

# ACROSS: A Deformation-Based Cross-Modal Representation for Robotic Tactile Perception

Wadhah Zai El Amri<sup>1</sup>, Malte Kuhlmann<sup>1</sup> and Nicolás Navarro-Guerrero<sup>1</sup>

**Abstract**—Tactile perception is essential for human interaction with the environment and is becoming increasingly crucial in robotics. Tactile sensors like the BioTac mimic human fingertips and provide detailed interaction data. Despite its utility in applications like slip detection and object identification, this sensor is now deprecated, making many valuable datasets obsolete. However, recreating similar datasets with newer sensor technologies is both tedious and time-consuming. Therefore, adapting these existing datasets for use with new setups and modalities is crucial. In response, we introduce ACROSS, a novel framework for translating data between tactile sensors by exploiting sensor deformation information. We demonstrate the approach by translating BioTac signals into the DIGIT sensor. Our framework consists of first converting the input signals into 3D deformation meshes. We then transition from the 3D deformation mesh of one sensor to the mesh of another, and finally convert the generated 3D deformation mesh into the corresponding output space. We demonstrate our approach to the most challenging problem of going from a low-dimensional tactile representation to a high-dimensional one. In particular, we transfer the tactile signals of a BioTac sensor to DIGIT tactile images. Our approach enables the continued use of valuable datasets and data exchange between groups with different setups.

## I. INTRODUCTION

Tactile feedback is gaining significant attention in robotics [1], [2]. Tactile sensors leverage various information modalities, come in diverse shapes and sizes, and are implemented in various technologies. This diversity makes the exchange of data and trained models challenging. Moreover, as sensor technology improves, datasets become obsolete. For instance, BioTac by SynTouch was a high-end tactile sensor designed like a human fingertip. It has an elastomer covering a rigid core filled with an incompressible conductive fluid. The sensor outputs voltage readings from 19 internal electrodes, capturing changes in the fluid. These readings are processed as time-series signal data [3]–[5]. The BioTac has been proven useful in various applications such as detecting object slips and the direction of slips [6], [7] or identifying objects [8]. However, this sensor is now deprecated. Consequently, many influential datasets, such as the BioTac SP direction of slip dataset [7], the BioTac SP grasp stability dataset [6] or the BioTac 2P grasp stability dataset [9], are now obsolete. These datasets capture sensor outputs, specifically BioTac signals, recorded while the sensors are mounted on robotic hands that grasp various objects under different conditions. The data is then used, for instance, to evaluate the stability of the grasps. Despite their obsolescence, such datasets remain important,

as grasp stability and slip detection continue to be an active field of research [10].

Furthermore, designing and collecting similar datasets is a time-consuming and complex task. It requires careful consideration of various factors, such as the choice of sensors and their resolution, the data collection methods, and the labeling process, among other requirements.

Hence, there is a need to convert useful datasets into formats compatible with newer sensor modalities, even if they involve different robotic or sensor configurations. This data conversion would allow researchers to leverage intrinsic information still relevant to specific tasks, while also saving time and resources by avoiding the need to collect entirely new datasets.

To this end, we propose ACROSS, a versatile approach for transferring tactile data between sensors of varying resolutions, including low-to-high, high-to-high, and high-to-low resolution transfers [11]. We demonstrate the effectiveness of ACROSS by converting low-resolution tactile time-series data from a BioTac sensor into a high-resolution vision-based DIGIT sensor [12]. Our method enables the utilization of existing datasets gathered with outdated sensors, avoiding the tedious process of gathering data from scratch. Moreover, it facilitates a transition between two intrinsically distinct tactile sensor modalities, e.g., signal data, to visual representations.

While highly accurate simulations could generate realistic sensor data, they are insufficient for direct cross-sensor transfer, as they do not intrinsically provide a way to relate the outputs of different sensors or bridge the gap between their modalities. Current physics simulation environments, such as Isaac Gym [13], primarily model sensors' deformation during object interaction but do not provide the actual sensor-specific outputs, such as BioTac sensor signals or DIGIT sensor images. Moreover, these environments do not provide a way to map deformations between different sensors, even when both are simulated within the same environment. ACROSS addresses this gap by enabling the transfer of sensor outputs, even across different simulation environments, making it a crucial framework for data transfer beyond what direct simulation alone can achieve.

Additionally, we provide an openly available dataset comprising over 155K unique 3D mesh deformation pairs from interactions involving BioTac and DIGIT sensors. This dataset includes various types of indenters and the force exerted on each sensor. The source code, dataset, and neural network checkpoints can be found on our website: <https://wzaielamri.github.io/publication/across>.

<sup>1</sup>L3S Research Center, Leibniz Universität Hannover, Hanover, Germany {wadhah.zai, malte.kuhlmann, nicolas.navarro}@l3s.de

## II. RELATED WORK

Tactile sensors can capture the same deformation of an object, but they may represent this deformation differently depending on the type of sensor used [14]. For instance, the elastomer's deformations can be represented as either time-series signals or images, depending on the modality of the sensor. Therefore, our proposed approach focuses on transferring the encoded information and knowledge at the deformation level rather than at the output level, which varies between sensors. Although some research attempted to transfer between modalities, for instance, Lee et al. [15] developed a framework to generate tactile images from the GelSight sensor using digital camera images of various cloth materials and vice versa, or the ViTac dataset [16], used to train the networks, includes labeled images from the GelSight sensor and a digital camera of 100 fabric pieces. Their framework utilizes two separate cycleGANs for the bidirectional transfer. Unfortunately, these approaches require labeled data from both sensor types. Moreover, the modality of the source and target sensors is the same, i.e., vision.

Tatiya et al. [17] introduced a framework for transferring knowledge across sensor modalities, allowing robots to handle sensor failures or operate with different sensor configurations. Their method leverages a variational encoder-decoder network (VED) to map sensory observations from one modality, such as vibration, to another, such as haptic, by learning a shared feature space between them. However, the approach similarly requires end-to-end labeled data, which can be resource-intensive. Additionally, the framework assumes the same set of objects is used across experiments, potentially limiting its generalization to novel objects.

In contrast, our approach does not require end-to-end labeled sensor outputs and can generalize to any contact form, force, or orientation. Furthermore, we facilitate transfers between distinct modalities and sensors with different morphologies and sizes. Our framework starts by converting the input signals into 3D deformation meshes. Next, we transition from the 3D deformation mesh of one sensor to that of another, and finally, we translate the generated 3D deformation mesh into the corresponding output space. Unlike prior research, we address the challenging task of converting a low-dimensional tactile representation into a high-dimensional one. Specifically, we transfer tactile signals from a BioTac sensor to DIGIT tactile images.

Our approach is inspired by Narang et al. [18], [19], who introduced a framework using a finite element method (FEM) model to simulate the deformation of the BioTac sensor interacting with different indenters. Two variational autoencoders (VAE) were used. The first, a vanilla VAE with linear layers, reconstructs the BioTac signals, while the second, a convolutional mesh autoencoder (CoMA), reconstructs the mesh deformation. CoMA network employs fast localized spectral filtering, i.e., Chebyshev filters alongside hierarchical pooling operations. These operations are adapted for 3D meshes by computing the spectral information of the mesh graph using Fourier transformation and then applying

Chebyshev filters for localized convolutional operations [20], [21]. Both VAEs are subsequently ‘frozen’, and an MLP is trained to project the latent vector from one modality to another using labeled data pairs. In our work, we adopt Narang et al.’s [18] approach, due to its performance, to generate BioTac deformation meshes from the sensor input signals.

Zhu et al. [22] applied a similar idea to synthesize the volumetric mesh of a vision-based tactile sensor, GelSlim [23]. Two separate VAEs were employed. The first VAE was trained to reconstruct the tactile images captured by GelSlim, ensuring that the network could accurately capture the visual features of the deformed elastomer. The second VAE was dedicated to reconstructing the volumetric mesh from the tactile imprints, capturing the 3D structure of the deformation. Zhu et al. introduced a self-supervised adaptation method that leverages a differentiable renderer to generate synthetic meshes to refine their approach further. This technique improved the encoder’s performance, which embeds the image into the latent space, and the projection Multilayer Perceptron (MLP), which maps between the two different latent representation spaces of the images and the volumetric mesh. By rendering the generated mesh using the differentiable renderer and comparing it to the observed tactile imprint, the system could compute gradients and backpropagate errors, thereby refining the model through further training. This combination of self-supervision and differentiable rendering allowed the system to bridge the gap between simulation and real-world tactile data and generate accurate mesh deformations from tactile images.

In contrast, other researchers have focused on creating images derived from the physical interactions of vision-based sensors. For instance, Wang et al. [24] provided TACTO, a simulation of vision-based sensors, which uses Pyrender [25] and normal forces to derive gel pad deformations and generate then the corresponding sensor images. This simulation is primarily demonstrated for the DIGIT sensor [12]. A vision-based sensor that consists of an elastomeric gel pad that reflects light emitted by a series of internal LEDs, enabling an internal camera to record the deformation of the gel membrane. Another solution, Taxim [26], predicts the output of image-based tactile sensors using example-based photometric stereo methods. It utilizes optical reflection functions to interpret the gel pad’s illumination while interacting with objects. The core concept involves simulating the sensor’s optical output using a polynomial table based on a second-order polynomial function. This function approximates the non-linearity of light in vision-based tactile sensors. Additionally, this approach enables the calibration and adjustment of these polynomial coefficients with real sensor data and allows for the simulation of marker motion fields on the gel surface. Given Taxim’s superior performance over other state-of-the-art methods, we incorporate it into our framework to generate DIGIT images from 3D deformation meshes.

## III. IMPLEMENTATION

Transferring between different modalities poses challenges due to data representations and encoding variations. We

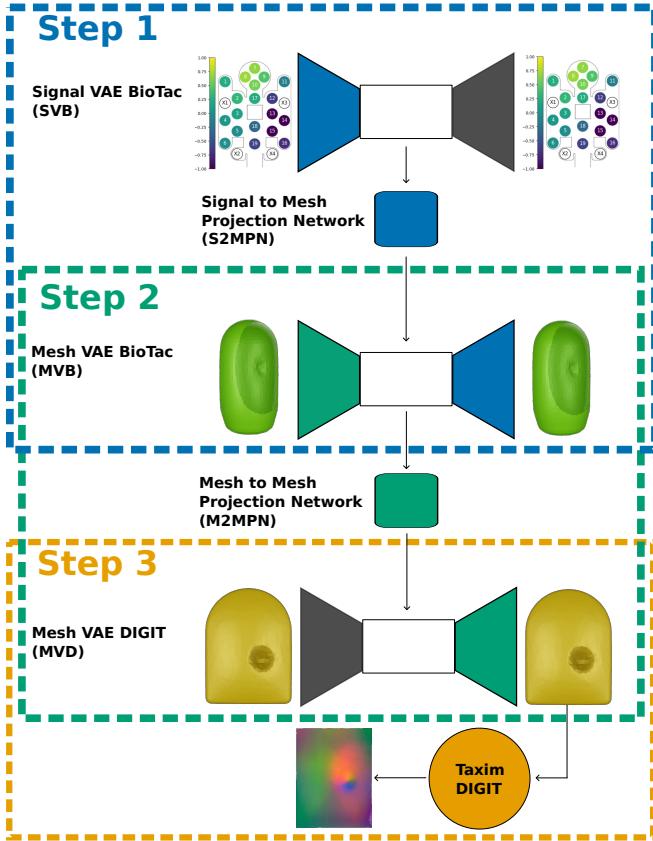


Fig. 1. An example of the ACROSS framework applied to translate BioTac signals into DIGIT images. Step I: Convert BioTac input signals to BioTac surface deformation. Step II: Convert BioTac surface mesh deformation to DIGIT surface mesh deformation. Step III: Generate DIGIT’s output from the surface mesh deformation.

propose a three-step solution to address these issues, depicted in Figure 1. Step I: We initially predict the BioTac surface deformation from the BioTac input signals. Step II: We convert the BioTac surface mesh deformation to DIGIT surface mesh deformation since the physical interaction of both sensors can be modeled by a mesh deformation independently of the sensor output modality. Step III: We generate the DIGIT sensor image from the converted deformation.

#### Step I: Predicting BioTac Mesh Deformation

We adopt a similar methodology to that proposed by Narang et al. [19]. We train a disentangled variational autoencoder ( $\beta$ -VAE) [27] to reconstruct the BioTac sensor outputs. This network is denoted as Signal VAE BioTac (SVB). To train the network, we use a curated dataset that combines two publicly available BioTac signal datasets, i.e., Narang et al. [19] and Ruppel et al. [28]. This combination allows us to effectively augment our data, increasing the diversity and amount of data available for training. By augmenting our dataset, we empirically reduced the overall loss and helped the network learn more robust features by exposing it to a broader range of examples.

We also train another  $\beta$ -VAE to reconstruct the 3D mesh deformation of the BioTac sensor, denoted as Mesh VAE

BioTac (MVB). To model these deformations and collect the dataset used for this task, we employ the validated Isaac Gym BioTac FEM simulation [19].

Next, we train an MLP network to map between the latent vectors of the SVB and the MVB network, referred to as Signal to Mesh Projection Network (S2MPN). For this latent space mapping, we use the publicly available dataset collected by Narang et al. [19]. This dataset comprises data pairs: BioTac electrodes outputs and mesh deformations, resulting from interactions with nine different indenters.

#### Step II: Modeling of Mesh Deformation

In this step, we train a third  $\beta$ -VAE [27], with the same architecture used for the MVB network, this time to reconstruct the DIGIT 3D deformations. This network is denoted as Mesh VAE DIGIT (MVD). The data used to train this network are also collected with Isaac Gym through simulating physical interactions with the DIGIT sensor and recording the corresponding 3D deformations. Given that the sensor’s elastomer is primarily composed of Smooth-On Solaris silicone [12], we configure the FEM soft-body hyperparameters accordingly: an elasticity modulus of 539 kPa and a Poisson’s ratio of 0.499 [29]. Furthermore, we set the dynamic friction coefficient between the DIGIT sensor elastomer and the indenters to 0.78 [19].

Afterward, we train an MLP network to map the latent space of the already trained MVB encoder network in Step I to the latent space of the trained MVD encoder network, and we denote this network as Mesh to Mesh Projection Network (M2MPN). To train the M2MPN, we collected unique paired mesh deformations for both BioTac and DIGIT using the Isaac Gym simulator. Details about the dataset collection procedures follow in Section IV-A.

#### Step III: From Surface Deformation to DIGIT’s Output

We adapt the simulation model Taxim [26] to simulate DIGIT images. Originally, Taxim calculated a height map of the gel pad using object point clouds to estimate the corresponding DIGIT image. We adjust this approach using the deformation mesh instead of the point cloud. Using Pyrender [25], we estimate the height map that the sensor’s internal camera would capture for each deformation mesh. Subsequently, we generate the corresponding DIGIT image by applying this height map and Taxim’s polynomial coefficients. Later, we improve the synthetic image by applying a pyramid Gaussian blur to remove its artifacts and make it more realistic. We describe this in detail in Section IV-B.

## IV. EXPERIMENTS AND RESULTS

This section introduces the datasets used in our framework, followed by a detailed description of our network architecture. Finally, it presents the results of our approach.

#### A. Datasets Description

To train the SVB network, we curated an unlabeled real BioTac signal dataset by normalizing each input channel separately and merging two existing datasets [19], [28] to

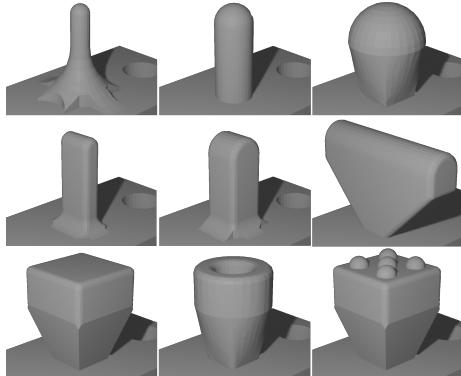


Fig. 2. The nine indenters used to collect the BioTac-DIGIT deformation dataset.

augment our data and improve the network’s generalization and performance on unseen data. Each data vector comprises 19 electrode values, normalized and adjusted to fall within the range of [-1, 1]. The dataset provided by Ruppel et al. [28] contains an error that increased throughout data gathering. This error is attributed to a rise in temperature during the data collection process, which affects the properties of the fluid and causes an output drift [5]. To address this, we fitted a linear function for each electrode by utilizing gradient descent to minimize the difference between each non-contact timestep and the linear function. We then used the linear function to shift the values towards the default values of the electrodes. Default values represent the sensor readings at rest, i.e., no touches are recorded. Furthermore, most non-contact data was excluded to ensure a balanced dataset. The combined dataset yielded almost 250K unique BioTac signal data points.

We then collect a BioTac deformation dataset consisting of approximately 860 unique indentations trajectories. Each trajectory includes 20 different 3D mesh deformations, with depths ranging from 0.1 mm to 2 mm in 0.1 mm increments. We set the maximum indentation to 2 mm, a value chosen to align with later simulations involving the DIGIT sensor. Given that the DIGIT has thinner sides, 2 mm is a suitable maximum indentation for these simulations. Furthermore, each trajectory is replicated using the nine indenter types shown in Figure 2 to have a wide variety of touches. This results in an overall dataset of approximately 155K unique 3D deformation meshes. We use the collected dataset to train the *MVB* network.

Since the *S2MPN* requires labeled data to train, we resimulate the BioTac mesh deformations corresponding to the real signals in Narang et al.’s [19] dataset, following their description. The dataset was then filtered to ensure consistency with our collected BioTac deformation dataset. Only data points within a maximum depth of 2 mm were selected, resulting in a dataset of almost 9K unique data points with the corresponding labeled BioTac signals. The data was divided into distinct trajectories to ensure an unbiased split. We then randomly selected 15% of those trajectories as our test set. The remaining data points were randomly split into training and validation sets, as shown in Table I.



Fig. 3. Transferred BioTac sensor (green) to align it with the DIGIT sensor surface (gold), in order to collect paired 3D deformation meshes.

TABLE I  
SIZE OF THE DATASETS USED IN OUR FRAMEWORK. THE NUMBER BETWEEN PARENTHESIS PRESENTS THE SIZE OF THE INPUT/ OUTPUT OF THE DATA. THE SAME DATASETS ARE USED FOR <sup>1</sup> AND <sup>2</sup>, AND THE SPLITS ARE IDENTICAL.

Networks	Datasets	Train	Validation	Test
<b>SVB</b>	BioTac Signals (19)	196397	27218	24551
<b>S2MPN</b>	BioTac Signals (19)	5841	1409	1417
	BioTac Meshes (4246x3)			
<b>MVB</b>	BioTac Meshes (4246x3) <sup>1</sup>	122924	15560	17121
<b>MVD</b>	DIGIT Meshes (6103x3) <sup>2</sup>	122924	15560	17121
<b>M2MPN</b>	BioTac Meshes (4246x3) <sup>1</sup>	122924	15560	17121
	DIGIT Meshes (6103x3) <sup>2</sup>			

In Step II, we collected a DIGIT mesh deformation dataset comprising approximately 155K unique 3D DIGIT mesh deformations, following the same trajectories as the BioTac mesh dataset. To ensure alignment between the two mesh datasets, we maintained consistent force, angle, and position parameters for each interaction pair since labeled and matching BioTac DIGIT meshes are required for training the *M2MPN*. In order to address the difficulty in representing side touches arising from the differing shapes of the DIGIT and BioTac sensors, we rotate and translate the BioTac sensor on its axis within its horizontal plane, as depicted in Figure 3. This transformation mimics the unfolding of the BioTac elastomer to align with and cover the flat surface of the DIGIT sensor. This solution ensures that forces and deformations resulting from side touches correspond with the other sensor.

### B. Network Descriptions

The encoder of our *SVB* network comprises five layers. The first three are linear layers with sizes [256, 128, 64] and two parallel linear layers which predict  $\mu$  and  $\log(\sigma^2)$ , each with a size of 8. The decoder is composed of four linear layers with sizes [64, 128, 256, 19]. Each layer uses a ReLU activation function. The network is trained to minimize the following function:

$$\ell_S = \text{MSE}(S - \hat{S}) + \beta_S \text{KL}(f(z_S | S) \| \mathcal{N}(0, 1)), \quad (1)$$

where  $f$  is the encoder of the *SVB* network,  $\beta_S$  is the weight of the KL divergence loss, and is equal to 0.005.  $z_S$  is the sampled latent vector given the input  $S$ . The mean-squared error (MSE) is calculated between the normalized predicted and ground-truth signal. The network uses the Adam optimizer with a learning rate equal to 0.0001. The hyperparameters

for all our proposed network architectures are empirically determined.

The *S2MPN* has four linear layers with sizes [512, 128, 256, 256]. Each linear layer is followed by an ELU activation function and a dropout layer with dropout rates equal to [0.4, 0.3, 0.2, 0.5]. It is optimized using Adam with a learning rate of 0.0005 and using the following MSE loss function:

$$\ell_{SMP} = \text{MSE}(z_{MB} - \hat{z}_{MB}), \quad (2)$$

where  $z_{MB}$  represents the predicted BioTac latent space and  $\hat{z}_{MB}$  is the target BioTac latent space.

Both our 3D mesh reconstruction VAEs, i.e., *MVB* and *MVD*, are built upon graph convolutional mesh autoencoders (CoMA). Both networks have identical architectures. The encoder is composed of four graph convolution layers with sizes [16, 16, 16, 32] and a kernel size of 6, each followed by a downsampling layer with a factor of 2. Next, a linear layer with a size of 512 is applied, followed by two parallel linear layers with sizes [128, 128] to represent  $\mu$  and  $\log(\sigma^2)$ , used for the latent space of the VAE. The networks are optimized using Adam, with an initial learning rate of 0.001 and a learning rate decay after each epoch equal to 0.99. The following cost function is minimized:

$$\ell_M = \text{MSE}(M - \hat{M}) + \beta_M \text{KL}(g(z_M | M) \| \mathcal{N}(0, 1)), \quad (3)$$

where  $g$  is the encoder of the corresponding mesh VAE,  $\beta_M$  is the weight of the KL divergence loss, and is equal to 0.005 for both BioTac and DIGIT mesh VAEs.  $z_M$  is the sampled latent vector given the input  $M$ . The mean-squared error (MSE) is calculated between the input and predicted normalized mesh, averaged over the entire batch and all 3D vertices. All networks are trained for a maximum of 300 epochs, with early stopping to prevent overfitting.

Finally, our *M2MPN* consists of four linear layers with sizes [512, 1024, 1024, 256]. Each linear layer is followed by a dropout layer with [0.2, 0.4, 0.0, 0.0] dropout rates, and an ELU activation function. The learning rate is set to 0.001. Early stopping is also employed, and the minimized loss function is defined as follows:

$$\ell_{MMP} = \text{MSE}(z_{MD} - \hat{z}_{MD}), \quad (4)$$

where  $z_{MD}$  represents the predicted DIGIT latent space vector and  $\hat{z}_{MD}$  is the target latent space vector.

In the final step of this pipeline, we use Pyrender [25] to obtain the height map of each DIGIT mesh deformation. The Taxim images generated from these height maps contain some artifacts due to the resolution of the gel pad mesh. To address this issue, we apply an additional pyramid Gaussian blur to the entire generated image, including the contact region, using kernel sizes of [51, 21, 11, 5]. This differs from the original Taxim, which does not apply Gaussian blur to the contact region. Figure 4 illustrates the artifacts and the improvements achieved after applying the additional pyramid Gaussian blur.

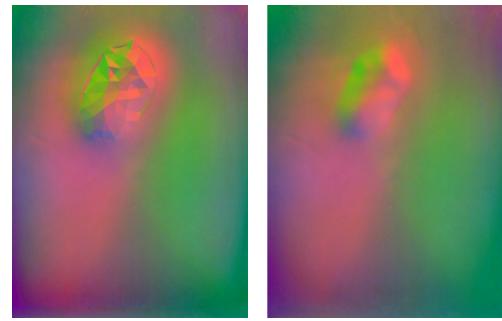


Fig. 4. Comparison of artifacts in the generated image before (left) and after (right) applying the additional pyramid Gaussian blur.

TABLE II  
ROOT-MEAN-SQUARE ERROR (RMSE) AND EUCLIDEAN DISTANCE (EUC. DIST.) MEASURED BETWEEN THE PREDICTION AND THE GROUND-TRUTH OUTPUTS FOR ALL SAMPLES IN THE TEST SET. FOR ALL NETWORKS, THE RMSE AND EUC. DIST. ARE REPORTED IN  $\mu\text{m}$ . VALUES ARE AVERAGED OVER ALL THE MESH VERTICES AND ON ONLY VERTICES IN THE DEFORMATION REGION.

Networks	RMSE	Euc. Dist.	RMSE	Euc. Dist.
	All Vertices	Deformation Region		
<b>S2MPN</b>	78.21 (41.88)	85.00 (49.80)	94.07 (48.32)	121.02 (62.31)
<b>MVB</b>	12.28 (4.75)	13.90 (5.15)	16.03 (4.29)	21.26 (4.92)
<b>MVD</b>	9.28 (4.20)	10.13 (3.98)	12.43 (3.93)	14.60 (4.15)
<b>M2MPN</b>	21.57 (19.08)	18.68 (19.38)	27.72 (20.07)	28.54 (21.90)

### C. Converting Real Data

In this subsection, we assess the performance of our trained networks in converting real-world data. We calculate the root-mean-square error (RMSE) averaged on all unseen test data for all our trained networks to quantify this. The RMSE for the *SVB* is measured between the ground truth normalized electrode values and the network prediction, and is equal to 0.060 with a standard deviation of 0.034. When testing both projector networks, i.e., *S2MPN* and *M2MPN*, we generate the mesh from the predicted latent vector using the frozen trained decoder of the corresponding VAE network, and we measure the average RMSE in micrometers ( $\mu\text{m}$ ) for all vertices between the ground-truth mesh and predicted mesh. Additionally, we calculate the RMSE averaged only for the vertices within the deformation region. The deformation region is defined as comprising all vertices that deviate by 10  $\mu\text{m}$  or more from the original mesh without indentations. Furthermore, we measure the Euclidean distance between the predicted and target meshes in micrometers ( $\mu\text{m}$ ) for all vertices and those specifically within the deformation region. The results of *S2MPN*, *MVB*, *MVD* and *M2MPN* are reported in Table II.

According to Table II, *S2MPN* exhibits higher RMSE and Euclidean distance error than the other networks. This performance discrepancy is primarily attributed to the limited training set of paired examples. Additionally, the used dataset from Narang et al. [18] includes misaligned signals and indenters positions that could not be corrected.

Further, we tested our entire framework on unseen real

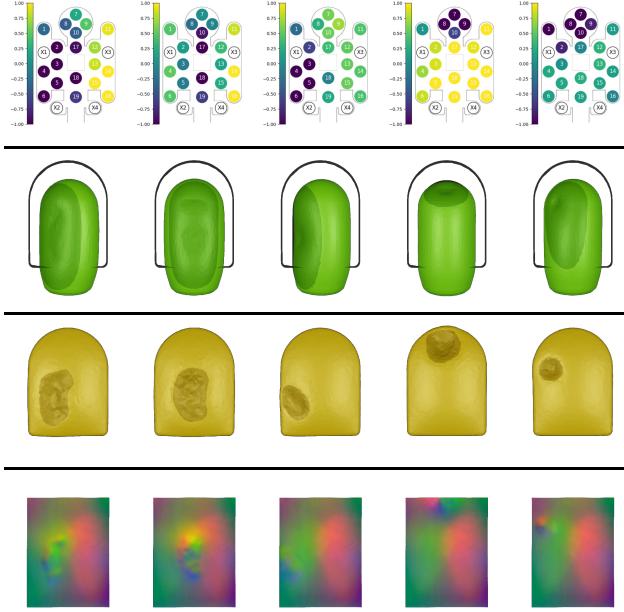


Fig. 5. Converted samples. First row: Real electrode values. Second row: Ground-truth BioTac mesh deformations. The outer frame represents the “unfolded” BioTac surface. Third row: Converted DIGIT mesh deformations. Fourth row: DIGIT output images. The third and fourth rows were generated using the first row as input.

BioTac signal recordings from Narang et al. [19] and converted them to DIGIT output images. Figure 5 shows the qualitative results of five selected BioTac signals that we converted to DIGIT images using our framework.

The spatial positions and depth of the indentations are accurately preserved between the BioTac input signal and the DIGIT output image across all timestamps in the test set, and not only in those examples shown in Figure 5. These accurate results can be verified in the video available on our website. However, the shapes of the indentations are partially reconstructed. This limitation arises due to the lower resolution of the BioTac sensor, which has only 19 electrodes. Compared to the higher-resolution DIGIT sensor output, this sensor cannot capture the detailed contours of indentations.

## V. DISCUSSION AND CONCLUSION

Our novel framework, ACROSS, represents a promising approach for re-utilizing datasets from deprecated sensors, and it enables the exchange of data across different setups. Despite differences in sensor modalities, we successfully demonstrated the framework’s capability to accurately convert previously unseen BioTac sensor data into DIGIT output images, as evidenced by the qualitative examples provided.

ACROSS consists of three steps: The first step involves converting the source sensor inputs into their corresponding 3D deformations. The second step involves mapping these source sensor deformation meshes to the target sensor deformation meshes. Finally, in the third step, we generate the output values based on the resulting meshes.

The core innovation of our framework lies in the 3D mesh deformation conversion between tactile sensors, which

share an inherent similarity. We demonstrate this approach by transferring low-resolution inputs, i.e., BioTac signals, into high-resolution outputs, i.e., DIGIT images. Nevertheless, the lower resolution and differing format of the BioTac sensor compared to the DIGIT sensor may result in the loss of detail that a real DIGIT sensor would capture, such as the precise shape of indentations.

To this end, we offer a dataset featuring paired 3D mesh deformations from BioTac and DIGIT sensors, as well as a DIGIT FEM model for simulating the mesh deformations. However, the current framework has limitations. For instance, given the curvature of the sensor surfaces and shape mismatches, precise alignment of the sensors was essential during data collection to accurately capture side touches on both surfaces. In the current paper, we addressed this by calculating a transformation matrix to align the BioTac surface with the DIGIT surface, involving rotation and translation of the BioTac sensor within its horizontal plane. Hence, mapping physical deformations between sensors with different morphologies and features is inherently challenging. We aim to develop an alternative method for representing the data that does not require calculating transformation matrices for alignment.

The proposed approach relies on modeling the deformation of the sensor, and its effectiveness is closely tied to the accuracy of simulations. Thus, making a validated simulation of this deformation a necessary prerequisite. While this allows the framework to generalize to various tactile sensors, it fundamentally limits its applicability to sensors for which such simulations exist. For example, vibration-based sensors [30], which are considered for tactile sensing, do not undergo deformation in a way that can be meaningfully simulated within this framework. Similarly, other non-deformable sensing technologies, such as capacitive sensors [31], may not fit within this approach.

Future work will focus on improving the signal-to-mesh model by expanding the training dataset and gathering more real-world data. We also plan to quantitatively assess the conversion quality by applying the framework to various tasks, i.e., classification tasks, and explore its adaptability by incorporating additional sensor modalities. Furthermore, by exploring alternative methods, we intend to refine the syntactic data generated by the Taxim algorithm, which currently lacks shadow information for the DIGIT sensor. We aim to improve the framework’s robustness and ability to generalize effectively to other sensors.

## REFERENCES

- [1] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, “Tactile Sensing – From Humans to Humanoids,” *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.
- [2] N. Navarro-Guerrero, S. Toprak, J. Josifovski, and L. Jamone, “Visuo-Haptic Object Perception for Robots: An Overview,” *Autonomous Robots*, vol. 47, no. 4, pp. 377–403, 2023.
- [3] N. Wettels, D. Popovic, V. J. Santos, R. S. Johansson, and G. E. Loeb, “Biomimetic Tactile Sensor for Control of Grip,” in *IEEE International Conference on Rehabilitation Robotics (ICORR)*, ser. 10th, 2007, pp. 923–932.

- [4] N. Wettels, J. A. Fishel, and G. E. Loeb, "Multimodal Tactile Sensor," in *The Human Hand as an Inspiration for Robot Hand Development*, ser. Springer Tracts in Advanced Robotics. Springer International Publishing, 2014, no. 95, pp. 405–429.
- [5] W. Zai El Amri and N. Navarro-Guerrero, "Optimizing BioTac Simulation for Realistic Tactile Perception," in *International Joint Conference on Neural Networks (IJCNN)*, Yokohama, Japan, June 2024, pp. 1–8.
- [6] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts-Escalano, P. Gil, and J. Garcia-Rodriguez, "TactileGCN: A Graph Convolutional Network for Predicting Grasp Stability with Tactile Sensors," in *International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.
- [7] B. S. Zapata-Impata, P. Gil, and F. Torres, "Learning Spatio Temporal Tactile Features with a ConvLSTM for the Direction Of Slip Detection," *Sensors*, vol. 19, no. 3, p. 523, 2019.
- [8] D. Xu, G. E. Loeb, and J. A. Fishel, "Tactile Identification of Objects Using Bayesian Exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3056–3061.
- [9] Y. Chebotar, K. Hausman, Z. Su, A. Molchanov, O. Kroemer, G. Sukhatme, and S. Schaal, "BiGS: BioTac Grasp Stability Dataset," in *ICRA on Workshop on Grasping and Manipulation Datasets*, 2016, p. 2.
- [10] Y. Gong, Y. Xing, J. Wu, and Z. Xiong, "Tactile-Based Slip Detection Towards Robot Grasping," in *Intelligent Robotics and Applications*. Springer Nature, 2023, pp. 93–107.
- [11] W. Zai El Amri, M. Kuhlmann, and N. Navarro-Guerrero, "Transferring Tactile Data Across Sensors," in *40th Anniversary of the IEEE Conference on Robotics and Automation (ICRA@40)*, Rotterdam, The Netherlands, Sept. 2024, pp. 1540–1542.
- [12] 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*, vol. 5, no. 3, pp. 3838–3845, 2020.
- [13] "Isaac Gym - Preview Release," <https://developer.nvidia.com/isaac-gym>.
- [14] R. S. Dahiya and M. Valle, *Robotic Tactile Sensing - Technologies and System*. Springer Netherlands, 2013.
- [15] J. Lee, D. Bollegala, and S. Luo, "'Touching to See' and 'Seeing to Feel': Robotic Cross-Modal Sensory Data Generation for Visual-Tactile Perception," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4276–4282.
- [16] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, "ViTac: Feature Sharing Between Vision and Tactile Sensing for Cloth Texture Recognition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2722–2727.
- [17] G. Tatiya, R. Hosseini, M. C. Hughes, and J. Sinapov, "A Framework for Sensorimotor Cross-Perception and Cross-Behavior Knowledge Transfer for Object Categorization," *Frontiers in Robotics and AI*, vol. 7, 2020.
- [18] Y. S. Narang, B. Sundaralingam, K. Van Wyk, A. Mousavian, and D. Fox, "Interpreting and Predicting Tactile Signals for the SynTouch BioTac," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1467–1487, 2021.
- [19] Y. Narang, B. Sundaralingam, M. Macklin, A. Mousavian, and D. Fox, "Sim-to-Real for Robotic Tactile Sensing Via Physics-Based Simulation and Learned Latent Projections," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6444–6451.
- [20] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering," in *International Conference on Neural Information Processing Systems (NIPS)*, vol. 30th. Curran Associates Inc., 2016, pp. 3844–3852.
- [21] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D Faces Using Convolutional Mesh Autoencoders," in *European Conference on Computer Vision (ECCV)*, ser. LNCS. Munich, Germany: Springer International Publishing, 2018, vol. 11207, pp. 725–741.
- [22] X. Zhu, S. Jain, M. Tomizuka, and J. Van Baar, "Learning to Synthesize Volumetric Meshes from Vision-based Tactile Imprints," in *International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 4833–4839.
- [23] I. H. Taylor, S. Dong, and A. Rodriguez, "GelSlim 3.0: High-Resolution Measurement of Shape, Force and Slip in a Compact Tactile-Sensing Finger," in *International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, May 2022, pp. 10781–10787.
- [24] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra, "TACTO: A Fast, Flexible, and Open-Source Simulator for High-Resolution Vision-Based Tactile Sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3930–3937, 2022.
- [25] "Mmatl/pyrender: Easy-to-use glTF 2.0-compliant OpenGL renderer for visualization of 3D scenes." <https://github.com/mmatl/pyrender>.
- [26] Z. Si and W. Yuan, "Taxim: An Example-Based Simulation Model for GelSight Tactile Sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2361–2368, 2022.
- [27] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *International Conference on Learning Representations (ICLR)*, 2017.
- [28] P. Ruppel, Y. Jonetzko, M. Görner, N. Hendrich, and J. Zhang, "Simulation of the SynTouch BioTac Sensor," in *International Conference on Intelligent Autonomous Systems (IAS)*, ser. Advances in Intelligent Systems and Computing, vol. 867. Springer International Publishing, 2019, pp. 374–387.
- [29] S. Schoenborn, T. Lorenz, K. Kuo, D. F. Fletcher, M. A. Woodruff, S. Pirola, and M. C. Allenby, "Fluid-Structure Interactions of Peripheral Arteries Using a Coupled in Silico and in Vitro Approach," *Computers in Biology and Medicine*, vol. 165, p. 107474, 2023.
- [30] O. A. Juñía Quilachamín and N. Navarro-Guerrero, "A Biomimetic Fingerprint for Robotic Tactile Sensing," in *International Symposium on Robotics (ISR Europe)*. Stuttgart, Germany: VDE Verlag GmbH, Sept. 2023, pp. 112–118.
- [31] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, "Recent Progress in Technologies for Tactile Sensors," *Sensors*, vol. 18, no. 4, p. 948, 2018.