] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt. Modeling of deformable objects for robotic manipulation: A tutorial and review. *Frontiers in Robotics and AI*, 7:82, 2020.

[2] J. Bednarek, M. Bednarek, P. Kicki, and K. Walas. Robotic touch: Classification of materials for manipulation and walking. In *2019 2nd IEEE international conference on Soft Robotics (RoboSoft)*, pages 527–533. IEEE, 2019.

[3] J. Wei, S. Cui, J. Hu, P. Hao, S. Wang, and Z. Lou. Multimodal unknown surface material classification and its application to physical reasoning. *IEEE Transactions on Industrial Informatics*, 18(7):4406–4416, 2022. doi:10.1109/TII.2021.3126601.

[4] M. S. Drehwald, S. Eppel, J. Li, H. Hao, and A. Aspuru-Guzik. One-shot recognition of any material anywhere using contrastive learning with physics-based rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23524–23533, 2023.

[5] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges. *Representations, and Algorithms*, page 82, 2019.

[6] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, N. Fitter, J. C. Nappo, T. Darrell, et al. Using robotic exploratory procedures to learn the meaning of

haptic adjectives. In *2013 IEEE International Conference on Robotics and Automation*, pages 3048–3055. IEEE, 2013.

[7] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, T. Darrell, and K. J. Kuchenbecker. Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems*, 63:279–292, 2015.

[8] A. Murali, Y. Li, D. Gandhi, and A. Gupta. Learning to grasp without seeing. In *International Symposium on Experimental Robotics*, pages 375–386. Springer, 2018.

[9] H. Dang and P. K. Allen. Learning grasp stability. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2012.

[10] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell. Deep learning for tactile understanding from visual and haptic data. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 536–543. IEEE, 2016.

[11] T. N. Le, F. Verdoja, F. J. Abu-Dakka, and V. Kyrki. Probabilistic surface friction estimation based on visual and haptic measurements. *IEEE Robotics and Automation Letters*, 6(2):2838–2845, 2021. doi:10.1109/LRA.2021.3062585.

[12] F. Yang, C. Ma, J. Zhang, J. Zhu, W. Yuan, and A. Owens. Touch and go: Learning from human-collected vision and touch. *arXiv preprint arXiv:2211.12498*, 2022.

[13] N. Cheng, C. Guan, J. Gao, W. Wang, Y. Li, F. Meng, J. Zhou, B. Fang, J. Xu, and W. Han. Touch100k: A large-scale touch-language-vision dataset for touch-centric multimodal representation. *arXiv preprint arXiv:2406.03813*, 2024.

[14] J. Kerr, H. Huang, A. Wilcox, R. Hoque, J. Ichnowski, R. Calandra, and K. Goldberg. Self-supervised visuo-tactile pretraining to locate and follow garment features. *arXiv preprint arXiv:2209.13042*, 2022.

[15] T. Bhattacharjee, H. M. Clever, J. Wade, and C. C. Kemp. Multimodal tactile perception of objects in a real home. *IEEE Robotics and Automation Letters*, 3(3):2523–2530, 2018.

[16] L. Fu, G. Datta, H. Huang, W. C.-H. Panitch, J. Drake, J. Ortiz, M. Mukadam, M. Lambeta, R. Calandra, and K. Goldberg. A touch, vision, and language dataset for multimodal alignment. *arXiv preprint arXiv:2402.13232*, 2024.

[17] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017.

[18] T. Bhattacharjee, J. Wade, and C. C. Kemp. Material recognition from heat transfer given varying initial conditions and short-duration contact. In *Robotics: Science and Systems*, volume 2015, 2015.

[19] T. Bhattacharjee, J. Wade, Y. Chitalia, and C. C. Kemp. Data-driven thermal recognition of contact with people and objects. In *2016 IEEE Haptics Symposium (HAPTICS)*, pages 297–304. IEEE, 2016.

[20] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp. Haptic classification and recognition of objects using a tactile sensing forearm. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4090–4097. IEEE, 2012.

[21] A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic. Classification of rigid and deformable objects using a novel tactile sensor. In *2011 15th International Conference on Advanced Robotics (ICAR)*, pages 427–434, 2011. doi:10.1109/ICAR.2011.6088622.

[22] H. Culbertson, J. L. Delgado, and K. J. Kuchenbecker. The penn haptic texture toolkit for modeling, rendering, and evaluating haptic virtual textures. *Departmental Papers (MEAM)*, 299, 2014.

[23] J. Sinapov, V. Sukhoy, R. Sahai, and A. Stoytchev. Vibrotactile recognition and categorization of surfaces by a humanoid robot. *IEEE Transactions on Robotics*, 27(3):488–497, 2011.

[24] J. Sinapov, C. Schenck, and A. Stoytchev. Learning relational object categories using behavioral exploration and multimodal perception. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5691–5698, 2014. doi:10.1109/ICRA.2014.6907696.

[25] Z. Erickson, S. Chernova, and C. C. Kemp. Semi-supervised haptic material recognition for robots using generative adversarial networks. In *Conference on Robot Learning*, pages 157–166. PMLR, 2017.

[26] S. Yu, K. Lin, A. Xiao, J. Duan, and H. Soh. Octopi: Object property reasoning with large tactile-language models. *arXiv preprint arXiv:2405.02794*, 2024.

[27] A. L. Burka. *Instrumentation, data, and algorithms for visually understanding haptic surface properties*. PhD thesis, University of Pennsylvania, 2018.

[28] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018.

[29] J. M. Romano and K. J. Kuchenbecker. Methods for robotic tool-mediated haptic surface recognition. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 49–56, 2014. doi:10.1109/HAPTICS.2014.6775432.

[30] A. Burka, S. Hu, S. Helgeson, S. Krishnan, Y. Gao, L. A. Hendricks, T. Darrell, and K. J. Kuchenbecker. Proton: A visuo-haptic data acquisition system for robotic learning of surface properties. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 58–65, 2016. doi:10.1109/MFI.2016.7849467.

[31] J. Wade, T. Bhattacharjee, and C. C. Kemp. A handheld device for the in situ acquisition of multimodal tactile sensing data, 2015. URL https://arxiv.org/abs/1511.03152.

[32] M. Kerzel, E. Strahl, C. Gaede, E. Gasanov, and S. Wermter. Neuro-robotic haptic object classification by active exploration on a novel dataset. In *2019 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2019.

[33] L. E. R. Bonner, D. D. Buhl, K. Kristensen, and N. Navarro-Guerrero. Au dataset for visuo-haptic object recognition for robots. *arXiv preprint arXiv:2112.13761*, 2021.

[34] S. Toprak, N. Navarro-Guerrero, and S. Wermter. Evaluating integration strategies for visuo-haptic object recognition. *Cognitive computation*, 10:408–425, 2018.

[35] B. Wu, W. Chen, Y. Fan, Y. Zhang, J. Hou, J. Liu, and T. Zhang. Tencent ml-images: A large-scale multi-label image database for visual representation learning. *IEEE access*, 7: 172683–172693, 2019.

[36] P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for squad, 2018. URL https://arxiv.org/abs/1806.03822.

[37] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[38] M. Strese, J.-Y. Lee, C. Schuwerk, Q. Han, H.-G. Kim, and E. Steinbach. A haptic texture database for tool-mediated texture recognition and classification. In *2014 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE) Proceedings*, pages 118–123, 2014. doi:10.1109/HAVE.2014.6954342.

[39] Y. Li, J.-Y. Zhu, R. Tedrake, and A. Torralba. Connecting touch and vision via cross-modal prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10609–10618, 2019.

[40] T. Bhattacharjee, A. Kapusta, J. M. Rehg, and C. C. Kemp. Rapid categorization of object properties from incidental contact with a tactile sensing robot arm. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 219–226. IEEE, 2013.

[41] H. Bai, H. Chen, E. Healy, C. C. Kemp, and T. Bhattacharjee. Analyzing material recognition performance of thermal tactile sensing using a large materials database and a real robot. In *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2255–2262. IEEE, 2022.

[42] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3479–3487, 2015.

[43] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh. Physically grounded vision-language models for robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12462–12469. IEEE, 2024.

[44] A. Agrawal, R. Prabhakar, A. Goyal, and D. Liu. Physical reasoning and object planning for household embodied agents. *arXiv preprint arXiv:2311.13577*, 2023.

[45] G. Tatiya, Y. Shukla, M. Edegware, and J. Sinapov. Haptic knowledge transfer between heterogeneous robots using kernel manifold alignment. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5358–5363. IEEE, 2020.

[46] A. Zorin, I. Guzey, B. Yan, A. Iyer, L. Kondrich, N. X. Bhattasali, and L. Pinto. Ruka: Rethinking the design of humanoid hands with learning. *arXiv preprint arXiv:2504.13165*, 2025.

[47] K. Shaw, A. Agarwal, and D. Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[48] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, D. Jayaraman, and R. Calandra. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, July 2020. ISSN 2377-3774. doi:10.1109/lra.2020.2977257. URL http://dx.doi.org/10.1109/LRA.2020.2977257.

[49] M. Lambeta, T. Wu, A. Sengul, V. R. Most, N. Black, K. Sawyer, R. Mercado, H. Qi, A. Sohn, B. Taylor, et al. Digitizing touch with an artificial multimodal fingertip. *arXiv preprint arXiv:2411.02479*, 2024.

[50] W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12), 2017.

[51] S. Clarke, S. Wistreich, Y. Ze, and J. Wu. X-capture: An open-source portable device for multi-sensory learning, 2025. URL https://arxiv.org/abs/2504.02318.

[52] Ansys Granta. CES EduPack 2023. Software, 2023. Available: https://www.ansys.com/products/materials/granta-edupack.

[53] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[54] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, and et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[55] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[56] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[57] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2025.

[58] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[59] V. P. da Fonseca, X. Jiang, E. M. Petriu, and T. E. A. de Oliveira. Tactile object recognition in early phases of grasping using underactuated robotic hands. *Intelligent Service Robotics*, 15 (4):513–525, 2022.

[60] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[61] I. Oguiza. tsai - a state-of-the-art deep learning library for time series and sequential data. Github, 2023. URL https://github.com/timeseriesAI/tsai.

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 kΩ B57541G1103F NTC thermistors for active and passive thermal sensing. 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 150 N. We calibrate the force-sensing resistor with an FX29 load cell. The resulting calibration curve is described by an exponential function of the voltage, achieving an R2 = 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 align all sensor data (from the active and passive thermal sensor, the force sensor, the contact microphone, and the two IMUs) based on their timestamps, ensuring that readings are synchronized to within 2 ms.

2. We convert raw voltage readings from the active and passive thermal sensors to °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 extract a total of nine features from all sensor data streams. For the thermal sensors, the force sensor, and the contact microphone, we use the smoothed readings directly. For the IMU data, we compute the angular velocity of the end-effector IMU relative to the

---

[1]Despite 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 a material recognition model trained on calibrated force values performs equally as well as one trained on uncalibrated force values.

IMU in the gripper body. This results in five primary features. We then create four additional features: differences in active thermal, passive thermal, and force readings between consecutive timesteps, and an impedance feature. Note that:

(a) For angular velocity feature, we compute the gyroscope readings of IMU-2 relative to motion of IMU-1, in the frame of IMU-1. We consider the relative angular velocity of IMU-2 about the Z-axis of IMU-1 as a feature.

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

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

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 $3°/s$ for our experiments. We find that this impedance feature reliably indicates contact and shows distinct responses for contact with rigid and compliant objects. 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 without releasing their grasp, resulting in a sudden change in force measurement from the FSR.

5. We apply a causal moving average filter on each feature except the contact microphone. The contact microphone readings are debiased and then downsampled to match the length of the other features.

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 <u>active thermal sensor readings</u>. The cup is assumed to not be in contact at the initial timestep. Contact is detected when the cup was not in contact on the previous timestep, and the active thermal difference reading is below -0.01 °C. We determine that contact is released when the sensor was in contact at the previous timestep, the active thermal rate is positive, and the temperature is below 53 °C. The condition on active thermal reading prevents false triggers caused by the on-off controller that maintains the temperature of the active thermal sensor.

- Contact on the right cup is determined by the <u>force sensor readings</u>. The cup is assumed to not be in contact at the initial timestep. 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 0.1V.

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 $51\,°C$. Prior work has shown that material recognition is challenging under varying initial conditions [18].

- Contact instances where one of the temperature sensors shows erroneous readings. This occurs when the thermal sensors disconnect during deployment. We do not discard this data point from our dataset as data from other haptic modalities remain usable towards recognition.

- Contact instances with maximum angular velocity below a threshold (chosen as $1°/s$, indicating a very slow grasp.

- Contact instances with objects that have heterogeneous surfaces.

B.4 Data annotation

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

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

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

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

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

```
Input audio transcription : It's a painted stainless steel cup.
Image : <encoded image>
--------------------------------------------------------------------------
1. Object: cup 2. Materials: steel 3. heterogeneous Surfaces: No
```

B.5 Dataset statistics

We demonstrate the diversity in the CLAMP dataset along three axes: distribution of object materials, grasping forces, and grasping speeds. In Figure 6, we show the spread in values of maximum force, maximum grasping speed, along with object material. Additionally, we plot the duration of contact instances (before padding) as a histogram.
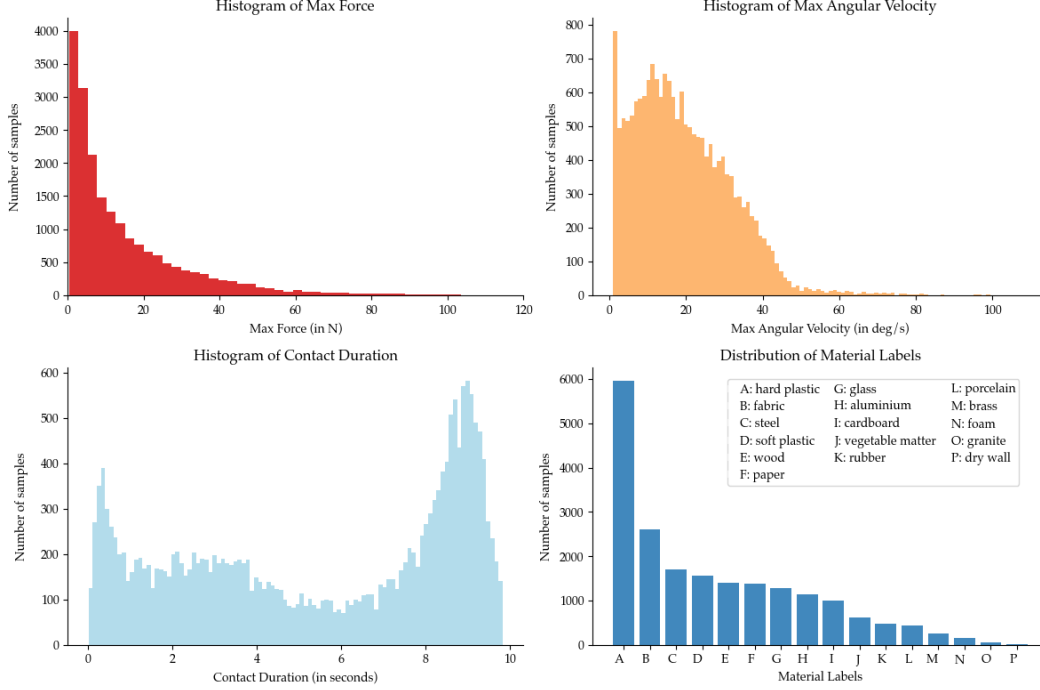
Figure 6: Visualizing the diversity in the CLAMP Dataset along various axes

For the material distribution, we only consider objects that do not have heterogeneous surfaces. Our dataset contains 5357 objects, out of which 680 ($\sim 13\%$) objects are objects with heterogeneous surfaces. We plot the distribution of material labels for contact instances with the remaining homogeneous objects. Figure 6 shows that the material labels contain significant class imbalance, with the largest class size (`hard plastic`) containing approximately 1000x as many samples as the smallest class (`dry wall`).

C The CLAMP model

C.1 Model training details

Hyperparameters and code details. The hyperparameters for the three experiments: haptic encoder training, CLAMP model pretraining, and CLAMP model finetuning on robot data, are provided in Table 4. The training code for the haptic encoder was written using the TSAI [61] API, while that for the CLAMP model pretraining and finetuning was written using PyTorch API.

Haptic encoder details. We use the InceptionTime architecture [55], choosing the number of filters as 256. Our model uses 7 1-dimensional convolutional kernels, the smallest and largest ones being of length 7 and 250 respectively, that are applied on each latent feature. These components make up one of the 6 Inception blocks in our model. The 6 blocks are followed by a 3-layer MLP that takes a latent embedding of dimension 2048 as input, and returns logits of dimension 14 as output.

Label weights. We use label weights from the CLAMP pretraining dataset to train the haptic encoder and then the CLAMP model. Since the robot-collected finetuning set has a different material distribution, we recalculate weights as the inverse class frequencies of the finetuning data instead of reusing the pretraining weights, in our finetuning experiments.

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

17

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 4: Hyperparameters for learning experiments

Loss function used for training CLAMP model. For the KL-divergence loss, we observe empirically that treating the vision distribution as the target and visuo-haptic as the student performs better than vice-versa.

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.

Experiments involving CLIP encoder. We use the ViT-B32 version of the CLIP model for our experiments. We finetune the last two layers of the model and attach a classification head for the vision-only experiments. For the visuo-haptic model with CLIP as the vision head, we concatenate a latent embedding of dimension 512 from CLIP, with the embedding of dimension 2048 from the haptic encoder (from the layer before the haptic encoder classification head), and then pass it to the MLP that fuses these visual and haptic embeddings to generate a material prediction.

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 rejected predictions.

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.

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

For the classes for which a log-probability was generated, we apply the softmax function and apply the fourth root. This step ensures a wider spread of probabilities across material classes. Finally, we normalize the transformed probabilities to generate logits with zero mean and unit standard deviation.

D Real robot experiments

D.1 Feature extraction for parallel jaw gripper embodiments

Feature extraction for data from the Franka with CLAMP device embodiment is identical to that for data from CLAMP devices. Feature extraction for the other two robot embodiments differs only in the proprioception and impedance features. We use the linear acceleration of IMU-2 relative to IMU-1 along IMU-2's Z-axis and process it to remove noise. First, we smooth the signal using a Savitzky-Golay filter. Then, we debias it using the mean of the last 50 timesteps, as the robot gripper is stationary near the end of data collection. Finally, we model and compensate for linear drift using readings from the first and last timestamps. This yields the proprioception feature for the parallel jaw gripper.

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 5: Rules for trash sorting

For impedance, we use the ratio of the force difference to linear acceleration feature. A lower threshold is not needed as the robots grasp objects with predefined speeds.

D.2 Data collection for robot embodiments

For all real-robot experiments, we assume the grasp of each object. For the sorting and metallic object retrieval experiments, each object also has a pre-defined pose from where an image is captured. For objects that are unstable (e.g. cello tape), we hold them or weakly attach them to the table using adhesive. This is not dissimilar to human data collection as users report holding the object with one hand while grasping using the CLAMP device in the other hand. Finally, for each object, we capture an image using the CLAMP device camera, to create the dataset.

The dataset for the "Franka with CLAMP device" embodiment is 60 objects, while datasets for the parallel jaw embodiments is 30 objects each. Comparing finetuned model performance across embodiments, we hypothesize that "Franka with CLAMP" underperforms "Franka with Parallel Jaw" because it includes data from wide objects that the parallel jaw embodiment cannot grasp. Because the robot grasping policy was open-loop, this led to variable and noisy contact between the CLAMP device grippers and these wider objects (wider objects led to inconsistent and partial surface area contact between sensors and object surfaces). This was not the case during human data collection because humans used the LED signals to close the loop on their grasping policy such that the contact was complete and consistent. To test our hypothesis, we finetune the CLAMP model on 30% data out of all data composed of only objects grasped by all embodiments. With 0.96 test accuracy, "Franka with CLAMP" slightly outperforms "Franka with Parallel Jaw" (0.95) and "WidowX with Parallel Jaw" (0.90), which better reflects the effect of device and various robot embodiment differences.

D.3 Sorting recyclable from non-recyclable items

All the objects are grasped in a fixed sequence, using pre-defined poses for image capture and grasp. The set of 10 objects is not changed across the 3 trials. We used an unknown prediction threshold $p_1 = 0.18$ and an uncertain prediction threshold of $p_2 = 0.04$. The rules governing sorting for this experiment, based on object material, are outlined in Table 5.

D.4 Retrieving objects from a cluttered bag

For this experiment, we place three objects in a cloth bag: a T-shirt, a steel fork, and a bag of popcorn. The robot captures an image, reaches into the bag, grasps each object, and retrieves objects if the CLAMP model predicts it to be one of (`aluminium`, `brass`, or `steel`). We use a total of two object sequences across the 13 trials: fork → T-shirt → popcorn and T-shirt → fork → popcorn. We

observe that grasping the bag first or second causes the cloth bag to crumple and thus prevent further grasps.

D.5 Separating ripe from overripe bananas

We finetune the haptic encoder transferred to compliance classification, on haptic data from the Franka with Parallel Jaw embodiment. We use the same hyperparameters as those used for finetuning in Table 4. For this experiment, we classified bananas as ripe if the compliance prediction was hard, and overripe if the prediction was soft.