Touch and Go: Learning from Human-Collected Vision and Touch
Supplementary Material

Fengyu Yang[1*] **Chenyang Ma**[1*] **Jiacheng Zhang**[1]
Jing Zhu[1] **Wenzhen Yuan**[2] **Andrew Owens**[1]
[1]University of Michigan [2]Carnegie Mellon University
https://touch-and-go.github.io

A Project webpage

We've provided a webpage for our dataset, which contains a link to the dataset. We also provide additional examples from our dataset (c.f., Fig 2 of the main paper).

B Dataset file structure

Our dataset is currently available through our webpage (and directly via this link). For long-term maintenance, we will upload our dataset to University of Michigan's EECS web servers after acceptance.

The `touch_and_go` directory contains a `dataset` directory of raw videos, `extract_frame.py` that convert raw videos to frames, `label.txt` of material labels for frames within the press, and `category_reference.txt` of the name for each category in `label.txt`.

Each raw video folder in the *Dataset* folder consists of six items:

- `video.mp4`: Raw RGB video recording the interaction of human probing objects.
- `gelsight.mp4`: Raw GelSight (tactile) video for objects.
- `time1.npy`: The recording time for each frame in "`video.mp4`".
- `time2.npy`: The recording time for each frame in "`gelsight.mp4`".
- `video_frame`: The folder containing all the frames in "`video.mp4`". (Generated after running `extract_frame.py`)
- `gelsight_frame`: The folder containing all the frames in "`gelsight.mp4`". (Generated after running `extract_frame.py`)

We have provided qualitative examples of the videos on our project page. To view the videos at full resolution, please download them.

C Egocentric recording setup

As shown in Fig. 1, we use a webcam to record the RGB video and a GelSight sensor to capture the tactile signals, which are both connected to one laptop computer. To obtain images that show clear, zoomed-in images of the objects being touched, two people collected data at once: one who presses

[*]The first two authors contributed equally to this paper.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

Material	Scene	Quantity	Label Num.
Synthetic Fabric	Indoor	1.65K	8
Concrete	Indoor/Outdoor	1.40K	0
Wood	Indoor/Outdoor	1.24K	3
Rock	Indoor/Outdoor	1.08K	15
Tree	Outdoor	0.91K	12
Plastic	Indoor/Outdoor	0.80K	1
Plants	Outdoor	0.78K	18
Metal	Indoor/Outdoor	0.76K	4
Gravel	Indoor/Outdoor	0.71K	16
Sand	Outdoor	0.70K	17
Tile	Indoor	0.63K	6
Rubber	Indoor/Outdoor	0.62K	10
Grass	Outdoor	0.61K	13
Brick	Indoor/Outdoor	0.60K	5
Paper	Indoor	0.45K	11
Leather	Indoor	0.38K	7
Glass	Indoor/Outdoor	0.23K	2
Natural Fabric	Indoor/Outdoor	0.22K	9
Soil	Indoor/Outdoor	0.16K	14
Others	Indoor/Outdoor	0.09K	19

Table 1: We provide statistics for different material categories.

the tactile sensor onto an object, and another who records an "approximately egocentric" video. Alternatively, one person may record both signals, while another holds the computer, providing them with a view of what they are pressing via the screen. In this way, they can ensure that the objects they are probing appear approximately in the center of the recorded images and increase the stability of the recording.

Figure 1: A photo of two humans collecting data in the wild.

D Category list

We conclude the objects appeared in our dataset into 20 categories according to their material property. All these categories are listed with decreasing number of quantities in terms of the number of touches. Label Num. denotes the number in the `label.txt` representing each category.

E Implementation details for self-supervised learning

When training the contrastive multiview coding (CMC) model, we use a learning rate of 0.03 and train for 240 epochs. We use SGD as our optimizer and set the weight decay to be 10×10^{-4} and the momentum to be 0.9. We use a batch size of 128 on 4 Nvidia 2080-Ti GPUs. For the linear probing

stage in both downstream tasks, we fixed the weight of our pretrained backbone and adopt the global average pooling at the last layer followed by a linear classifier. We use a learning rate of 0.01 for ResNet-18 and 0.1 for ResNet-50. For both material classification and robot grasping, we train the linear classifiers with 60 epochs and a batch size of 256.

F Details for Tactile-driven image stylization

Architecture. Our model consists of a multi-modal generator, a tactile-visual texture discriminator and a patch-wise structure discriminator. We can further break up our multi-modal generator into three components, an image encoder $G_{\text{enc\_I}}$, a tactile encoder $G_{\text{enc\_T}}$ and a decoder G_{dec}. Given our dataset that contains unpaired instances $S_n = \{\mathbf{x}_I, \mathbf{x}'_T\}$, the output image $\hat{\mathbf{x}}_I$ can be expressed as $\hat{\mathbf{x}}_I = G(\mathbf{x}_I, \mathbf{x}'_T) = G_{\text{dec}}(\text{concat}(\text{G}_{\text{enc\_I}}(\mathbf{x}_\text{I}), \text{G}_{\text{enc\_T}}(\mathbf{x}'_\text{T})))$.

Structure preserving loss ($\mathcal{L}_{\text{CUT}}$). Our goal in this tactile-guided image stylization is to restyle the source image with the textures that are associated with the target tactile input while preserving the source structure. Following previous approaches [7, 5], we introduce an a noise contrastive estimation (NCE) loss [7] on the image encoder $G_{\text{enc\_I}}$ that helps preserve the structural information between the visual input $\mathbf{x}_I$ and the generated image $\hat{\mathbf{x}}_I$.

This loss is motivated by recent contrastive learning to maximize the probability for the neural network to select the corresponding patch in both the original image $\mathbf{x}_I$ and the generated image $\hat{\mathbf{x}}_I$. Specifically, we select a query patch from the generated $\hat{\mathbf{x}}_I$, one positive patch and N negative patches from the original image $\mathbf{x}_I$. Then we encode these patches into a K dimensional vectors by a MLP so that query vector $\boldsymbol{q}$, positive vector $\boldsymbol{v}^+$ belong to $\mathbb{R}^K$ and negative vectors $\boldsymbol{v}^- \in \mathbb{R}^{N\times K}$:

$$l(\boldsymbol{v}, \boldsymbol{v}^+, \boldsymbol{v}^-) = -\log \frac{\exp(\frac{\boldsymbol{q}\cdot\boldsymbol{v}^+}{\tau})}{\exp(\frac{\boldsymbol{q}\cdot\boldsymbol{v}^+}{\tau}) + \sum_{n=1}^{N} \exp(\frac{\boldsymbol{q}\cdot\boldsymbol{v}^-}{\tau})} \tag{1}$$

where τ is the temperature parameter.

Since our image encoder is a multi-layer convolutional network, we take advantage of multiple feature stacks generated from different layers. Specifically, we select L layers of feature stacks and pass them into a MLP M and the output is $M(G^l_{\text{enc\_I}}(\mathbf{x}_I)) = \{\boldsymbol{v}^1_l, \boldsymbol{v}^2_l, ..., \boldsymbol{v}^N_l, \boldsymbol{v}^{N+1}_l\}$, where $l \in \{1, ..., L-1, L\}$. Here, we denotes $G^l_{\text{enc\_I}}(\mathbf{x}_I)$ as the feature stacks at layer l. Similarity, we apply this to the generated image $\hat{\mathbf{x}}_I$ so that we get our query vector for each layer, which can be represented as $\{\boldsymbol{q}^1_l, \boldsymbol{q}^2_l, ..., \boldsymbol{q}^N_l, \boldsymbol{q}^{N+1}_l\}$. Thus, for each sample index n at layer l, we let $\boldsymbol{v}^n_l \in \mathbb{R}^{N\times C_l}$ as the positive samples and other features $\boldsymbol{v}^{(N+1)\backslash n}_l \in \mathbb{R}^{N\times C_l}$ as negative samples, where C_l indicates the channel of the layer l. Thus our multi-layer NCE loss can be represented as the following:

$$\mathcal{L}_{\text{CUT}} = \mathbb{E}_{\mathbf{x}_I \sim S_n} \sum_{l=1}^{L} \sum_{n=1}^{N+1} l(\boldsymbol{q}^n_l, \boldsymbol{v}^n_l, \boldsymbol{v}^{(N+1)\backslash n}_l) \tag{2}$$

where S_n contains mismatched image-tactile pairs $\{\mathbf{x}_I, \mathbf{x}'_T\}$, as defined in our main text.

Implementation details. Our image encoder and decoder of the generator are fully convolutional neural networks consisting of 9 blocks of ResNet-based CNN bottlenecks. The first convolution layer is set to 7×7 and the rest are set to 3×3. For the tactile encoder, we adopt a ResNet-18 [3] backbone pretrained on the ImageNet [1]. For the discriminator we adopt the PatchGAN architecture [4]. To compute the NCE loss, we extract features from five different layers: the input image layer, the first and second downsampling convolution layer and the first and fifth residual blocks. We train our model on 4 Nvidia 2080-Ti GPUs for 100 epochs with the batch size of 8 and learning rate of 0.0002. For input visual images, we use a random crop and horizontal flip.

G More results for tactile-drive image stylization

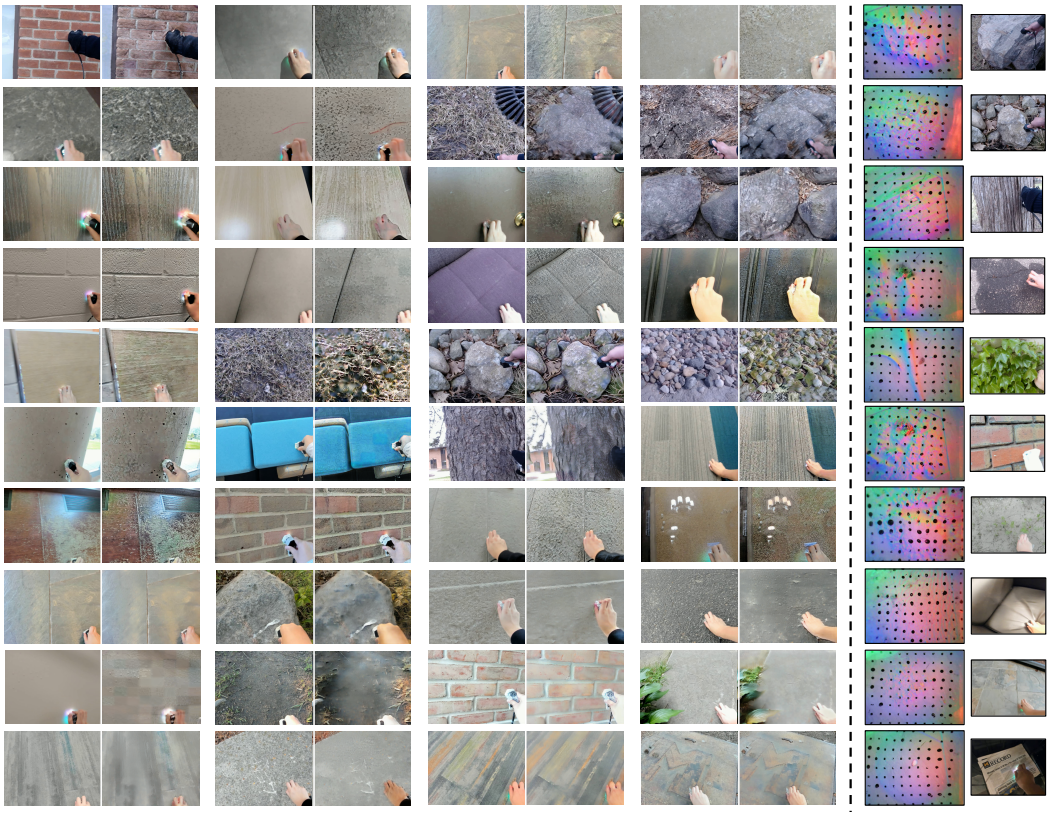

Figure 2: More visualizations of our model on tactile-driven image stylization. For each row, we show an input image (left) and the manipulated image (to its right) obtained by stylizing with a given tactile input (right side). For reference, we also show the image that corresponds to the tactile example at rightmost (not used by the model). *Zoom in for better view.*

H Details for multimodal future touch prediction

Overall Architecture Following [2], our model adopts widely-used residual network from [8] while replacing the 2D convolution to 3D convolution, which utilizes a encoder-decoder architecture. To adapt for multimodal prediction, we introduce two encoders for tactile inputs and visual inputs with identical structures but different weights. Then we concatenate these features along the channel and feed them into the decoder consisting of transposed convolution layers, similar to the architecture of tactile-driven stylization.

Training details For the video prediction task, we train our model using Adam Optimizer with the learning rate of 2×10^{-4} for all experiments. We utilize the batch size of 8 on 4 Nvidia 2080-Ti GPUs and train for 30 epochs. We initialize the weights from a Gaussian distribution with the mean 0 and std of 0.02. To obtain multi-frame prediction, we recursively feed our output images back to the original model. During this process, the loss are backward through the entire chain of recursive functions and gradients are accumulated, following [2, 6].

Evaluation Metrics Following [2], we adopt three evaluation metrics: MAE, SSIM and LPIPS. Structural similarity (SSIM) is a similarity metric to quantify image quality degradation. The higher the SSIM, the better the generated frame. Learned Perceptual Image Patch Similarity (LPIPS) measures the distance between image patches. The lower the LPIPS, the higher the similarity.

I Datasheet

Motivation

Q1. For what purpose was the dataset created?

Answer: The goal of this dataset is to provide training data for multimodal learning systems that learn to associate the sight of objects with their corresponding tactile data (i.e., how they "feel"). In contrast to previous efforts, our dataset contains a large number of in-the-wild recordings from indoor and outdoor scenes.

Q2. Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

Answer: Six researchers at the University of Michigan and Carnegie Mellon University (affiliated as of 2022) have created the dataset: Fengyu Yang, Chenyang Ma, Jiacheng Zhang, Jing Zhu, Wenzhen Yuan and Andrew Owens.

Q3. Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.

Answer: Our dataset is funded in part by Cisco Systems and The University of Michigan.

Q4. Any other comments?

Answer: No.

Composition

Q5. What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?

Answer: Each instance is a visuo-tactile image pair containing the visual image and its corresponding tactile signal, i.e. the result of someone pressing the object with a GelSight tactile sensor.

Q6. How many instances are there in total (of each type, if appropriate)?

Answer: There are in total approximately 246k visuo-tactile image (frame) pairs of about 13.9k touches in our dataset.

Q7. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?

Answer: Yes. We have provided the full dataset.

Q8. What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features?

Answer: The raw data consists of videos recorded by human collectors and the corresponding tactile videos. The RGB videos and tactile videos are synchronously recorded, and compressed with a video codec.

Q9. Is there a label or target associated with each instance?

Answer: Yes. We label all frames where human are probing an object with its material label.

Q10. Is any information missing from individual instances?

Answer: No.

Q11. Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?

Answer: Since we walk through scenes, recording objects around us, the objects in a video are close in space. Tactile signals from the same materials or objects are likely to be similar.

Q12. Are there recommended data splits (e.g., training, development/validation, testing)?

Answer: As illustrated in the main text, different tasks require different train/val/test splits. In general, to avoid having the same (or nearly the same) images appear in both training and test set, we recommend splitting the dataset by video (rather than by touch or by frame). We will provide the splits used in our experiments.

Q13. Are there any errors, sources of noise, or redundancies in the dataset?

Answer: It is a challenging task to infer the material according to the RGB images. We have at least 5 people label each image, though still possible to have some images correctly labeled for its material category.

Q14. Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?

Answer: The data is self-contained.

Q15. Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)?

Answer: No.

Q17. Does the dataset relate to people?

Answer: No.

Collection Process

Q18. How was the data associated with each instance acquired?

Answer: The data is directly collected by two people (authors) walking through a variety of environments, probing objects with tactile sensors and simultaneously recording their actions on videos.

Q19. What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?

Answer: We collect our dataset using a RGB camera and a GelSight tactile sensor. Details of the hardware is illustrated in the main text.

Q20. If the dataset is a sample from a larger set, what was the sampling strategy?

Answer: No, the dataset is not a sample from a larger set.

Q21. Who was involved in data collection process (e.g., students, crowd-workers, contractors) and how were they compensated (e.g., how much were crowd-workers paid)?

Answer: Our dataset is collected by authors of this paper.

Q22. Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)?

Answer: The dataset is collected across the winter, spring and summer (February 2022 to June 2022). The objects in our dataset are taken from scenes at the specific time (and season) in which the data was collected.

Q23. Were any ethical review processes conducted (e.g., by an institutional review board)?

Answer: Our dataset only contains natural scenes, with no humans subjects (including no humans on screen). It therefore does not qualify as human subjects research.

Q24. Does the dataset relate to people?

Answer: No.

Preprocessing, Cleaning, and/or Labeling

Q25. Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?

Answer: Yes. We collect our raw data in the format of RGB and GelSight videos. To facilitate training and downstream tasks, we preprocess the raw videos by converting them into frames, detecting the frames within the press, and label the pressed frames by their material. Detailed description are in the *Dataset* section of the main text.

Q26. Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?

Answer: Yes. We save the original videos for unanticipated future uses of other tasks.

Q27. Is the software used to preprocess/clean/label the instances available?

Answer: Yes. The source code to extract frames is available on our webpage.

Q28. Any other comments?

Answer: No.

Uses

Q29. Has the dataset been used for any tasks already?

Answer: Yes. As illustrated in the main text, we apply our dataset to a variety of multimodal learning tasks. First, we learn tactile features through self-supervised learning, by training a model to associate images with touch. Secondly, we use our dataset to perform material classification task via GelSight Images. Thirdly, we propose a novel task of *tactile-driven image stylization*: making an image "feel more like" a given tactile input. Finally, we study multimodal models for future touch prediction: predicting future frames of a touch sensor's recording, given both visual and tactile signals.

Q30. Is there a repository that links to any or all papers or systems that use the dataset?

Answer: We do not have a repository to record all papers using our dataset. However, we can track these papers via Google Scholar.

Q31. What (other) tasks could the dataset be used for?

Answer: Our dataset is potentially suitable for tasks that require visual, tactile, or visuo-tactile understanding, such as visual-tactile image translation, shape/hardness estimation, etc.

Q32. Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?

Answer: Our dataset was mainly collected in one geographic location (near University of Michigan's campus). Consequently, the data we recorded may not generalize to all spaces. The use of humans in the data collection process also potentially introduces bias, which differs from "robotic" or "virtual data" bias. It was also recorded by a relatively small number of human collectors. The way that they interacted with the objects may therefore not be fully representative.

Q33. Are there any tasks for which the dataset should not be used?

Answer: Our dataset is designed for visuo-tactile learning tasks. It may be not appropriate for tasks outside this domain.

Q34. Any other comments?

Answer: No.

Q35. Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?

Answer: Yes. Our dataset is publicly available.

Q36. How will the dataset will be distributed (e.g., tarball on website, API, GitHub)

Answer: Our dataset contains a link to a Google Drive directory that contains all of the raw videos, a "`extract_frame.py`" file to convert videos into frames and a separate "`label.txt`" file containing all material labels for frames within the press (See B for more details).

Q37. When will the dataset be distributed?

Answer: We have currently provided all raw data, including videos, tactile recordings, labels, and code. Our dataset will be officially released starting by October 2022 (e.g., in an easy-to-download format and with full documentation).

Q38. Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?

Answer: Our dataset is distributed under the license of CC BY.

Q39. Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?

Answer: No.

Q40. Any other comments?

Answer: No.

Maintenance

Q41. Who will be supporting/hosting/maintaining the dataset?

Answer: Our dataset is currently hosted on a public Google Drive directory. We will also mirror the dataset using a web server provided by The University of Michigan, so that it will be available indefinitely.

Q42. How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

Answer: The email of authors of our dataset is available on the project webpage.

Q43. Is there an erratum?

Answer: No. If we notice errors in the future, we will put them in an erratum.

Q44. Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?

Answer: There is no routine update plan for our dataset. To correct labeling errors, please contact authors of our dataset.

Q45. If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?

Answer: No. Our dataset is not related to people.

Q46. Will older versions of the dataset continue to be supported/hosted/maintained?

Answer: No. We only maintain the latest dataset unless there is a significant update.

Q47. If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?

Answer: We have provided information about how the data was collected, including the sensors and the dataset collection procedure. Thus, those who want to collect similar data can easily do so.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[2] Daniel Geng, Max Hamilton, and Andrew Owens. Comparing correspondences: Video prediction with correspondence-wise losses. *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[5] Tingle Li, Yichen Liu, Andrew Owens, and Hang Zhao. Learning visual styles from audio-visual associations. *European Conference on Computer Vision (ECCV)*, 2022.

[6] William Lotter, G. Kreiman, and David D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *ArXiv*, abs/1605.08104, 2017.

[7] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for conditional image synthesis. In *ECCV*, 2020.

[8] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.