A SUPPLEMENTARY

A.1 DATA COLLECTION SETUP AND TACTILE-HAND VISUALIZATION

Figure. 5 shows our data collection setup and tactile-hand visualization to facilitate understanding of the hand-tactile pressure map signal. **Additional Visualization and Data Overview** We add additional visualization to illustrate our data in Fig. 6. Visualization of motion sequences can be found on our supplementary website. **Background on Glove** The glove is made using piezoresistive film. The working principle of this material is when the material stretches, it causes the deformation of carbon nanotubes, which then induces the relative electrical resistance changes and thus difference in voltage readouts. The material in general is sturdy and the material property does not change much with wear and tear.

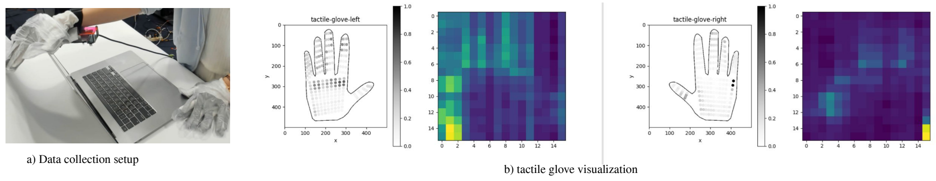

Figure 5: Data Collection Setup and Tactile-Hand Visualization

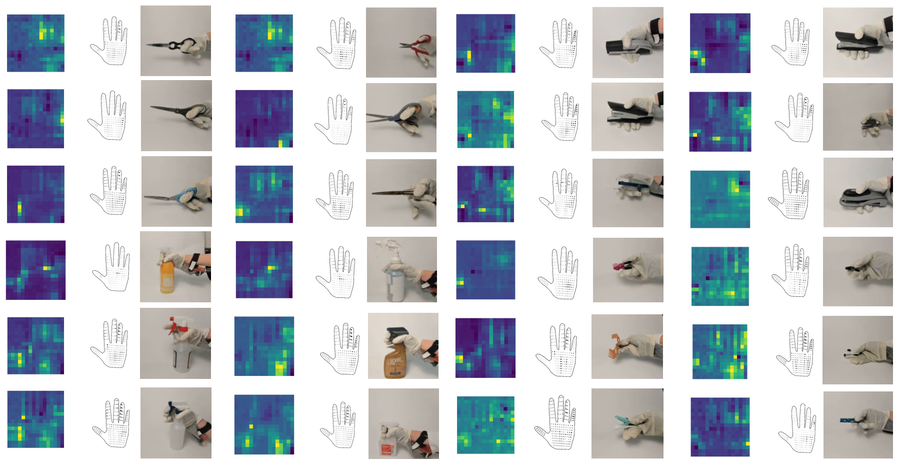

Figure 6: Data Visualization. We show a demo of our data with 12 samples from our 89 objects.

Glove Calibration. The glove only needs to be calibrated once to convert the sensor readouts in voltages to force magnitudes in Newton before use. We show the process of data calibration in Fig. 7.

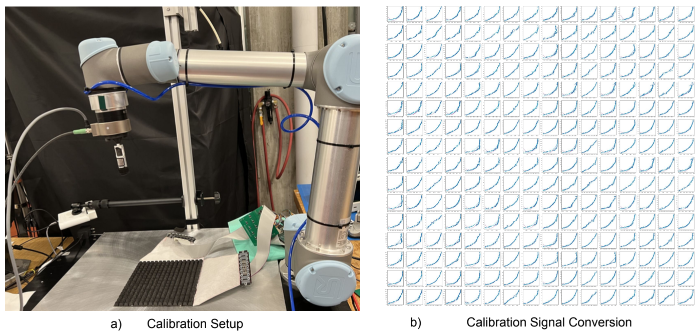

Figure 7: Sensor Calibration. The left shows the sensor calibration setup before the embroidery of the insulation layer. The right shows the collected calibration data from 0 to 50N of continuous force applied to each of the 16 $\times$ 16 voltage readout positions.

A.2 Downstream tasks

To motivate downstream robotic applications, we show two experiments inspired by robotic perception applications. The high-level intuition is that if our learned embedding space trained with human-collected data can sufficiently represent data collected by robotic arms, we can transfer knowledge from human demonstration to robotic tasks. First, we collected some data using a robotic hand to show the cross-modal prediction results leveraging the pretrained embedding with human data. Additionally, we would like to see how much knowledge is transferrable from human demonstrations to data to a robotic hand. To achieve this goal, we train a simple classifier using our trained embedding space with human data to enable manipulation and object classification through tactile sequences. While we only show two applications here, other downstream tasks can be enabled by our proposed method and dataset.

Robotics inspired application. We recognize that different robotic hands might induce different levels of domain gap. To minimize the domain gap, we use a robotic motor to drive a tenden-driven OpenBionics hand, and record a tactile sequence shown below Fig. 8 a) and b). We use test-time optimization approach to find a latent code in the shared embedding space to decode into other modalities. We notice that because the robotic hand is non-conformable, the collected sequences of data are of inherent domain gap and thus result in different possible predictions that equally fit the optimization energy.

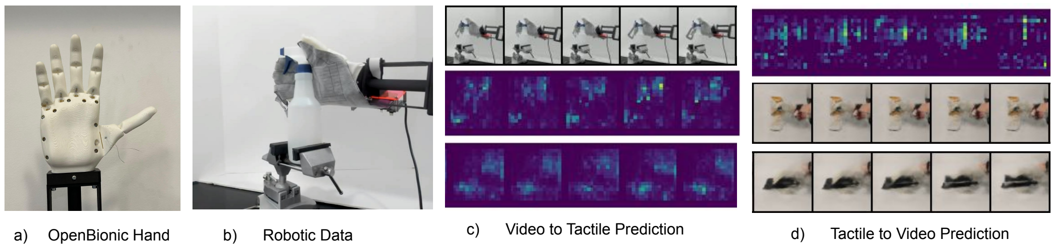
a) OpenBionic Hand
b) Robotic Data
c) Video to Tactile Prediction
d) Tactile to Video Prediction

Figure 8: Robotic application using shared manifold learning trained with human collected data. a) and b) show the task and data collection setup. c) d) shows the cross-modal prediction results of only using the video or the tactile sequence to infer the other modality of signals. The top row shows the query signal, and the bottom two rows showcase different predictions made by our method.

Object classification with shared embedding . One other potential application is to help robotic tasks in low-light settings, by equipping robotic hands with tactile sensors. We can leverage our pretrained cross-modal manifold to recognize objects and even operate objects when vision is impaired. With our trained shared latent space we train a classifier to classify the corresponding object category and manipulation type from the signal sequence. We withhold some latent codes from the classifier to test the baseline performance on human data. We use test-time optimization to obtain a latent code in the shared embedding space using the robotic tactile data. We show the performance in Table. 2 below.

test data type	manipulation	object
human test data	0.96	0.97
robotic tactile data	0.86	0.48

Table 2: Object classification of human and robotic data using shared embedding space

We notice that human manipulation and robotic manipulation share a significant amount of transferrable and common knowledge. In that a classifier trained using the shared latent code that has only seen human data and also be used to classify robotic manipulation data. On the other hand, the non-conformability of the robotic hand significantly restricts the information on the details of the object geometry in the tactile signal and thus limits the accuracy of classifying objects using robotic tactile data.

While these are simple robotic applications of our proposed dataset and method, we note here that other more advanced robotic applications are possible. We focus on learning a joint representation for different modalities of signals and leave more challenging robotic tasks as future work.

A.3 Multi-Modal Prediction

We randomly initialize three different latent codes to start our test-time optimization for a given tactile video sequence. We show that we are able to recover different cross-modal sequences with different initializations, shown in Figure. 9. We want to note that it is possible that different latent initializations converge to the same mode of prediction when there exists a global optima, or when two latent codes are initialized around the same neighborhood of a local minima.

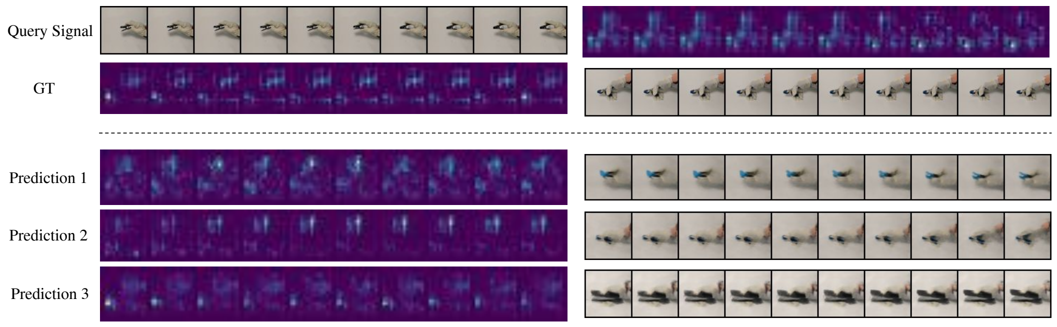

Figure 9: MultiModal Prediction via variational initialization test-time optimization.

A.4 Ablation Experiments

Table 3: Ablation Study

	video to tactile	tactile to video
first frame inference	0.0028	0.0247
w/out projection layer	0.0022	0.0194
dim z_h= 256	0.0018	0.0212
dim z_h= 128	0.0018	0.0201
dim z_h= 64	0.0019	0.0196
dim z_h= 5	0.0023	0.0188
ours	0.0019	0.0174

We show our ablation experiments in Table. 3. We observe that removing the projection layer results in decrease of cross-modal prediction performance. We can also observe that by varying the latent dimension of the tactile signal z_h, the performance for video to tactile prediction does not change too much, which means that z_h is over-parameterized. Through decreasing the latent dim z_h, performance for tactile to video starts to increase. when latent dimension decreases, the neural field enforce the latent code to use limited space to register the same amount of information, and thus captures the principled information in sparse signals such as tactile. Our proposed method leverages latent dimension of 16, which trades off the marginal gain of accuracy reconstructing tactile signal, but captures much more principled information in the tactile space. Finally, we also show that our model can also be directly adopted to only use the first frame signal during test-time optimization, we notice a performance decrease from using the full sequence data. We believe the difference resulted from an increase of variational inference, or in other words, with less optimization signal, there are now more local minima during test-time optimization that fit the optimization objective.

A.5 Implementation Details

All our manifolds are randomly initialized with Gaussian distribution with $(\mu, \sigma) = (0, 1)$. Tactile manifold $\mathcal{M}_h$ is initialized to be 16 dimensional and all other manifolds are initialized to be 256 dimensional. We use MLPs to express our neural fields Φ. All MLP neural fields are of three layers and 512 hidden dimension. Specifically, the shared manifold is initialized to be an 256 dimensional embedding of with $N_{train} = 123,561$ randomly sampled from a Gaussian distribution of $(\mu, \sigma) = (0, 1)$. Γ_c is parameterized by a 256×256 matrix, and Γ_h is parameterized using a

256×16 matrix. Neural fields ϕ_c, ϕ_x, ϕ_h are all 6-layer MLPs using ReLU activation and 512 dimensional hidden-layers. We also use positional encoding for our signal-agnosticneural field with frequency coefficient set to $\omega = 30$.

A.6 Training Details

All baseline methods and our proposed method are implemented using the open-source Pytorch (Paszke et al., 2019) package with CUDA 11.7 backend. We use the same hardware and training setup to train all methods mentioned in this work. We use a workstation equipped with NVIDIA Tesla V100 GPUs and 64-core AMD CPUs. We use Adam optimizer for training the baselines and our methods with learning rate initialized to be $1e-3$; batch size is set to 64. All methods are trained to full convergence. For test-time optimization of latent code for GEM (Du et al., 2021) and our proposed approach, we use Adam optimizer to run 1000 timesteps with batchsize of 32. For 1000 timesteps, we use 56 seconds in total, 17.87 iteration/second, on average every example use 1.75 seconds.