Cross-Sensor Touch Generation

Samanta Rodriguez[*1] **Yiming Dou**[*1,2] **Miquel Oller**[1] **Andrew Owens**[1,2] **Nima Fazeli**[1]
[1]University of Michigan [2]Cornell University
[*]Equal contribution

Abstract:
Today's visuo-tactile sensors come in many shapes and sizes, making it challenging to develop general-purpose tactile representations. This is because most models are tied to a specific sensor design. To address this challenge, we propose two approaches to cross-sensor image generation. The first is an end-to-end method that leverages paired data (Touch2Touch). The second method builds an intermediate depth representation and does not require paired data (T2D2: Touch-to-Depth-to-Touch). Today's visuo-tactile sensors come in many shapes and sizes, making it challenging to develop general-purpose tactile representations. This is because most models are tied to a specific sensor design. To address this challenge, we propose two approaches to cross-sensor image generation. The first is an end-to-end method that leverages paired data (Touch2Touch). The second method builds an intermediate depth representation and does not require paired data (T2D2: Touch-to-Depth-to-Touch). Both methods enable the use of sensor-specific models across multiple sensors via the cross-sensor touch generation process. Together, these models offer flexible solutions for sensor translation, depending on data availability and application needs. We demonstrate their effectiveness on downstream tasks such as in-hand pose estimation and behavior cloning, successfully transferring models trained on one sensor to another. Project page: https://samantabelen.github.io/cross_sensor_touch_generation.

Keywords: Tactile Sensing, Manipulation, Representation Learning

1 Introduction

Tactile sensing is a fundamental enabling technology for dexterous manipulation. Yet, in comparison to their visual counterparts, touch sensors remain highly diverse and lack standardization. For example, the robotics community has demonstrated numerous manipulation capabilities [1, 2, 3] using a variety of vision-based tactile sensors such as GelSight [4], Soft Bubble [5], GelSlim [6], Finger Vision [7], DIGIT [8], and DenseTact [9]. This sensor diversity poses a significant challenge: specialized algorithms must typically be developed and optimized per sensor. These sensor-specific algorithms are difficult to reuse when their corresponding sensors are unavailable, and adapting them to other sensors can be time-consuming and expensive. Moreover, machine learning models trained on one tactile sensor often fail to generalize to other sensors due to significant distribution shifts.

Despite their diversity, vision-based tactile signatures are largely composed of fine-grained shape features [10, 4, 6, 7, 8, 9]. As a result, they convey similar information, such as an object's surface geometry and the contact shape. In this paper, we ask whether this overlapping information can be used to translate tactile signals between sensors, thereby enabling models designed for one tactile sensor to be transferred to another.

We propose two approaches to address this challenge. First, we frame cross-sensor translation as a cross-modal prediction task and train a diffusion model to generate the tactile signal of one sensor conditioned on that of another. We train this model using paired tactile data collected by probing

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

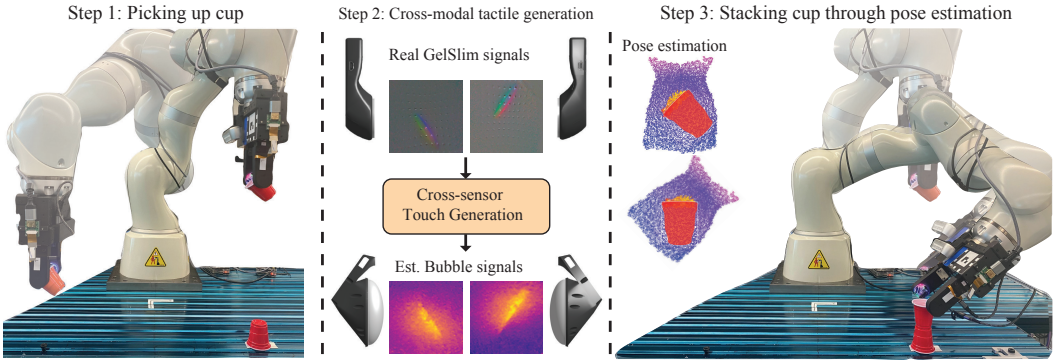


Figure 1: **Transferring manipulation skills between touch sensors via cross-modal prediction.** We execute a manipulation skill designed for one touch sensor (Soft Bubble) on a robot equipped with a different sensor (GelSlim). We demonstrate two approaches to the translation of one touch signal to another — that is, we predict what the object would have felt like if it were manipulated with Soft Bubble rather than GelSlim. The signal is then used for the downstream skill.

the same object location with two different sensors. Second, we introduce an approach that uses shape as an intermediate representation: we first predict a depth map from the source touch signal, then translate that depth into the target sensor's signal. This method does not require paired data and leverages the fact that shape is shared across vision-based tactile signals, thereby facilitating the integration of new sensors into the framework with minimal data collection.

We evaluate our approach on two manipulation tasks: in-hand object pose estimation (Fig. 1) and behavior cloning. In both tasks, the robot has access to measurements from the equipped (source) sensor and uses algorithms designed for a different (target) sensor. Successful execution requires accurate generation of both the target tactile image and the corresponding depth map.

Our contributions are as follows: First, we present two approaches for cross-sensor tactile signal generation: one that uses paired data with a conditional diffusion model to produce fine-grained target signals, and another that leverages depth as an intermediate representation, enabling signal translation without the need for paired data. Second, we evaluate cross-sensor transfer with quantitative and qualitative image generation metrics, and in-hand object pose estimation as a tactile-specific metric. Third, we demonstrate the practical utility of our approach in robotic manipulation scenarios, showing that it enables precise task transfer across heterogeneous tactile sensors.

2 Related Work

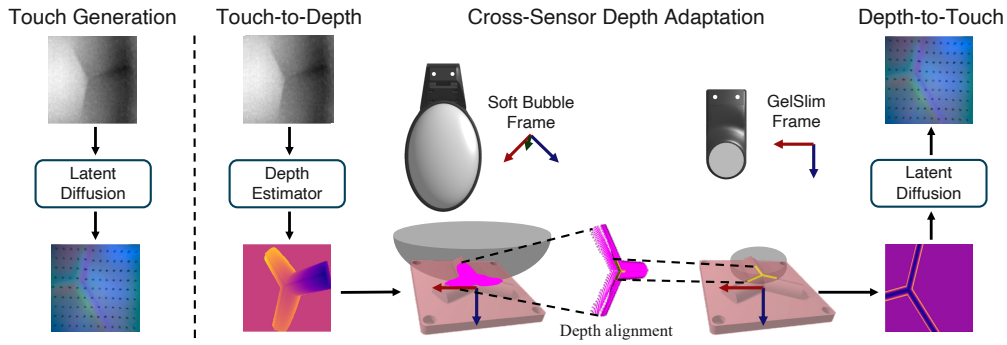
Vision-based tactile sensing. In the last decade, the robotics community has adopted a variety of vision-based tactile sensors, such as GelSight [4, 10], Soft Bubble [5], GelSlim [6], Finger Vision [7], DIGIT [8], and DenseTact [9]. These sensors convert touch signals into vision-like signals, representing touch as 2D images or 3D representations (e.g., point clouds). These sensors are rapidly gaining popularity and have proven valuable in a variety of applications [11, 12, 1, 13, 2]. We use the Soft Bubble [14], DIGIT [3] and GelSlim [15] sensors in our experiments. The Soft Bubble [14] is composed of a thin, highly compliant, air-filled membrane paired with a camera-based depth sensor. Tactile signatures are perceived as deformations of the membrane due to external contacts. The DIGIT [3] and GelSlim [15] measure deformations of an elastomeric skin illuminated by multi-colored LEDs using an RGB camera. We choose these three sensors because of their vastly different deformations and compliance, contact areas, image quality, and 3D (vs. 2D) representation. As for the algorithms, existing manipulation, perception and controls representations are tied to specific touch sensors. For example, a variety of methods leverage sensor specific local geometry, contact force estimation, or texture [16, 13]. Further, in-hand object pose estimation algorithms have been developed for different visuotactile sensors (e.g., for Soft Bubble [14], GelSlim [2], and DIGIT [3]), for local geometry estimation (e.g., Soft Bubble [14], GelSlim [15], and DIGIT [17]). We reduce the need for sensor-specific methods by enabling models to transform one touch signal to another.

Cross-sensor tactile representation. Other work aims to design a unified representation for tactile sensors. UniTouch [18] and AnyTouch [19] align the tactile embeddings with pretrained image and text embeddings, and supports multi-sensor training by sensor-specific tokens. Sparsh [20] and T3 [21] encode images from different sensors into a shared representation space and train the encoders with self-supervised learning. CTTP [22] uses paired tactile data to learn a touch feature space based on contrastive learning. This method requires all downstream methods to operate on a special feature representation. In contrast, we focus on the explicit transfer of raw touch signals, without the need for a latent feature space. This brings two major advantages: (i) we can use it for tasks that require explicit geometry from the raw tactile signals, and (ii) we can apply it to existing downstream touch processing models "zero shot", without adaptation. As an example, we directly apply iterative closest point (ICP) on a generated Soft Bubble image to perform robotic object manipulation tasks (Sec. 4).

Cross-modal generation. A variety of early generative models transformed images from one format to another [23, 24, 25, 26]. Recent works in cross-modal image translation frequently use diffusion [27] for its ability to generate high-quality images with stable training. These models have been used with a variety of different conditioning signals, resulting in models that perform text-to-image [28, 29, 30, 31, 32, 33, 34], audio-to-image [35], video-to-audio [36], etc. Our work is closely related to methods that estimate touch from vision. These works have proposed models under various settings, including desktop [37], object-centric [38], sub-scene [39, 18], and full-scene [40]. Like many of these works [41, 16, 18, 40, 42], we use diffusion to generate touch signals. However, our conditioning is based on the touch signal from another sensor rather than from a visual signal. Our framework transforms across significantly different visuo-tactile sensors, including Soft Bubble (not gel-based), GelSlim, and DIGIT (gel-based). To the best of our knowledge, this work is the first to address cross-sensor generation.

3 Method

In this section, we first describe our data collection processes. Next, we propose two approaches (shown in Fig. 2) for cross-sensor tactile generation: (1) one-stage end-to-end generation using paired touch signals, and (2) two-stage generation by using depth as an intermediary representation.



(a) Touch-to-Touch (b) Touch-to-Depth-to-Touch

Figure 2: **Translating signals between touch sensors.** We investigate two different approaches to cross-sensor touch translation. (a) We train a latent diffusion model to direct predict one sensor's signal from another's, using paired training data. (b) We use depth as an intermediate representation, thus avoiding the need for paired training data. We predict depth from touch, adapt the depth map to match the specifications of another sensor, then generate a touch signal from the resulting depth map. We use the resulting touch translation models for robotic manipulation tasks.

3.1 Dataset

We use a robot to collect paired touch data, such that two different sensors probe the same physical location. We also obtain touch paired with depth data, which we use for a model that predicts depth as an intermediate representation for touch translation.

Collecting Paired Multimodal Tactile Signals. We use a KUKA LBR Med 14 R820 robotic arm with a WSG-50 gripper to collect paired tactile signals from two distinct sensors: Soft Bubble and GelSlim. These sensors differ significantly in mechanical design, contact compliance, contact area, and spatial resolution, necessitating a highly repeatable and spatially precise setup for generating semantically aligned tactile pairs. To ensure alignment and both sensors capture meaningful geometric features under consistent conditions the gripper is positioned so that the center of each sensor contacts the same point on the target object, and compliance differences are compensated by adjusting the gripper. Paired signals must share *mutual semantic information*—that is, both must capture recognizable features despite differences in spatial resolution and contact area. For example, the Soft Bubble sensor covers roughly $16\times$ more surface area than the GelSlim but at a much lower resolution (2.36 vs. 23.72 pixels/mm). To address this, we carefully select objects with distinctive geometry (e.g., the elbow of a hex key) and post-process the data to ensure that both signals contain salient, recognizable features within their respective sensing capabilities.

A Dataset of Multimodal Touch-Depth Pairs. To enable supervised learning for both tactile-to-depth and depth-to-tactile translation, we construct a dataset of paired tactile and depth observations following a procedure inspired by [43]. Data is collected using a robot equipped with a variety of vision-based tactile sensors, including Soft Bubbles, GelSlims, DIGITs, and dotless GelSlim variants, offering diverse tactile imaging characteristics. We use 12 indenters of known geometry, spanning flat, curved, and angled profiles. Each indenter is pressed into the sensor surface within a controlled 3D sampling grid covering $10, \mathrm{mm} \times 10, \mathrm{mm}$ in translation and up to 45° in orientation, centered at the indenter's origin. This ensures uniform spatial coverage and varied contact patterns. For each indenter, we collect 60 tactile samples at distinct poses, resulting in a total of 720 samples per sensor. Each tactile observation is paired with a ground-truth depth map that captures the 3D contact surface. These depth maps are generated using the robot's proprioceptive data, calibrated camera models, and geometric rendering based on the known object and contact poses.

3.2 One-Stage Generation using Paired Touch Signals

For the one-stage approach, we train a cross-modal diffusion model to directly translate from one sensor to another based on paired tactile images. We refer to this method as touch-to-touch (T2T). For the implementation, we use a generative model based on latent diffusion [32] to generate the touch signal of the target sensor by conditioning on the source measurement (Fig. 2). We use a ResNet-18 [44] to encode the touch images from the source sensor into a 2D feature map. The feature map is then concatenated with the noise signal and passed into the denoising UNet.

We specifically apply our model to translating from the GelSlim to Soft Bubble sensors (and vice versa). When the Soft Bubble sensor is used as the target sensor, we inflate the 1-channel signals into 3-channel signals by tiling the image channel-wise. When the Soft Bubble tactile signal is generated from the GelSlim image using diffusion, we perform three post-processing steps to ensure accurate tactile information generation (beyond visual fidelity). First, we take the average value of the three channels of the prediction to map it back to one channel. Next, we normalize the prediction (in the range $[-1, 1]$) back to the depth map values by using the maximum and minimum values of the depth maps across the training dataset. Finally, to deal with small scaling/bias in the predictions (which can drastically change their interpretations as point clouds), we shift the pixel values of the generated Soft Bubble images. We calculate the mean and standard deviation of the generated and ground truth Soft Bubble images on the training dataset and use these values to renormalize the generated images.

In our experiments, we compare our end-to-end cross-sensor generation model to VQ-VAE, a commonly used baseline for cross-modal generation. We provide implementation details in the supplementary material.

3.3 Two-Stage Generation using Unpaired Touch Signals

The one-stage method can be trained end-to-end, but it requires paired touch signals, which can be difficult to collect in practice. Therefore, we also propose a two-stage framework that translates touch to a depth map as an intermediate representation. This "touch-to-depth-to-touch" (T2D2) approach enables training without paired touch signals. This model contains three modules: (i) a depth estimation model that predicts a depth map from the source tactile image, (ii) a depth adaptation stage that modifies the source sensor depth map to match the specifications of target sensor and (iii) a tactile image generation model that synthesizes the target sensor output based on the adapted depth map.

Depth estimation model. Our goal is to estimate a depth map from a touch signal. This requires capturing both the contact geometry of the source sensor and the geometry of the object surface. We adapt Depth Anything V2 [45], a state-of-the-art monocular depth estimation model, to this task.

We modify the model to jointly estimate both the depth map and a contact mask, which indicates the sensor parts that are in contact with the object. To do this, we add a decoder head that takes intermediate high-resolution features from the depth decoder to predict a binary contact map. We train the model on real touch signals paired with ground-truth depth maps, which are obtained from a neural tactile de-rendering method [43] (Sec. 3.1).

Depth adaptation. Before converting a given depth map into a touch signal, we need to determine which parts of it are visible (e.g., due to a sensor's limited field of view) and to transform it into the coordinate system of the sensor (e.g., accounting for the fact the pose of the vision-based sensor's camera). We call this process *depth adaptation*. For the adaptation stage, we use the depth map D'_S and its corresponding binary mask M'_S obtained from our depth estimation model to find the equivalent sensor-specific depth map D''_T and mask M''_T as if the target sensor was in contact.

To perform this adaptation, we first determine the set of pixels Ω that are indicated by the predicted mask M'_S to be in contact with the sensor. For each such pixel $(u, v) \in \Omega$, we transform it as follows. We back-project the depth value into 3D space using the vision-based touch sensor's camera intrinsic matrix K_S^{-1} and transform the resulting 3D point from the source sensor frame to the target sensor frame using the rigid transformation $T_{S\to T}$. Specifically, $T_{S\to T}$ is defined as the composition of the transformation from the source sensor to the alignment frame, $T_{S\to A}$, and from the alignment frame to the target sensor, $T_{A\to T}$; that is, $T_{S\to T} = T_{A\to T} \circ T_{S\to A}$. The alignment frame is defined as a common reference frame shared across different tactile sensors. This standardization allows for consistent interpretation of the contact geometry, regardless of each sensor's unique geometry or pose. This process is expressed as:

$$\mathcal{P}_T = \left\{ T_{S\to T} \left(D'_S(u,v) \cdot K_S^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \right) : (u,v) \in \Omega \right\}, \tag{1}$$

where $\mathcal{P}_T$ represents the resulting point cloud in the target sensor frame. We then use $\mathcal{P}_T$ to obtain the target sensor's depth map and its contact mask. We provide details in the supplementary material.

Generating touch from depth. Given a sensor-specific depth map, we generate a corresponding touch signal using diffusion. To do this, we use a model very similar to the one described in Sec. 3.2, but with depth conditioning in lieu of touch conditioning.

4 Experiments

We evaluate cross-sensor touch generation using visual metrics, a tactile-specific metric, and downstream robotic tasks.

4.1 Evaluation Metrics

Visual Metrics. The visual quality of the generated tactile images is assessed with three standard image-based metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Fréchet Inception Distance (FID). PSNR and SSIM measure pixel-level fidelity

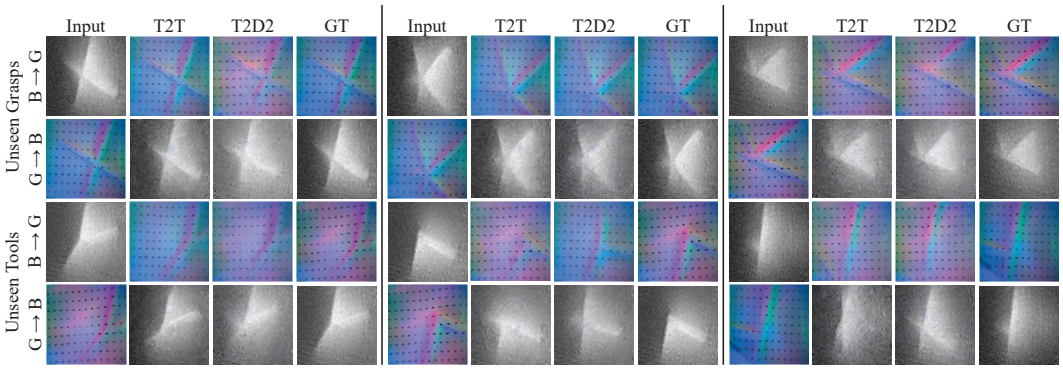


Figure 3: **Generation Qualitative Results**. Qualitative results for unseen grasps and tools using T2T and T2D2. Rows indicate sensor transfer directions (B → G, G → B); columns show input, model outputs, and ground truth.

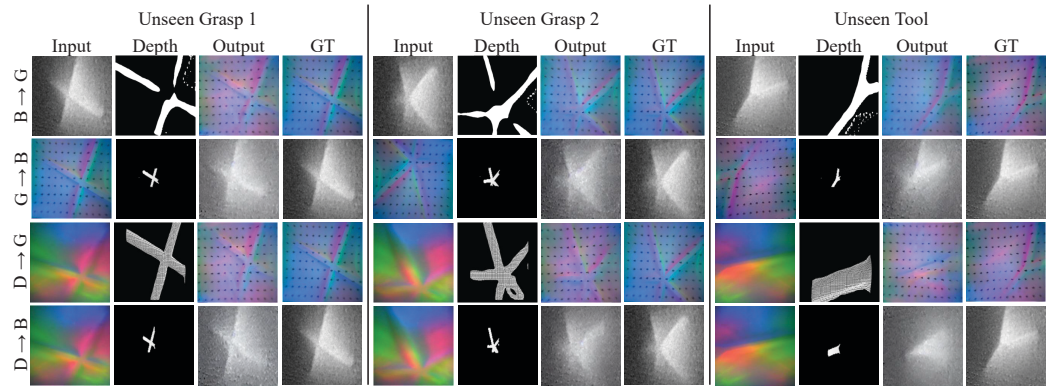


Figure 4: **T2D2 Qualitative Results**. Evaluation of T2D2 model on unseen grasps and tools. Each block shows the input, adapted depth map, generated tactile output, and ground truth (GT) for various sensor transfers.

and perceptual similarity to the ground truth, respectively, while FID quantifies distributional alignment between generated and real samples.

Tactile-Specific Metric. As our tactile-specific metric, we use in-hand object pose estimation, where a neural network trained on the target sensor tactile images is applied to source sensor images translated by our method. Performance is quantified by translation and angular pose errors.

Downstream Robotic Tasks. We further evaluate cross-sensor touch generation on two downstream robotic tasks: peg-in-hole insertion and marble rolling. We perform the peg-in-hole insertion task, originally designed for the Soft Bubble sensor, using a robot equipped with GelSlim sensors. Our T2T model predicts the corresponding Soft Bubble signal from the GelSlim reading, which is then used to estimate object pose via Iterative Closest Point (ICP) alignment. This pose estimate enables the robot to complete tool insertion and cup stacking with previously unseen objects, as illustrated in Fig. 1. For the marble rolling task, we train a behavior cloning policy to roll a marble—initially placed at a random position—toward the center of the GelSlim sensor image. The policy is trained using supervised learning on GelSlim tactile data and learns to guide the marble using the tactile feedback provided by the sensor. We then test the transferability of this policy to a different sensor, DIGIT, by using our T2D2 model to generate the corresponding GelSlim tactile signal from the DIGIT input. This allows the original policy, trained exclusively on GelSlim data, to be applied to DIGIT inputs without any retraining or modification. This process is shown in Fig. 5.

4.2 Visual and Tactile Metrics Results

The results in Table 1 and Table 2 show that while T2D2 enables cross-sensor tactile generation using unpaired data and a depth-based intermediate representation, this flexibility comes at the cost of fine-grained accuracy necessary for downstream tasks like pose estimation. In contrast, T2T,

which performs direct image-to-image translation, preserves more structural fidelity, resulting in lower translation and angular errors. Fig. 3 shows qualitative results for both methods.

This trend is also reflected in the visual metrics. T2T consistently achieves higher PSNR, SSIM, and lower FID scores than T2D2, indicating sharper, more perceptually accurate, and distributionally faithful outputs. For example, in the Bubbles to GelSlims transfer on unseen grasps, T2T achieves a PSNR of 30.92 and SSIM of 0.93, compared to 21.69 and 0.83 for T2D2. Similar trends are observed across unseen tools as well.

While PSNR values are relatively comparable between the GelSlims to Bubbles and Bubbles to GelSlims translations, all other metrics indicate that translating from GelSlims to Bubbles is a more challenging task. This is especially evident in the angular and translation errors, which are consistently higher in the GelSlims to Bubbles direction for both models. A key reason is that the Bubble sensor is considerably larger than the GelSlim sensor, requiring the models to effectively learn to outpaint or infer tactile signals beyond the spatial extent of the input. In contrast, translating from Bubble to GelSlim remains within the bounds of the original signal, making the task less ambiguous.

T2D2 also shows an increased error when transferring from Digits to GelSlims and from Digits to Bubbles, reflecting the added challenge of translating from a structurally different sensor. Still, qualitative results in both directions are strong, indicating that the generated signals retain coherent tactile features, shown in Fig. 4. Incorporating Digits into the T2D2 pipeline was efficient, as the model supports unpaired training, and fine-tuning the depth estimator required collecting only about one-tenth of the paired data typically needed to train an end-to-end model. These results demonstrate the scalability of T2D2 for incorporating new sensors with minimal supervision.

Transfer	**Model**	**PSNR** ↑	**SSIM** ↑	**FID** ↓	**Trans. Error [mm]** ↓	**θ Error [$^\circ$]** ↓
Bubbles GT	-	-	-	-	0.22 ± 0.12	0.51 ± 0.41
GelSlims → Bubbles	T2T	25.85	0.59	135.66	0.92 ± 0.58	1.87 ± 1.23
GelSlims → Bubbles	T2D2	20.51	0.52	122.06	3.23 ± 1.73	11.53 ± 9.09
Digits → Bubbles	T2D2	20.53	0.48	111.77	3.38 ± 1.86	9.85 ± 8.10
GelSlims GT	-	-	-	-	0.29 ± 0.17	0.83 ± 0.59
Bubbles → GelSlims	T2T	30.92	0.93	33.25	0.40 ± 0.29	1.27 ± 0.97
Bubbles → GelSlims	T2D2	21.69	0.83	53.99	2.63 ± 1.93	8.47 ± 6.82
Digits → GelSlims	T2D2	17.73	0.72	65.97	4.10 ± 1.93	17.67 ± 11.47

Table 1: **Unseen Grasps**. Evaluation metrics for cross-modal tactile generation tasks, including visual (PSNR, SSIM, FID) and tactile-specific (translation and angular error) metrics.

Transfer	**Model**	**PSNR** ↑	**SSIM** ↑	**FID** ↓	**Trans. Error [mm]** ↓	**θ Error [$^\circ$]** ↓
Bubbles GT	-	-	-	-	0.42 ± 0.26	1.03 ± 0.90
GelSlims → Bubbles	T2T	21.77	0.48	167.72	3.40 ± 1.82	14.00 ± 8.13
GelSlims → Bubbles	T2D2	18.70	0.45	134.74	4.40 ± 2.14	12.71 ± 9.57
Digits → Bubbles	T2D2	18.91	0.46	129.30	3.63 ± 2.29	14.75 ± 13.83
GelSlims GT	-	-	-	-	0.78 ± 0.52	1.71 ± 1.95
Bubbles → GelSlims	T2T	21.50	0.81	62.38	2.53 ± 1.56	6.35 ± 8.43
Bubbles → GelSlims	T2D2	18.33	0.73	68.67	3.58 ± 1.90	9.99 ± 8.08
Digits → GelSlims	T2D2	17.37	0.68	105.30	4.87 ± 2.18	13.07 ± 9.73

Table 2: **Unseen Tools**. Evaluation metrics for cross-modal tactile generation tasks, including visual (PSNR, SSIM, FID) and tactile-specific (translation and angular error) metrics.

4.3 Downstream Robotic Task Results

For the peg-in-hole insertion task, we compare T2T with another end-to-end generative method, VQ-VAE. In Table 3, T2T consistently outperforms VQ-VAE across all tasks, achieving higher success rates in both tool-based (unseen tools from our dataset) and real-object scenarios. The largest improvements are observed in pencil insertion (21/30 vs. 7/30) and tool 1 insertion (18/30 vs. 9/30),

demonstrating T2T's effectiveness in generating accurate cross-modal tactile signals for manipulation. For the marble rolling task, the policy succeeds in **15 out of 20 trials** when using DIGIT inputs translated to GelSlim, compared to **20 out of 20** using real GelSlim signals. This experiment demonstrates that our cross-sensor generation framework supports not only perception tasks but also simple sensor-conditioned control, enabling policy reuse across different tactile hardware.

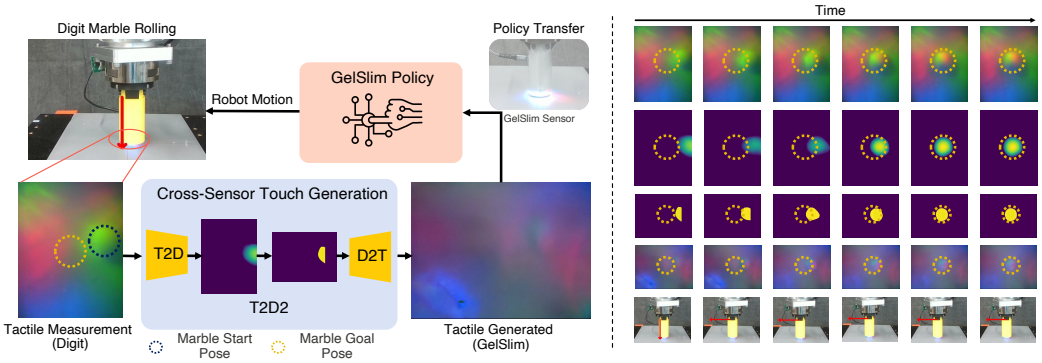


Figure 5: **Marble rolling policy transfer via T2D2.** We train behavior-cloning policy on GelSlim tactile images to roll a marble from random starts to the image center. At test time on DIGIT, we translate each DIGIT tactile signature to its GelSlim counterpart with T2D2 and run the same policy unchanged. **Left:** pipeline—DIGIT → (T2D2) → GelSlim → GelSlim-trained policy. **Right:** transferred roll-outs on DIGIT converge to the center (zero-shot; no retraining).

Method	Tool 1 Insertion	Tool 2 Insertion	Tool 3 Insertion	Pencil Insertion	Cup Stacking
VQ-VAE	9/30	8/30	15/30	7/30	21/30
T2T	18/30	10/30	15/30	21/30	22/30

Table 3: Insertion and Cup Stacking Tasks Success on Unseen Objects

5 Conclusion

Our exploration of cross-sensor tactile generation reveals several key insights about the nature of visuo-tactile sensing and the utility of generative models in bridging sensor-specific differences. One of the most important observations is that, despite large differences in hardware design and signal modalities, tactile data from different sensors often encodes fundamentally similar geometric and contact information. This latent consistency enables translation between sensor outputs, allowing models trained for one sensor to generalize to others through learned generative mappings.

The performance of the two proposed approaches—T2T and T2D2—also underscores the tradeoff between data requirements and model fidelity. T2T, which leverages paired data, consistently generates high-fidelity, structurally accurate tactile signals, achieving strong performance on both visual and tactile-specific metrics. In contrast, T2D2 offers greater flexibility by requiring only unpaired data and an intermediate depth representation. However, this flexibility introduces challenges in preserving high-frequency geometry and leads to degradation in downstream performance, particularly for precision-sensitive tasks like in-hand pose estimation.

Another important outcome of our work is the demonstration that generative translation between sensors is a viable path to sensor interoperability. Instead of attempting to design universal representations or train all models jointly across multiple sensors, we show that translating touch from one sensor to another can unlock existing, optimized pipelines without any retraining. This capability lowers the barrier to reuse in tactile perception and manipulation systems and opens new directions for modular tactile intelligence, where sensor-specific capabilities can be shared or borrowed on demand via generative translation.

6 Limitations

While our study demonstrates the feasibility of cross-sensor tactile generation, several limitations remain. First, the T2T model requires precisely paired tactile data across sensors, which can be time-consuming and hardware-intensive to collect, especially when working with sensors that differ significantly in contact geometry, field of view, or resolution. Although T2D2 addresses this by operating with unpaired data, its reliance on accurate depth estimation and calibration introduces sensitivity to errors in alignment and sensor-specific intrinsic parameters.

Second, both approaches assume that there exists sufficient semantic overlap between the source and target signals—i.e., that the contact patches capture comparable geometric features. This assumption may not hold when sensors have drastically different contact modalities or when the object geometry is highly complex or non-rigid. In such cases, generation fidelity may suffer due to the model having to hallucinate large, unobserved regions, particularly when translating from a smaller to a larger contact area (e.g., GelSlim to Bubble).

Finally, while our models successfully transfer across three sensor types and demonstrate strong downstream task performance, they are limited to vision-based tactile sensors and specific manipulation settings. Extending this framework to other tactile modalities (e.g., resistive, magnetic, or proprioceptive sensors), or to more dynamic or continuous interactions, may require significant architectural adaptations. As tactile sensing systems continue to diversify, generalization across modalities—not just across hardware—remains an open and challenging frontier.

Acknowledgments

This work was supported in part by the NSF GRFP (Award No. 2241144), the NSF CAREER program (Award Nos. 2339071 and 2337870), and the NSF NRI (Award No. 2220876). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Oller, M. P. i Lisbona, D. Berenson, and N. Fazeli. Manipulation via membranes: High-resolution and highly deformable tactile sensing and control. In *Conference on Robot Learning*, pages 1850–1859. PMLR, 2023.

[2] S. Kim and A. Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10241–10247. IEEE, 2022.

[3] S. Suresh, Z. Si, S. Anderson, M. Kaess, and M. Mukadam. Midastouch: Monte-carlo inference over distributions across sliding touch. In *Conference on Robot Learning*, pages 319–331. PMLR, 2023.

[4] W. Yuan, S. Dong, and E. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17:2762, 11 2017. doi:10.3390/s17122762.

[5] A. Alspach, K. Hashimoto, N. Kuppuswamy, and R. Tedrake. Soft-bubble: A highly compliant dense geometry tactile sensor for robot manipulation. *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 597–604, 2019. URL http://arxiv.org/abs/1904.02252.

[6] E. Donlon, S. Dong, M. Liu, J. Li, E. H. Adelson, and A. Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. *CoRR*, abs/1803.00628, 2018. URL http://arxiv.org/abs/1803.00628.

[7] A. Yamaguchi. Fingervision for tactile behaviors , manipulation , and haptic feedback teleoperation. 2018.

[8] M. Lambeta, P.-W. Chou, S. Tian, B. Yang, B. Maloon, V. R. Most, D. Stroud, R. Santos, A. Byagowi, G. Kammerer, et al. 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, 2020.

[9] W. K. Do and M. Kennedy. Densetact: Optical tactile sensor for dense shape reconstruction. *2022 International Conference on Robotics and Automation (ICRA)*, pages 6188–6194, 2022.

[10] M. K. Johnson and E. H. Adelson. Retrographic sensing for the measurement of surface texture and shape. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1070–1077. IEEE, 2009.

[11] 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*, 2018.

[12] Digit tactile sensor - gelsight. URL https://www.gelsight.com/product/digit-tactile-sensor/.

[13] R. Li, R. Platt, W. Yuan, A. Ten Pas, N. Roscup, M. A. Srinivasan, and E. Adelson. Localization and manipulation of small parts using gelsight tactile sensing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3988–3993. IEEE, 2014.

[14] N. Kuppuswamy, A. Castro, C. Phillips-Grafflin, A. Alspach, and R. Tedrake. Fast model-based contact patch and pose estimation for highly deformable dense-geometry tactile sensors. *IEEE Robotics and Automation Letters*, 5(2):1811–1818, 2019.

[15] 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 *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787. IEEE, 2022.

[16] F. Yang, J. Zhang, and A. Owens. Generating visual scenes from touch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22070–22080, 2023.

[17] W. Xu, Z. Yu, H. Xue, R. Ye, S. Yao, and C. Lu. Visual-tactile sensing for in-hand object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8803–8812, 2023.

[18] F. Yang, C. Feng, Z. Chen, H. Park, D. Wang, Y. Dou, Z. Zeng, X. Chen, R. Gangopadhyay, A. Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. *arXiv preprint arXiv:2401.18084*, 2024.

[19] R. Feng, J. Hu, W. Xia, T. Gao, A. Shen, Y. Sun, B. Fang, and D. Hu. Anytouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. *arXiv preprint arXiv:2502.12191*, 2025.

[20] C. Higuera, A. Sharma, C. K. Bodduluri, T. Fan, P. Lancaster, M. Kalakrishnan, M. Kaess, B. Boots, M. Lambeta, T. Wu, et al. Sparsh: Self-supervised touch representations for vision-based tactile sensing. *arXiv preprint arXiv:2410.24090*, 2024.

[21] L. W. Jialiang Zhao1, Yuxiang Ma2 and E. H. Adelson1. Transferable tactile transformers for representation learning across diverse sensors and tasks. *arXiv preprint arXiv:2406.13640v1*, 2024.

[22] S. Rodriguez, Y. Dou, W. v. d. Bogert, M. Oller, K. So, A. Owens, and N. Fazeli. Contrastive touch-to-touch pretraining. *arXiv preprint arXiv:2410.11834*, 2024.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[24] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2017.

[25] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR, 2017.

[26] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[27] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2015.

[28] O. Avrahami, D. Lischinski, and O. Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022.

[29] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[30] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[31] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

[32] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[33] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[34] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[35] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[36] S. Luo, C. Yan, C. Hu, and H. Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[37] 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.

[38] R. Gao, Y. Dou, H. Li, T. Agarwal, J. Bohg, Y. Li, L. Fei-Fei, and J. Wu. The objectfolder benchmark: Multisensory learning with neural and real objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17276–17286, 2023.

[39] 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.

[40] Y. Dou, F. Yang, Y. Liu, A. Loquercio, and A. Owens. Tactile-augmented radiance fields. *arXiv preprint arXiv:2405.04534*, 2024.

[41] C. Higuera, B. Boots, and M. Mukadam. Learning to read braille: Bridging the tactile reality gap with diffusion models. *arXiv preprint arXiv:2304.01182*, 2023.

[42] G. M. Caddeo, A. Maracani, P. D. Alfano, N. A. Piga, L. Rosasco, and L. Natale. Sim2real bilevel adaptation for object surface classification using vision-based tactile sensors. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15128–15134. IEEE, 2024.

[43] J. A. Eyzaguirre, M. Oller, and N. Fazeli. Tactile neural de-rendering. *arXiv preprint arXiv:2409.13923*, 2024.

[44] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[46] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[47] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[48] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

A Appendix

A.1 Implementation Details

Diffusion Model Our implementation of the diffusion model closely follows Stable Diffusion [32], with the difference that we use a ResNet-50 to generate the GelSlim encoding from GelSlim images for conditioning.

The model is optimized for 30 epochs by Adam [46] optimizer with a base learning rate of 10^{-5}. The learning rate is scaled by gpu number $\times$ batch size. We train the model with batch size of 48 on 4 NVIDIA A40 GPUs.

At inference time, the model conducts 200 steps of the denoising process with a 2.54 guidance scale.

VQ-VAE We use a VQ-VAE architecture similar to the one proposed by Van den Oord et al [47] for the style transfer. Before training VQ-VAE, we processed the sensor images by obtaining the difference between the deformed image at the moment of contact with the object and the undeformed image. In addition, we resize these images from the sensor images' original size to 128x128 and keep their corresponding numbers of channels. The input to our model is a 128x128 subtracted Gelslim RGB image, and the output is the corresponding 128x128 subtracted depth map Soft Bubbles image. The input image x is passed through a CNN encoder to generate a vector in the latent space z. This latent vector is then quantized via a collection of discrete vectors known as the *codebook*, such that $z_e(x)$ is transformed into $z_q(x)$. This quantized latent vector is passed through a CNN decoder to generate the final image $\tilde{x}$. The encoder parameters, quantization codebook vectors, and decoder parameters are all learned such that mean squared-error in the latent space quantizations and output reconstructions are minimized.

A.2 Depth Esimation Details

We map a tactile image I_S to a depth map D'_S and a contact mask M'_S. For the depth map, we minimize the scale-invariant logarithmic loss [48, 45]:

$$\mathcal{L}_{\text{silog}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log D_{S,i} - \log D'_{S,i}\right)^2 - \lambda\left(\frac{1}{n}\sum_{i=1}^{n}\left(\log D_{S,i} - \log D'_{S,i}\right)\right)^2}, \tag{2}$$

which measures the discrepancy between the logarithms of the ground truth depth D_S and the estimated depth D'_S. We simultaneously train the model to estimate the contact mask, M'_S, using binary cross entropy loss.

A.3 Depth Adaptation Stage Details

For the adaptation stage, we use the depth map D'_S and its corresponding binary mask M'_S from the source sensor tactile image I_S obtained with our Depth Estimation Model to find the equivalent depth map D''_T and mask M''_T as if the target sensor was in contact.

First, we define the set of valid pixel coordinates, using the contact mask M'_S, as follows:

$$\Omega = \{(u, v) \mid M'_S(u, v) = 1\} \tag{3}$$

For each valid pixel $(u, v) \in \Omega$, we back-project the depth value into 3D space using the inverse intrinsic matrix K_S^{-1} and subsequently transform the resulting 3D point from the source sensor frame to the target sensor frame using the rigid transformation $T_{S\to T}$. Here, $T_{S\to T}$ is defined as the composition of the transformation from the source sensor to the alignment frame, $T_{S\to A}$, and from the alignment frame to the target sensor, $T_{A\to T}$; that is, $T_{S\to T} = T_{A\to T} \circ T_{S\to A}$.The alignment frame is defined as a common reference frame shared across different tactile sensors, representing the same touch event. This standardization allows for consistent interpretation of the contact geometry, regardless of each sensor's unique geometry or pose. This process is expressed as follows.

$$\mathcal{P}_T = \left\{ T_{S \to T} \left(D'_S(u,v) \cdot K_S^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \right) : (u,v) \in \Omega \right\}. \tag{4}$$

Where, $\mathcal{P}_T$ represents the resulting point cloud in the target sensor frame. Then we find the target sensor depth map and its mask, defined as:

$$D''_T(u,v) = \begin{cases} Z, & \text{if there exists } p = (X,Y,Z)^\top \in \mathcal{P}_T \text{ such that } \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z} K_T\, p \\ & \text{and } (u,v) \text{ is within the image bounds,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

$$M''_T(u,v) = \begin{cases} 1, & \text{if there exists } p \in \mathcal{P}_T \text{ such that } p \text{ projects to } (u,v) \\ & \quad \text{and } \mathrm{SDF}(p, \mathcal{M}_{\text{target}}) \leq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

A.4 Sensor Alignment Details

Figure 6 shows each sensor's main coordinate frames: GelSlim camera frames (gR, gL), Soft Bubbles camera frames (BR, BL), and grasp frame (G). For our Touch2Touch dataset, we align both sensors by locating the grasp frame of each at the same section of the manipulated object to obtain paired tactile signatures. In addition, we keep both sensors at a distance D, shown in Figure 6. The final step to align the tactile signatures is to rotate one sensor image by 180°. This rotation is necessary for the grasp frames to be aligned. We can see the grasp frames projection in the image plane of each sensor in Figure 6 and notice the need to rotate for alignment. This figure also shows the difference in size between the sensors' images in pixel space and millimeters.

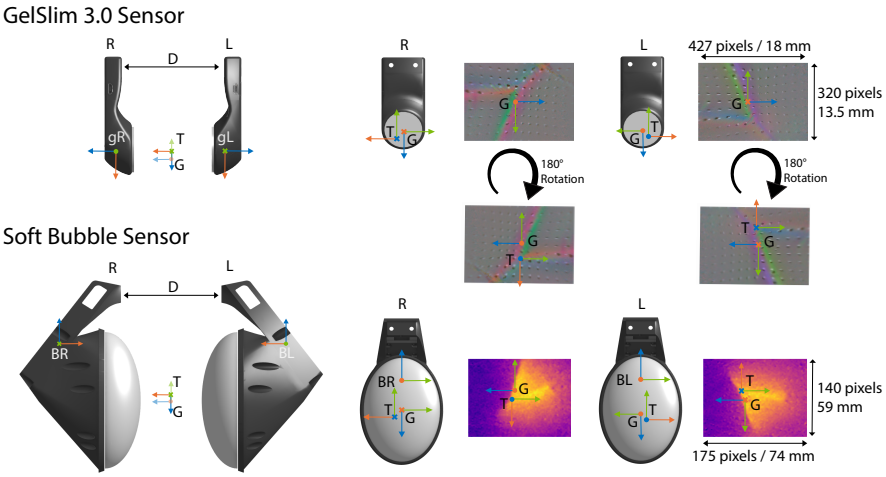


Figure 6: **GelSlim and Soft Bubble Alignment.** This figure shows each sensor's main coordinate frames: GelSlim camera frames (gR, gL), Soft Bubbles camera frames (BR, BL), grasp frames (G), and tool frames (T). D corresponds to the distance we keep between the same type of sensors during data collection. We can see the grasp frames projection in the image plane of each sensor and notice the need to rotate for alignment. For each coordinate frame, the x-axis is shown in red, the y-axis is shown in green and the z-axis is shown in blue.

A.5 Data Collection Tools

Figure 7 shows the geometries of both the seen and unseen tools during the training of generative models. The unseen tools are designed to contain geometric features that are distinct from those of the training tools.

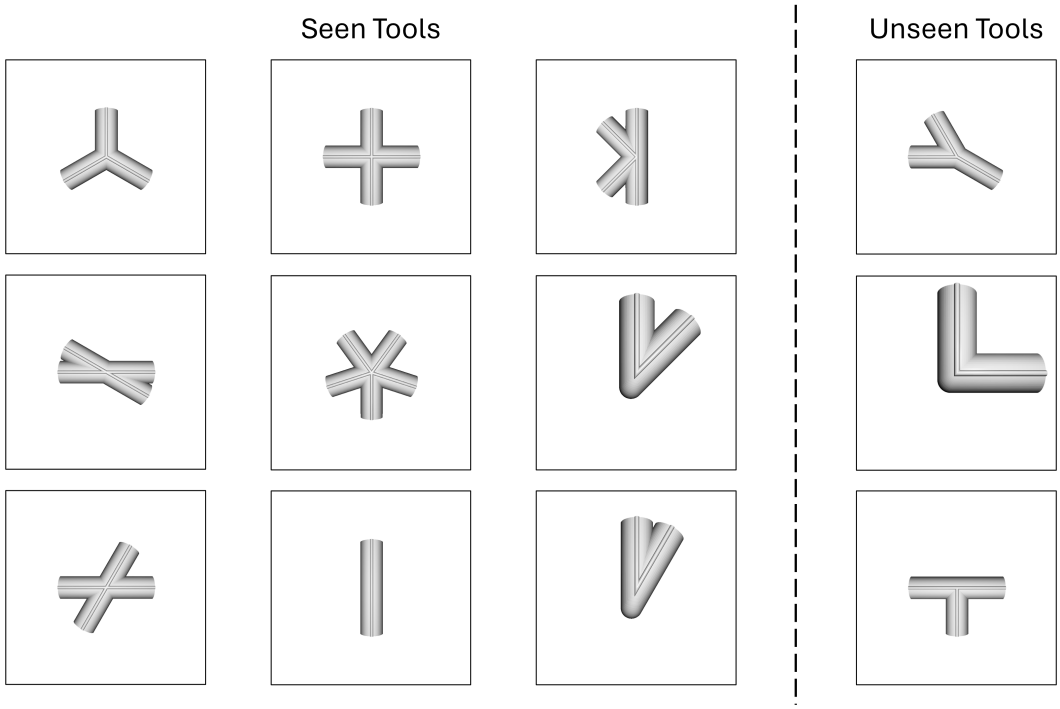


Figure 7: **Dataset Tools.** The left side of the image shows the geometries of the tools that were seen during training of the generative models. The right side of the image shows the geometries of tools that were not seen during training of the generative model.