TEMU - TACTILE ENCODER FOR MANIPULATION: R2. BASELINE AND PROPOSAL

Bhaswanth Ayapilla * **Aayush Fadia*** **Megan Lee*** **Parth Singh*** **Ranai Srivastav***
{bayapill, afadia, meganlee, parthsin, ranais}@andrew.cmu.edu

1 INTRODUCTION

We present TEMU (Tactile Encoder for Manipulation), a multimodal learning framework for predicting slip during robotic grasping. Slip detection is a critical capability for robust manipulation since a robot must reliably identify when an object is about to slip from its grasp in order to correct its grip before failure occurs.

Our work is grounded on the PoseIt (Kanitkar et al., 2022) dataset, which captures multi-sensor recordings of a robot gripper grasping and holding a variety of objects across 16 different grasp configurations. The dataset comprises of the following modalities:

1. Tactile images: captured via GelSight tactile sensor
2. 3 distinct RGB camera images: full view, side view, and top view
3. Force/Torque (F/T) sensors: measures the three-dimensional forces and toeques exerted on the gripper during grasping
4. Gripper pose
5. Robot arm joint positions and velocities

2 BASELINE MODELS AND METRICS (2 PAGES)

All models in this section were implemented and trained entirely by our team. The PoseIt dataset is specialized enough that no public repositories or pretrained checkpoints exist that could be directly applied to our setup, so everything was built from scratch. That said, the ideas and architectural choices for the simple multimodal and competitive baselines were informed by existing literature, particularly the original PoseIt paper (Kanitkar et al., 2022) and related work on multimodal fusion for manipulation (Li et al. (2018), Higuera et al. (2024), Mandil et al. (2024), Nazari et al. (2022)), with all training and evaluation carried out independently.

2.1 [0.5 POINTS] UNIMODAL BASELINES

We evaluate each sensing modality independently to understand its individual predictive power before attempting any fusion. These baselines serve two purposes:

- They establish a floor for what each modality contributes on its own
- They help us diagnose modality imbalance in later fusion experiments

2.1.1 TACTILE ONLY (RESNET-50 + MLP)

We use a frozen ResNet-50 (He et al., 2015) backbone (pretrained on ImageNet) to encode the GelSight tactile contact image at each timestep into a 2048-dimensional feature vector. Features across all valid timesteps in the last 3 seconds of the grasp are aggregated via masked mean pooling, and the resulting representation is passed through a two-layer MLP (2048 $\rightarrow$ 128 with ReLU activation and Dropout of 0.3 $\rightarrow$ 2) to produce a binary stability prediction. Although tactile images directly encode contact geometry and shear deformation patterns, a key limitation is that the ResNet-50 backbone

* Everyone Contributed Equally

is pretrained on ImageNet which is a diverse set of natural images whose low-level features may not transfer well to GelSight imagery. To help tackle this, Li et al. (2018) proposes to pre-process the tactile image by subtracting the pre-contact tactile image from all frames in the sequence and feeding the resulting frames through the pretrained ResNet-50 backbone. The model now no longer needs to learn to ignore irrelevant background texture, and the deformation signal is more cleanly isolated.

2.1.2 External RGB Camera Only (ResNet-50 + MLP)

A frozen ResNet-50 encoder processes the external RGB camera view at each timestep, producing a 2048-dimensional feature vector. Temporal aggregation and classification follow the same masked mean pooling and MLP architecture as the tactile model. We expect this to be the weakest unimodal baseline, since slip is a contact-level phenomenon largely invisible from an external viewpoint until the object has already moved significantly.

2.1.3 Force/Torque Only (LSTM)

The F/T sensor senses the forces and torques being applied to the end-effector of the arm. It produces a six-dimensional time-series signal at each timestep (three forces, three torques). We sample one (6 dim) reading per second, and featurize that into a 256-dim vector via a series of Linear, ReLU and dropout (p=0.1) layers. We then apply a 2-layer BiLSTM on this, with a dropout of 0.1. We then concatenate the first and last embeddings of the BiLSTM into a 512-dim vector, and use an MLP to classify whether the grip and pose results in a non-stable grasp that will slip during the shake phase.

We hypothesize that the Force-Torque values will give a clue into how tightly the object is held.

2.1.4 Robot Joint States Only (LSTM)

The joint states of the robot are a collection of 12 values per timestep. These correspond to the joint angles and velocities of all of the six degrees of freedom that the robot has. We use the exact same model as above.

This ablation is interesting because while training we only consider the items that have not been dropped in the grasp and pose phases. The prior here is that the object has not fallen during the grasp and pose phase. How might the model use this information? It is a reasonable bet then that if the joints have shaken a lot (this can be found out by looking at joint states or the aforementioned force-torque readings), that the grasp is strong, because we have it as a given that the object does not fall during these phases. So, this assumption means that there is a relationship between the movement in the grip and pose states and the strength of the grasp!

2.2 [0.5 points] Simple Multimodal Baselines

Our simple multimodal baselines fuse modalities via naive concatenation, without any fancy attention or transformer architectures. The goal is to test whether combining modalities improves any performance at all, and to characterize the extent to which naive fusion fails relative to more state-of-the-art models.

Before describing the individual multimodal baselines, we address a practical challenge that arises the moment we try to combine these modalities: different sensors operate at different frequencies. The tactile and RGB cameras capture frames at a fixed rate, while the F/T sensor samples at a significantly higher frequency. To resolve this, we fix a sampling parameter for each modality independently. We sample a fixed number of tactile/RGB frames and a fixed number of F/T readings within each timestep window. These sampled observations are then concatenated to form the input for that timestep. This keeps the input dimensionality fixed across all timesteps.

2.2.1 Tactile + RGB (LSTM)

A ResNet-50 encoder independently processes the tactile and RGB frames at each timestep, producing a feature vector for each. These are concatenated and passed into a 2-layer bidirectional LSTM, whose final hidden state is fed to an MLP classifier with a binary output. Our motivation is to test whether combining the two visual modalities and adding temporal context improves over either one alone.

2.2.2 Tactile + F/T (LSTM)

We use the same architecture as above, but using the force/torque readings instead of the RGB images. We sample F/T readings and tactile images, and concatenate them as mentioned earlier. This combination is interesting because it pairs the two modalities we most expect to be informative for slip.

2.2.3 Tactile + RGB + F/T (LSTM)

All three modalities are independently encoded, concatenated, and passed through a uni/bi-directional LSTM. This is our primary simple multimodal baseline and closely follows the approach from the original PoseIt paper. We expect this to be a reasonable but suboptimal baseline as it has access to all the right information but no mechanism to ensure each modality is used effectively during training.

2.2.4 Tactile + RGB + F/T + Actions (LSTM)

This extends the previous baseline by appending the robot's action vector, which includes the gripper position and arm joint velocities, to the concatenated feature vector at each timestep. During the grasp and pose stages, the robot moves quite significantly, so the action vector may carry meaningful context about the nature of the motion leading up to the stability phase. So the idea is to test whether knowing what the robot is doing provides additional context for predicting slip, which is also tested in works like Mandil et al. (2024) and Nazari et al. (2022).

2.3 [0.5 points] Competitive Baselines

Our main reference is the PoseIt paper (Kanitkar et al., 2022), which uses tactile, RGB, F/T and gripper force, and feeds them into a 2-layer bidirectional LSTM for binary slip prediction. Because slip detection on tactile and F/T data lacks standard benchmarks with pretrained models for our setup, we do not compare against external systems. Instead, we can think of architectural variants of our implementation as competitive baselines. Table 1 reports unidirectional vs. bidirectional LSTM, LSTM vs. GRU, and different LSTM depths, all trained under similar settings.

All ablations were trained with the following shared setup unless noted otherwise. We used SGD with momentum 0.9 and weight decay 0.01, initial learning rate 0.01, and a step decay by a factor of 10 at iteration 300. Training ran for 600 iterations with batch size 200, random train/val/test split, and a maximum sequence length of 9 seconds. We applied deferred resampling (DRS) with $\sigma = 0.5$ from iteration 400 to balance stable and unstable grasps, and used BCEWithLogitsLoss with pos_weight set from the training label distribution to handle class imbalance. Vision inputs (RGB and GelSight) were encoded by frozen ResNet50 backbones; temporal fusion used a 2-layer bidirectional LSTM (or GRU) with hidden dimension 500 and dropout 0.1. The 6-layer variants were trained for 1200 iterations with batch size 100 and annealing at iteration 600.

	Dev				
Methods	**Loss**$\downarrow$	**Accuracy(%)**$\uparrow$	**Precision**$\uparrow$	**Recall**$\uparrow$	**F1**$\uparrow$
Tactile Only (ResNet + MLP)	0.7395	34.04	0.357	0.789	0.492
RGB Only (ResNet + MLP)	0.6720	60.64	1.000	0.026	0.051
F/T Only (LSTM)	0.6855	48.87	0.376	0.862	0.524
Joint State Only (LSTM)	0.6900	55.6	0.451	0.667	0.538
Tactile + RGB (LSTM)	0.6582	78.09	0.690	0.817	0.748
Tactile + F/T (LSTM)	0.8423	61.24	0.474	0.569	0.517
Tactile + RGB + F/T (LSTM)	0.6934	37.89	0.359	1.000	0.528
Tactile + RGB + F/T + GF (LSTM)	0.6727	53.42	0.432	0.800	0.561
Tactile + RGB + F/T (GRU)	0.823	56.18	0.45	0.892	0.598
Tactile + RGB + F/T (Unidirectional LSTM)	0.8546	39.89	0.382	0.970	0.549
Tactile + RGB + F/T (6 layer LSTM)	0.6941	39.13	0.391	1.000	0.562
Tactile + RGB + F/T (6 layer GRU)	0.833	35.80	0.353	0.984	0.519

Table 1: All ablations were trained with the following shared setup unless noted otherwise. We used SGD with momentum 0.9 and weight decay 0.01, initial learning rate 0.01, and a step decay by a factor of 10 at iteration 300. Training ran for 600 iterations with batch size 200, random train/val/test split, and a maximum sequence length of 9 seconds. We applied deferred resampling (DRS) with $\sigma = 0.5$ from iteration 400 to balance stable and unstable grasps, and used BCEWithLogitsLoss with pos_weight set from the training label distribution to handle class imbalance. Vision inputs (RGB and GelSight) were encoded by frozen ResNet50 backbones; temporal fusion used a 2-layer bidirectional LSTM (or GRU) with hidden dimension 500 and dropout 0.1. The 6-layer variants were trained for 1200 iterations with batch size 100 and annealing at iteration 600.

[1 points] for formatted table. [1 points] Results for for N*2 approaches (N is number of teammates)

3 Results (1 page)

We evaluate all models on the PoseIt dataset using three complementary metrics. Together, they tell us how the model handles different types of mistakes that matter in robotic grasping.

[1 points] For three metrics

Accuracy measures the percentage of stability windows correctly classified as slip or no-slip. It provides a quick, high-level summary of performance. However, accuracy alone can be misleading when the dataset is imbalanced. In our case, if no-slip windows are much more common, a model that always predicts "no-slip" could achieve high accuracy while being practically useless for real deployment.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision measures how often the model is correct when it predicts a slip. This matters in real-world settings where false alarms are costly. If the robot frequently believes a slip is happening when it isn't, it will trigger unnecessary corrective actions. These interruptions reduce efficiency and may even destabilize an otherwise secure grasp. High precision means the robot only reacts when it truly needs to.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall measures how many real slip events the model successfully detects, making it especially important for safety in manipulation tasks. If a slip goes unnoticed, the robot may drop the object or fail the task entirely. Because the cost of missing a slip is typically much higher than triggering an unnecessary correction, we prioritize recall when comparing models, particularly in cases where overall accuracy alone does not fully reflect the impact of different types of errors.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score is the harmonic mean of precision and recall. It captures the trade-off between the two and is particularly useful in imbalanced settings like ours, where accuracy alone can be misleading. A higher F1 score is better, with 1 indicating perfect precision and recall.

$$\text{F1 Score} = \frac{2TP}{2TP + FP + FN}$$

4 Model Proposal (1 page)

4.1 [1.5 points] Overall model structure

We propose two transformer-based architectures for multimodal slip detection, both of which move beyond the naive LSTM concatenation baselines.

1. Vanilla Multimodal Transformer - We treat all modalities jointly in a single shared attention space and run self-attention through all

2. Multimodal Bottleneck Transformer - We separate within-mmodality self-attention from the cross-modal fusion through an attention bottleneck mechanism (Nagrani et al., 2022)

Both architectures share the same binary classification objective at the top.

4.2 [1 points] Encoders

All three modalities are first processed by dedicated encoders that convert raw sensor observations into fixed-dimensional token vectors before being passed into any transformer.

For tactile and RGB, we sample F1 frames per timestep and stack them along the channel dimension, producing an input tensor of shape `(3*F1,H,W)`. This is passed through a 2D CNN encoder that processes all F1 frames jointly and outputs a single feature vector per timestep. We stack along the channel dimension rather than processing frames independently to try and address the quadratic problem of transformers: increasing the length of the input tokens leads to a quadratic increase in the computation cost, which is not desirable. Since our modalities produces multiple samples within a timestamp due to their high sensor-frequency, we made the above deliberate choice, thereby allowing the CNN to capture motion and deformation patterns across the F1 frames rather than treating each frame in isolation. An alternative here would be to use a pretrained ViT backbone that processes each frame as a sequence of patch tokens, which would give richer spatial representations but at significantly higher computational cost. We use a CNN for both modalities in the interest of keeping the architecture lightweight and trainable from scratch on the PoseIt dataset, which is not large enough to fine-tune a large pretrained vision transformer reliably (Dosovitskiy et al. (2021)).

For force/torque, we sample F2 readings per timestep, each being a 6-dimensional vector (3 forces, 3 torques), giving an input of shape `(F2, 6)`. This is passed through a 1D CNN encoder that processes the F2 readings as a short time-series and outputs a single feature vector per timestep. An alternative would be to simply flatten the F2 readings and pass them through an MLP, which is what our LSTM baselines do, but this ignores the temporal ordering of the readings within a timestep.

4.2.1 Architecture 1: Vanilla Multimodal Transformer

In the vanilla transformer, all three modalities are encoded into a single unified token sequence. At each timestep, the tactile encoder, RGB encoder, and F/T encoder each produce one token, giving three tokens per timestep. These are arranged into a flat sequence of $3 * T_k$ tokens, with a [CLS] token appended at the end. Before entering the transformer, each token receives two additional embeddings: a temporal positional embedding that encodes which timestep it came from, and a learned modality embedding that identifies which sensor it came from. Without the modality embedding, the transformer has no way to distinguish a tactile token from an RGB token at the same timestep since they have identical positional encodings.

This full sequence is then passed through L stacked transformer encoder blocks, each consisting of multi-head self-attention followed by a feed-forward network with residual connections and layer normalization. Through self-attention, every token can attend to every other token regardless of modality or timestep. The [CLS] token's final representation is passed through a linear classification head to produce the slip prediction.

Figure 1 shows the model architecture.

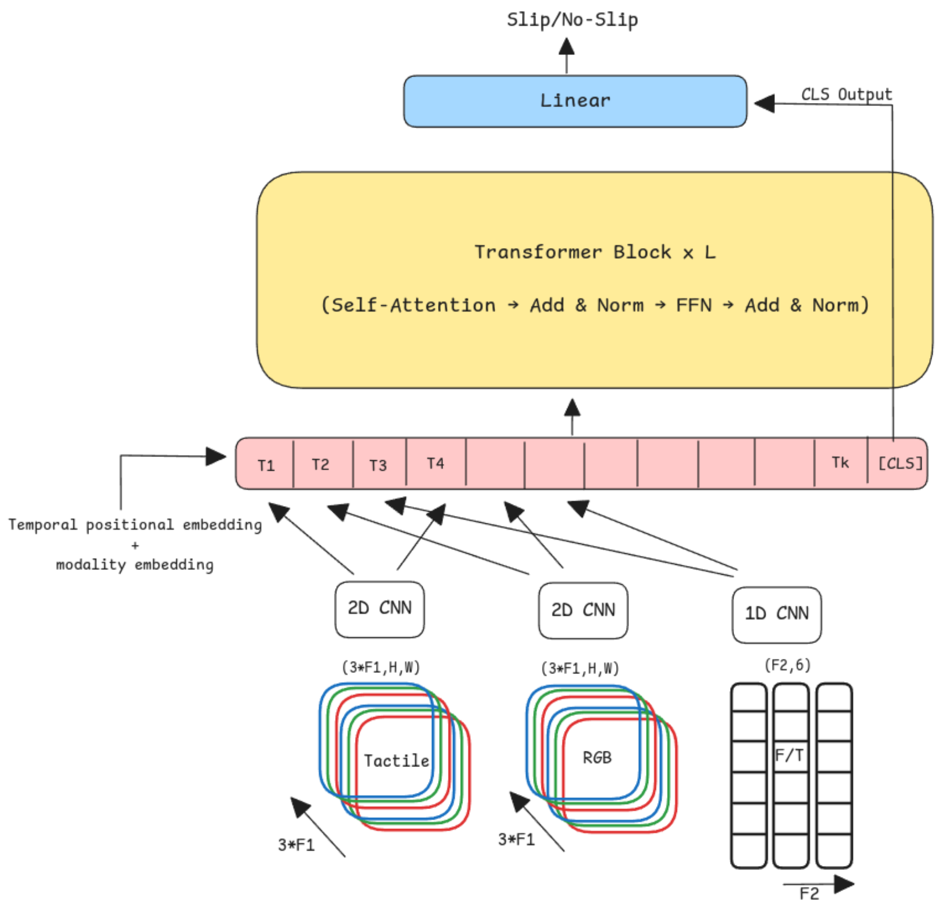

Figure 1: Vanilla Transformer Architecture

4.2.2 Architecture 2: Multimodal Bottleneck Transformer

The bottleneck transformer addresses a fundamental limitation of the vanilla approach: when all modalities compete in a shared attention space, dominant modalities can crowd out weaker ones. Our second architecture separates within-modality self-attention from cross-modal fusion, giving each modality dedicated capacity to develop its own temporal representations before any cross-modal interaction occurs.

Each modality has its own independent transformer encoder (Transformer x Lf) that runs self-attention exclusively within that modality's token sequence. So the tactile transformer sees only tactile tokens across all T_k timesteps, the RGB transformer sees only RGB tokens, and the F/T transformer sees only F/T tokens. Each sequence also has its own [CLS] token that aggregates a global summary of that modality's temporal information. The main purpose of the [CLS] tokens here is to compute auxiliary losses. These within-modality transformers run in parallel with no communication between them.

After within-modality encoding, cross-modal fusion happens through a small set of B learnable bottleneck tokens `(FSN1, FSN2, ..., FSNB)`. These tokens have no modality-specific content of their own. Rather, they are randomly initialized and learned entirely through training. Cross-modal attention is then performed: the bottleneck tokens (which provide the Query vectors) attend to all three modalities (which all provide Key and Value vectors). This cross-attention block is repeated Lb times. Rather than allowing all-to-all attention between every token across all modalities like we did in the previous architecture (which would be expensive due to the quadratic problem of transformers and potentially noisy), information must flow through this small bottleneck, forcing

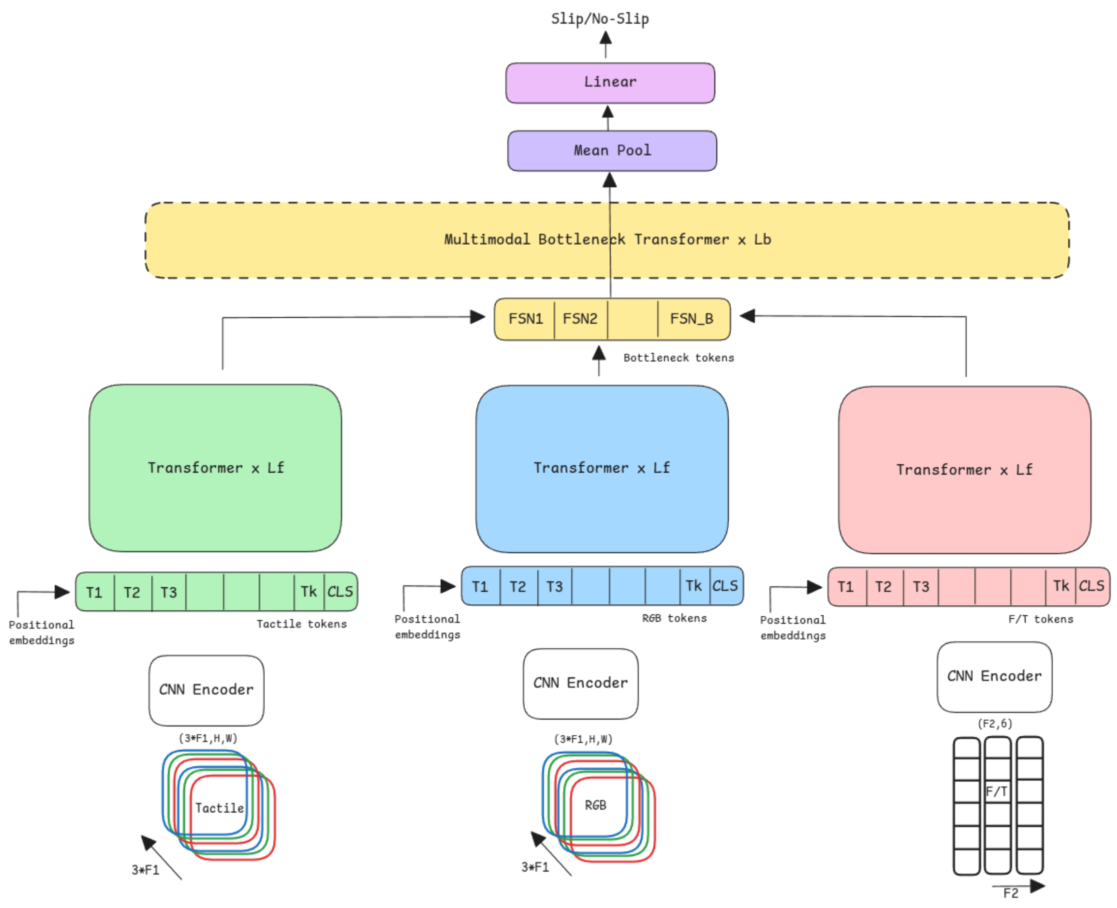

Figure 2: Attention Bottlenecks Transformer Architecture

the model to be selective about what cross-modal information it captures. The number of bottleneck tokens B is a hyperparameter that we will play around with.

For classification, we mean-pool across the B bottleneck tokens at the final layer and pass the resulting vector through a linear head to produce the slip prediction. We choose to do classification with the bottleneck tokens here since they are the only components that have attended to all three modalities, and hence carry genuine cross-modal information.

Figure 2 shows the model architecture.

4.3 [1 POINTS] DECODERS

Since our task is binary classification rather than sequence generation, we do not use a traditional decoder. Instead, the decoder in our architecture refers to the classification head that maps the final learned representation to a slip or no-slip binary prediction.

4.4 [1 POINTS] LOSS FUNCTIONS

Both architectures are trained with binary cross-entropy loss on the final classification output.

For the bottleneck transformer specifically, we additionally supervise each modality's [CLS] token with its own auxiliary binary cross-entropy loss. Each CLS token is passed through a small linear head and trained to predict slip independently from its modality alone. The total loss is then:

$$L_{\text{total}} = L_{\text{main}} + \lambda \left(L_{\text{aux\_tactile}} + L_{\text{aux\_rgb}} + L_{\text{aux\_ft}}\right)$$

where L_{main} is the loss from the bottleneck classification head and λ is a weighting hyperparameter.

The motivation for these auxiliary losses is directly tied to the modality imbalance problem. Without them, the within-modality encoders only receive gradient signal through the bottleneck, which may be dominated by the most informative modality. The auxiliary losses ensure every encoder receives a direct, task-relevant gradient regardless of what happens at the fusion stage, forcing each modality to develop meaningful independent representations.

References

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL `https://arxiv.org/abs/2010.11929`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing, 2024. URL `https://arxiv.org/abs/2410.24090`.

Shubham Kanitkar, Helen Jiang, and Wenzhen Yuan. Poseit: A visual-tactile dataset of holding poses for grasp stability analysis, 2022. URL `https://arxiv.org/abs/2209.05022`.

Jianhua Li, Siyuan Dong, and Edward Adelson. Slip detection with combined tactile and visual information, 2018. URL `https://arxiv.org/abs/1802.10153`.

Willow Mandil, Kiyanoush Nazari, and Amir Ghalamzan E. Action conditioned tactile prediction: case study on slip prediction, 2024. URL `https://arxiv.org/abs/2205.09430`.

Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion, 2022. URL `https://arxiv.org/abs/2107.00135`.

Kiyanoush Nazari, Willow Mandil, and Amir Ghalamzan E. Proactive slip control by learned slip model and trajectory adaptation, 2022. URL `https://arxiv.org/abs/2209.06019`.