Prediction and Control of Slip Velocity in Robotic Grippers via Transformer-Based Fusion of Visuotactile, Image, and Vibration Signals

Master thesis by Noah Becker
Date of submission: August 31, 2026

1. Review: Jan Peters
2. Review: Tim Schneider
3. Review: Kay Pompetzki
Darmstadt



TECHNISCHE UNIVERSITÄT DARMSTADT

Intelligent Autonomous Systems

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 APB TU Darmstadt

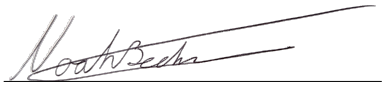
Hiermit erkläre ich, Noah Becker, dass ich die vorliegende Arbeit gemäß § 22 Abs. 7 APB der TU Darmstadt selbstständig, ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt habe. Ich habe mit Ausnahme der zitierten Literatur und anderer in der Arbeit genannter Quellen keine fremden Hilfsmittel benutzt. Die von mir bei der Anfertigung dieser wissenschaftlichen Arbeit wörtlich oder inhaltlich benutzte Literatur und alle anderen Quellen habe ich im Text deutlich gekennzeichnet und gesondert aufgeführt. Dies gilt auch für Quellen oder Hilfsmittel aus dem Internet.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§ 38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, August 31, 2026



N. Becker

Abstract

Manipulating an object by letting it slip through our fingers is as intuitive as it is easy for humans. Yet the forces and dynamics behind slipping, and the control required to exploit it are, even after 300 years of research into the subject, not fully understood. Learning-based approaches have become more popular, though they often content themselves with merely detecting binary slip. This thesis goes further, asking whether slip velocity can be estimated continuously by fusing visuotactile images with high-frequency vibration signals, which capture the slip-relevant mechanical cues that vision-based tactile sensors alone cannot reach. A purpose-built test bench subjects a GelSight Mini sensor and an ADXL355 accelerometer to controlled slip across different surface textures, and records an independent optical ground truth of slip velocity. On this dataset, eight neural network architectures are compared. A hybrid design combining per-sensor late fusion with a transformer trunk performs best, reaching a mean absolute error of $2.13\,\mathrm{mm/s}$ during slip with a latency of $8.7\,\mathrm{ms}$, establishing that slip velocity is continuously inferable at rates compatible with reactive grasp control.

Acknowledgements

I could not have completed this thesis alone. Alone I would not have had the foresight to find this topic, the expertise to understand it, or the support to pull it through.

I thank my phd supervisors Kay Pompetzki and Tim Schneider for their support and motivation and my proffessor Jan Peters for for sparking my intrest in robotics.

Kyrylo Sovailo, Chunyao Zhu, and Eric Gattung, who worked with me on earlier visuotactile research.

Paul List, Thomas Ellenberger, Timo Kolbe, Faris Begic, and Florian Mack, who helped build the test bench.

Helen Reger for the advice on the physical processes involved in this work.

And I thank my family and friends for their support throughout this project.

Contents

1 Introduction **8**
- 1.1 Research questions . . . 10
- 1.2 Contributions . . . 11
- 1.3 Thesis structure . . . 12

2 Background **14**
- 2.1 Tribology . . . 14
  - 2.1.1 Friction . . . 14
  - 2.1.2 Incipient and Gross Slip . . . 16
  - 2.1.3 Stick-Slip . . . 18
  - 2.1.4 The relationship between velocity and slip vibration . . . 19
- 2.2 Sensors . . . 21
  - 2.2.1 GelSight . . . 21
  - 2.2.2 Accelerometer . . . 23

3 Related Work **25**
- 3.1 Slip Frequency . . . 25
- 3.2 Vibration Sensor Alternatives . . . 26
  - 3.2.1 Vision-based tactile sensors . . . 26
  - 3.2.2 Piezoelectric sensors . . . 26
  - 3.2.3 Laser Doppler Velocimeter . . . 28
  - 3.2.4 Barometric Sensors . . . 29
- 3.3 Learning Based Slip Detection . . . 31
- 3.4 Data Fusion . . . 32
  - 3.4.1 Fusion Structure . . . 32
  - 3.4.2 Transformer Based architectures . . . 34

4 Experimental Platform **36**
- 4.1 Test Object . . . 36

4.2 Frame and Mounting . . . 38
4.3 Actuators . . . 39
4.3.1 Gripper . . . 39
4.3.2 Linear Motor . . . 39
4.4 Sensors . . . 40
4.4.1 Vibration Sensors . . . 40
4.4.2 GelSight . . . 42
4.4.3 Webcam . . . 43
4.4.4 Force Torque Sensor . . . 44
4.5 Synchronisation . . . 44
4.6 Wiring and Connections . . . 45
4.7 Data Acquisition . . . 48

5 Methodology **50**
5.1 Task formulation . . . 50
5.2 Data Representation . . . 52
5.2.1 Tactile Representations . . . 52
5.2.2 Vibration Representations . . . 54
5.3 Network Preliminaries . . . 56
5.3.1 Architecture Overview . . . 56
5.3.2 Time Alignment . . . 57
5.3.3 Time encoding . . . 57
5.3.4 Double Head architecture . . . 58
5.3.5 Branch Encoders . . . 60
5.4 Base Network Architectures . . . 64
5.4.1 Late-Fusion Network (`Fusion`) . . . 65
5.4.2 Continuous-Time Transformer (`Transformer`) . . . 66
5.4.3 Hybrid Representation-Fusion Network (`Hybrid`) . . . 70
5.5 Themed Architectures . . . 72
5.5.1 Rate-Decoupled Core Network(`RateCore`) . . . 72
5.5.2 Stripped-Down Single-Stream Network (`Strip`) . . . 75
5.5.3 Cross-Modal Self-Supervised Pretraining (`Twin`) . . . 78
5.5.4 Parametric Decomposition of Slip Breakaway Event (`Event`) . . . 80
5.5.5 Evidence-Weighted Fusion Network (`Evidence`) . . . 84

6 Results **86**
6.0.1 Baseline . . . 86
6.1 General Results . . . 86

6.2 Error Sources . . . 90
6.3 Unseen Surface Test . . . 93
6.4 Sensor Ablation . . . 94
6.5 Depth Input . . . 95
6.6 Parameter Size . . . 97
6.7 Latency . . . 99
6.8 Label Efficiency (Twin) . . . 101
6.9 Slip Velocity Spike . . . 102
6.10 Future Velocity Prediction . . . 106
6.11 Limitations . . . 107

7 Conclusion **110**
7.1 Core Learnings . . . 110
7.2 Discussion . . . 111
7.3 Outlook . . . 112

Bibliography **113**

"If we continue to study [friction] more and more, measuring more and more accurately, the law will continue to become more complicated, not less. In other words, as we study this law of [friction] more and more closely, we find out that it is 'falser' and 'falser', and the more deeply we study it, and the more accurately we measure, the more complicated the truth becomes; so in that sense we consider it not to result from a simple, fundamental process."
Richard Feynman [24]

1 Introduction

Tribology is the study that deals with the design, friction, wear, and lubrication of interacting surfaces in relative motion [54]. Classically it's a subfield of mechanical engineering, but with precise robot manipulation having become a greater research focus, tribology has become a more interdisciplinary field.

Humans have surprisingly efficient touch mechanisms, with our fingertips having about 240 mechanoreceptors per square cm [51]. That is approximately 39 PPI, with the total "brain bandwidth" used for processing touch being 100 bits/s [65]. Compare this to vision, where peak photoreceptor density in the central fovea exceeds $160,000$ units/mm^2 (roughly 16×10^6 units/cm^2 $=$ 10,000 PPI) [16], the visual system demands a vastly higher data throughput. The optic nerve transmits sensory information to the visual cortex at an estimated rate of 4.3×10^6 to 10×10^6 bits/s [46, 60].

Yet despite this discrepancy in bandwidth, humans are very capable of physical interaction without visual input. A study by Memeo et al. [78] showed that participants can assemble complex objects with 90% accuracy without visual input. Jenmalm & Johansson had blindfolded humans pick up topologically complex objects, and measured the time until a stable grasp was reached, and how large the "safety margin", i.e. the difference between the required grip force and the actual grip force, was. They found that a stable grasp was reached within 100 ms, with a safety margin of 40%. This is considerable in view of the fact that the round-trip time of a signal from the hand and back is approximately 70 ms [50], meaning that the stable grasp is computed in less than 30 ms.

Assuming that this human skill is transferable to digital systems, tactile sensing could provide an efficient manipulation modality, either complementing or in some cases replacing vision-based systems. Tactile-based systems are also invariant to lighting, occlusion, optical surface properties, and background clutter. Beyond this robustness, tactile sensing directly measures quantities inaccessible to vision: contact forces, pressure distribution, and shear, which vision can only infer indirectly from an object's appearance and motion.

Realizing tactile sensing, however, runs into practical constraints: high-resolution tactile arrays remain expensive and lack the ubiquity of cameras, whose signal format has the benefit of decades of learning-based processing research behind it. Visuotactile sensing solves this by using a camera as the transduction element for the tactile signal, rather than substituting vision for touch. The main downside of external camera-based systems are the large amounts of data that must be processed. In turn the upside of visuotactile sensing is that the input is limited to the contact area. As the camera is much closer to the point of interest, a much lower resolution can be used to get the same or better fidelity.

Manipulation as a whole is a complex problem, spanning grasp planning, object reorientation, in-hand regrasping, and dexterous control under uncertainty, each an open research field in its own right. Rather than address this breadth, this thesis isolates a single sub-problem that underlies all of it: slip. Slip is the signal that decides whether a grasp holds or fails. It is also narrow enough to define, measure, and evaluate rigorously, while remaining representative of the real-time demands that make tactile manipulation hard. Crucially, slip is a dynamic phenomenon: it must be caught as it happens, inside the control loop. Visuotactile sensing falls short here, it does not have the modality to sense one of the most important signals in slip: vibration.

A clear gap in current vision-based tactile manipulation is frequency based inference. Johansson and Westling [50] have shown: humans rely on high frequency mechanical perturbations to detect slip [120]; disrupting tactile feedback disrupts the normal, fine-tuned grasp control and results in frequent object dropping despite excessive compensatory grip forces [52, 5]. Humans are able to discern frequencies up to $1000\,\text{Hz}$, with the slip relevant frequencies theorized to be around $300\,\text{Hz}$ [120]. That is an order of magnitude greater than most commercial vision systems. Even high end cameras that can reach those polling rates produce amounts of data that are prohibitively large for inference. For example, a 1080p monochrome video at $1000\,\text{Hz}$ equates to approximately $2\,\text{GB/s}$; too much for the live, low-latency processing loop that slip-reactive control demands.

The solution proposed here is to augment visuotactile sensing with vibration sensing, fusing the two modalities into a single perception pipeline (Figure 1.1). This leads to

the central research question of this thesis: *can slip velocity be estimated from visuotactile images combined with high-frequency vibration signals?*

Vibration sensing is the natural complement to the visuotactile sensor: it captures precisely the high-frequency mechanical cues that humans exploit to detect slip, yet at a fraction of the data rate of high-speed video, keeping the perception pipeline within the latency budget that reactive grasp control demands.

Answering this question calls for data that existing benchmarks do not provide. Most prior work treats slip as a binary event [41, 77, 103, 1, 69, 119, 20, 21, 39] whereas closing a grasp-control loop requires a continuous estimate of how fast contact is being lost. To this end, a dedicated test bench is used to subject visuotactile and vibration sensors to controlled slip while simultaneously recording a precise, independent ground truth of slip velocity, providing the data on which the estimation methods of this thesis are developed and evaluated.

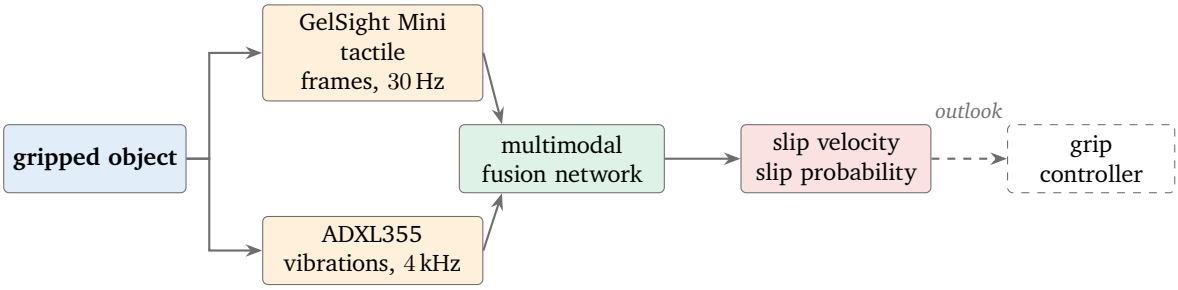


Figure 1.1: Overview of the proposed slip velocity estimation pipeline, showing the two sensors used in this thesis: a GelSight visuotactile sensor and an ADXL355 accelerometer for detecting slip vibrations. Their data is fused using methods explored in Chapter 5 and used to estimate the slip velocity.

1.1 Research questions

The central research question (RQ) of this thesis is: *can slip velocity be estimated continuously from visuotactile images combined with high-frequency vibration signals?*. This question decomposes into four research questions, around which the thesis is organized:

RQ1 What modalities are useful and necessary for estimating slip velocity?

RQ2 What representations of those modalities are useful for estimating slip velocity?

RQ3 What neural network architecture and size is best for determining slip velocity?

RQ4 How does one construct a setup to gather the data necessary to train slip velocity predicting models?

1.2 Contributions

In answering these questions, the thesis makes the following contributions:

1. **A slip velocity dataset** recorded across three surface textures (Section 4.1) under controlled, repeatable pulling conditions, with an independent optical ground truth of slip velocity (Section 5.1). The acquisition procedure and the resulting corpus are described in Section 4.7. *(RQ4)*
2. **A reproducible, fully documented sensing platform** (Chapter 4). Starting from a test bench developed in prior work [70], this thesis contributes the vibration sensing chain (Section 4.4.1), the PTP-based synchronisation of all streams (Section 4.5), and the calibration and data-acquisition procedures (Section 4.7), and documents the failure modes and pitfalls encountered along the way. *(RQ4)*
3. **A study of sensor modalities and their representations** that defines and implements a set of data abstractions (Section 5.2) and quantifies which modalities are necessary and sufficient for slip velocity estimation (Sections 6.4 and 6.5), via a branching multi-modal network with modality dropout and a deployment head restricted to tactile and vibration sensor inputs (Sections 5.1 and 5.3.4). *(RQ1, RQ2)*
4. **Eight neural network architectures** (Sections 5.4 and 5.5), compared in Chapter 6, spanning:
    - Per-sensor branch encoders fused by concatenation into an MLP trunk (Section 5.4.1).
    - A continuous-time tokenizer feeding every stream into a transformer at its native rate (Section 5.4.2).
    - A discrete-time variant that pre-summarizes each stream into a fixed token layout before the transformer (Section 5.4.3).

- A split architecture: a fast vibration-based stack running at the accelerometer rate, refreshed by a slower tactile context stack whenever an image arrives (Section 5.5.1).
- A single-tensor stack that reduces the tactile field to a handful of scalars and uses no image encoder at all (Section 5.5.2).
- A resilience-based stack that down-weights sensor branches whose data is corrupted but still present (Section 5.5.5).
- A stack that models the slip spike as a Gaussian transient on top of a plateau (Section 5.5.4).
- A pretraining-based architecture that uses cross-modal self-supervision to reduce the labelled data needed in a new setup (Section 5.5.3, evaluated in Section 6.8).

5. **A multimodal fusion transformer** (Section 5.4.2) that tokenizes tactile features, vibration sensor spectrograms, and optional side modalities with modality and relative-time embeddings (Section 6.3). *(RQ3)*
6. **A comparative evaluation of slip-velocity estimation architectures** showing that the hybrid architecture performed the best
7. **A path toward slip velocity control** that establishes a latency budget for the full pipeline and evaluates the different architectures for use in slip velocity control in Section 7.3. *(RQ3)*

1.3 Thesis structure

In order to explain the whole research, implementation and evaluation process involved in determining slip-velocity, the thesis is structured as follows: Chapter 2 introduces the physics of friction and slip and the sensing modalities available to perceive it. Chapter 3 surveys analytical, learning-based, and multimodal approaches to slip perception and identifies the research gap. Chapter 4 describes the experimental platform, the data-acquisition pipeline, and the dataset. Chapter 5 formulates the learning problem over the tactile and vibration sensor input pair, defines the data representations, and develops the network architectures. Chapter 6 evaluates these models on held-out data, on an unseen surface, and under sensor ablation, and extends the evaluation to the slip spike, measured

inference latency, and the network-size sweep that establishes how small the model can be made. Chapter 7 discusses the findings and concludes.

2 Background

This chapter assembles the physical and technical foundations of the thesis: the mechanics of friction and slip (Section 2.1), slip velocity as a control variable (Section 2.1.4), and the sensor modalities available for perceiving slip (Section 2.2).

2.1 Tribology

This section covers the theory behind friction and slip. As mentioned in the introduction, both are vastly complex physical phenomena, and the following will try to illustrate why that is, and what assumptions are made about the phenomena.

2.1.1 Friction

In order to understand slip and how to detect it, a basic understanding of the underlying friction principles is necessary. Despite friction being widely used in engineering, there is no unified theory for how it works. This thesis uses friction models presented by Persson et al. [89], Krylov et al. [66] and the contact area based friction model by Bowden and Tabor [10].

According to Krylov et al. [66], the friction forces measured on macroscopic bodies are essentially the electromagnetic forces between the atoms coming into contact. A precise modeling of friction would therefore involve modeling the couplings between electrons and nuclei on a microscopic level using quantum electrodynamics equations. This, according to the researchers, “is a hopelessly complicated problem”. The model of friction and slip this thesis present is therefore a simplification of the underlying phenomena.

The main difficulty with friction becomes apparent when looking at the microscopic surface structure of two touching materials. At this scale, virtually all materials have

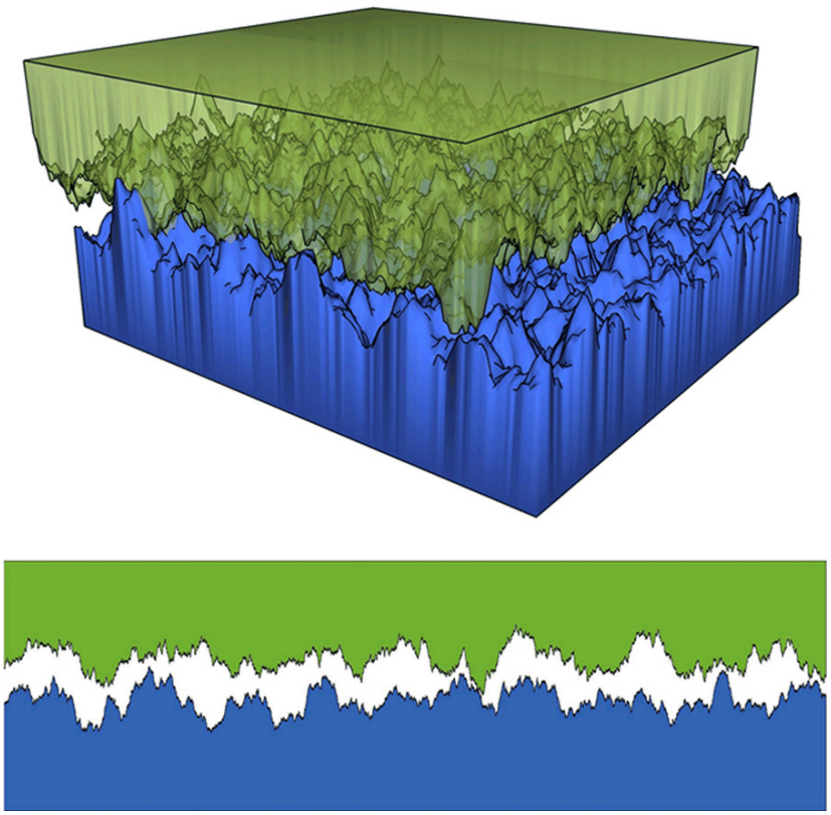


Figure 2.1: Fractal surface contact visualization showing multiscale asperity interaction between two surfaces. [32]

jagged fractal surfaces [73]. Two surfaces in contact only touch at their asperities, the microscopic peaks of the rough surface topology, as visualized in Figure 2.1. The real contact area is therefore much smaller than the apparent (geometric) contact area. As normal force increases, asperities deform and the real contact area grows, leading to increased frictional resistance [10]. Any movement between the two objects is constantly changing the contact surface in a highly non-linear fashion. In addition, the surfaces can interlock, a phenomenon described as *stick-slip* and treated in detail in Section 2.1.3.

For most pragmatic engineering purposes, this highly complex landscape is reduced to Coulomb's friction model, described by the function $F = \mu N$, where the friction force F is put in relation to the friction coefficient μ, and the normal force N. μ is material dependent and determined empirically. In addition, a distinction is generally made between static

and kinetic friction. Static friction is the tangential force F_{static} that acts between two touching objects up to a maximal static friction force.

Kinetic friction acts when the objects move relative to each other. It is approximately constant and proportional to the normal force. Generally, the kinetic friction coefficient is smaller than the static friction coefficient.

The Coulomb model thus treats the transition from sticking to sliding as instantaneous and uniform across the contact: the moment the tangential load exceeds the static limit, the entire interface begins to slide at once. The next section shows that in reality this transition is gradual and spatially structured.

2.1.2 Incipient and Gross Slip

Slip is generally understood as the relative motion of two touching surfaces. In the literature, this motion is commonly divided into two phases. During *gross slip*, the entire contact patch moves relative to the counter-surface: the object is macroscopically sliding. *Incipient slip* denotes the phase before that, in which parts of the contact area are already slipping while the rest remains stuck.

That such a phase must exist follows from the elasticity of the contact. When an elastic contact is loaded tangentially, the shear stress is not distributed uniformly but concentrates at the periphery of the contact patch. As the tangential load grows, the local friction limit is first exceeded at the edges, and a ring of incipient slip forms around a central stuck zone. This ring grows inward until the stuck zone vanishes, and gross slip begins, a process described analytically by the Cattaneo-Mindlin model [79] and studied in detail for various contact geometries [87]. Figure 2.2 shows this progressive detachment for a flat contact patch.

Going from incipient slip to gross slip is usually associated with a burst in slip velocity, as the potential energy that has been stored in the elasticity of the two objects gets released as the stuck zone disappears. This phenomenon is called *rupture* and is described, among others, by Ben-David et al. [9]. Ruptures can not only occur during slip onset. Even during gross slip sudden changes in contact area, for instance when through statistical processes two surfaces interlock sufficiently, rupture and as such velocity spikes can also occur spontaneously [8].

For slip detection, this two-phase picture is a very useful model. Since incipient slip precedes gross slip, a sensor that resolves the partial slip of the contact area can detect

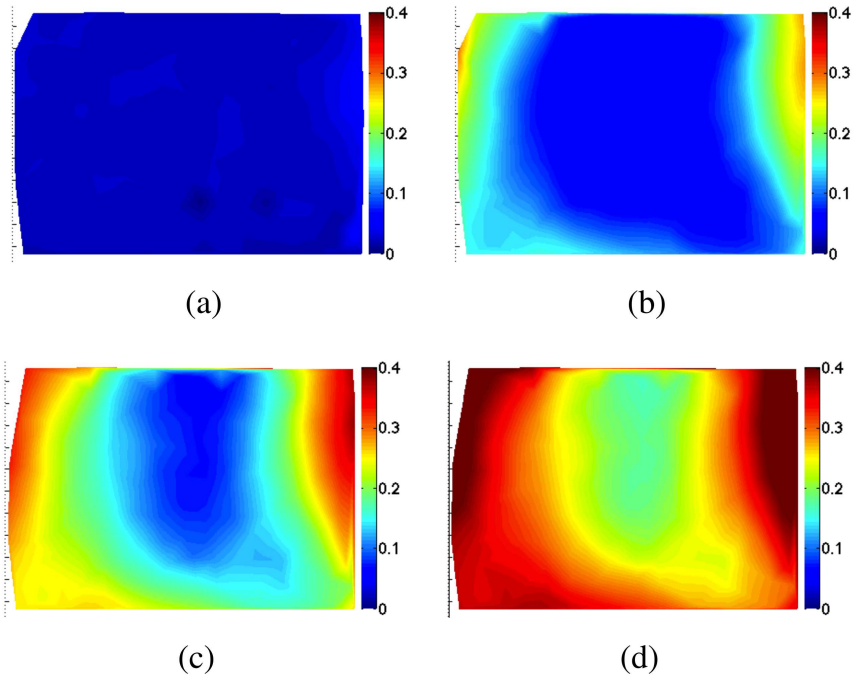


Figure 2.2: Shear displacement heatmaps showing progressive detachment of a contact patch under increasing tangential load. The color scale gives the difference between the applied shear displacement and the measured displacement of the elastomer surface in mm: dark blue marks sticking regions, red marks regions of local slip. Slip nucleates at the edges and corners, where shear stress concentrates, and propagates inward until the whole patch slides. Adapted from [119].

slip *before* the object as a whole starts to move, leaving a grip controller time to react [109, 21].

However, the clean division into a stuck zone and a slipping surface assumes a smooth continuum contact. As discussed in the previous section, real surfaces touch only at a sparse population of microscopic asperity junctions. Under tangential load, these junctions reach their local failure thresholds at different times, so at essentially any load some junctions are slipping and reforming while others hold. A "fully stuck" interface is therefore a statistical statement about an ensemble of micro-contacts rather than a true state of rest;

slow creep is always present, and junctions strengthen with the age of the contact [19, 98].

The converse is also true: there is no perfectly uniform gross slip. Experiments by Rubinstein, Cohen and Fineberg, who imaged the real contact area between two blocks during the onset of sliding, showed that the transition to gross slip is not a simultaneous event across the contact surface. Instead, rupture fronts occur locally and propagate across the contact at different speeds, and precursor fronts stop and relock partway across the contact surface at loads well below the nominal static friction threshold [96, 97]. The onset of sliding thus resembles the propagation of a crack more than a friction coefficient being exceeded everywhere at once. Notably, this behavior does not disappear for stiff materials. Partial slip requires elasticity, and since no material is truly rigid, even steel or glass contacts exhibit it; higher stiffness merely shrinks the length scales and speeds up the dynamics.

In summary, the binary distinction between stick and slip is useful at the macroscopic scale, but zoomed in, a frictional interface is always a patchwork of stuck, creeping, and slipping regions. This heterogeneity matters for this thesis in two ways: it means that any binary slip label assigned to sensor data is necessarily a threshold on a continuous process, and it is the mechanism behind the stick-slip oscillations described next.

2.1.3 Stick-Slip

The stick-slip phenomenon is a mechanical behavior observed in frictional systems, where an object subjected to an external force alternates between static ("stick") and kinetic ("slip") phases. This behavior arises from the fact that the kinetic friction is weaker than static friction. A visualization of this process can be seen in Figure 2.3:

1. Two surfaces, here a black object and a blue substrate, are pressed against each other, and the objects are sticking together. Because the two surfaces are interlocking, the real area of contact is large. A constant velocity v_k is adding energy to the system. This energy is stored in the elasticity of the black material, modeled as a spring with stiffness k_k. The force exerted by the spring can be modeled as $F_{spring} = k_k v_k t$ and is linearly growing with time t.
2. Once $F_{spring} > F_s$, the object starts to slip. Now the force resisting the spring force is the kinetic friction F_k, and $F_{total} = F_{spring} - F_k$. Since F_k is smaller than F_s, F_{total} momentarily increases. The object slides "over" the grooves of the substrate, as at the microscopic scale, the real surface area of the two surfaces is reduced.

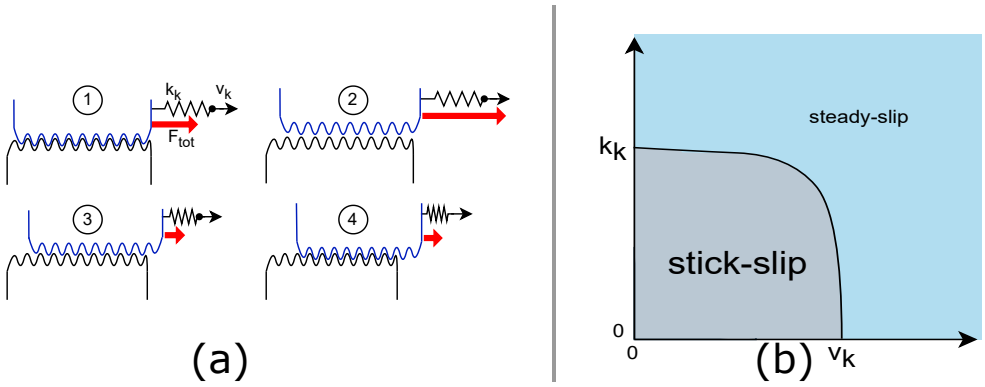


Figure 2.3: **(a)** Model of the stick-slip phenomenon. Surface of the substrate (blue) and the object (black) (1) Surfaces stick; energy stored in elasticity of object (2) Slipping starts; kinetic friction takes over (3) Stored energy released; motion slows (4) Low velocity; surfaces re-lock **(b)** General form of a kinetic friction phase diagram, v_k is velocity and k_k is stiffness of the material. Stick-slip occurs in the gray area, steady-slip occurs in the blue area.

3. The object slips, and the energy stored in the spring is used to move the object. As the spring force decreases, so does the velocity of the object.
4. At some point, the velocity of the object becomes small enough that it ceases to slide over the grooves, becoming interlocked again. At this point, static friction dominates again.

As long as v_k and k_k are below a threshold, the object will alternate between sticking to the substrate and slipping. This relationship is visualized in Figure 2.3. As v_k is increased, the time between sticking and slipping decreases. This causes the objects to vibrate, a phenomenon that is explained in the next section.

2.1.4 The relationship between velocity and slip vibration

Now that it has been determined that slipping is always associated with vibrations, the remaining question is what effect slip velocity has on those vibrations. On a violin for example, pulling a bow across a string causes the friction between the two to induce energy into the string, and because of the stick-slip phenomenon the string then starts to

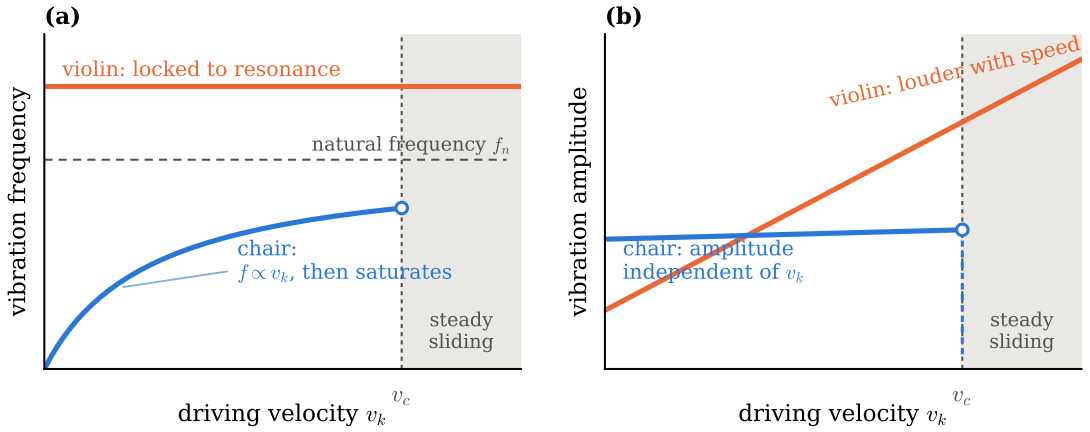


Figure 2.4: The two regimes of friction-induced vibration, sketched conceptually. **(a)** Vibration frequency as a function of driving velocity v_k. In the relaxation-oscillation regime (chair, blue) the frequency grows linearly with v_k, saturates near the natural frequency f_n of the elastically coupled system, and vanishes into steady sliding beyond a critical velocity v_c [82]. In the self-excited, resonance-locked regime (violin, orange) the frequency is fixed by the resonator, independent of the driving velocity [114]. **(b)** Vibration amplitude: for the violin the amplitude grows with bow speed, while the slip amplitude of the relaxation oscillator is nearly independent of v_k and collapses at the transition to steady sliding.

vibrate. Increasing the speed of the bow increases the volume of the note, but not its pitch. Now take a chair, and push it across the ground. At low speed only a very low and quiet sound can be heard. Push it faster and the tone increases both in pitch and in volume. The difference lies in what sets the frequency of the stick-slip cycle. Friction-induced vibrations fall into two regimes: relaxation oscillation and self-excited vibration [2, 45], sketched in Figure 2.4.

The chair is the simpler case, and follows directly from the model of the previous section. Lacking a pronounced resonance, its motion is a *relaxation oscillation*, governed directly by friction and elasticity: the surfaces stick, elastic stress builds up as the chair is pushed, and slip occurs once the static friction limit is reached. The cycle time is essentially the time needed to reload the elastic element. Since this reloading happens at a rate proportional to the driving velocity, the period scales as $T \propto 1/v_k$, and the vibration frequency grows linearly with speed [82, 35]: the chair's judder rises in pitch as it is

pushed faster. This linear growth does not continue indefinitely. At higher speeds the slip duration becomes comparable to the period, the frequency saturates near the natural frequency of the elastically coupled slider [82]. The violin belongs to the second regime: the vibrating object possesses a strong, lightly damped resonance. During the so-called Helmholtz motion, the string sticks to the bow and is released once per round trip of the traveling wave, so the period of the stick-slip cycle is dictated by the string's own resonance. The bow merely feeds energy into this oscillation, which is why bow speed controls the amplitude (volume) but not the pitch [114, 2]. Brake squeal behaves the same way: the tone sits at a structural resonance of the brake assembly, and the driving conditions determine whether and how loudly it squeals, not at which frequency [59].

In practice, no system falls purely into either regime. Resonance lock-in is never perfect: hysteresis in the friction between bow and string lengthens the period of the Helmholtz motion at high bow force, so the note plays audibly flat unless the player compensates, a phenomenon known as the *flattening effect* [114, 76]. Conversely, a relaxation oscillator does not track the driving velocity forever: as noted above, its frequency saturates near the natural frequency of the coupled system. The contact between a robot's elastomeric finger and a slipping object sits near the relaxation end of this continuum: the elastomer is heavily damped and has no sharply tuned resonance, so the stick-slip frequency is not fixed but tracks the slip velocity. This is precisely what makes the vibration signature a candidate feature for slip velocity prediction. Conversely, it is also what limits its observability, as the frequency quickly exceeds the $25\,\mathrm{Hz}$ frame rate of the visuotactile sensor used in this thesis (Section 2.2.1).

2.2 Sensors

In this thesis, two sensors are used for slip inference. The first is a vision-based sensor, the GelSight Mini, and the second is the ADXL355 vibration sensor. In the following, their working principles and the reasoning behind why they were chosen will be explained.

2.2.1 GelSight

The GelSight was developed to map the 3D shape and texture of surfaces using tactile sensing, as initially introduced by Johnson et al. [53]. The sensor comprises a transparent elastomer slab covered with a reflective coating membrane. When an object is pressed

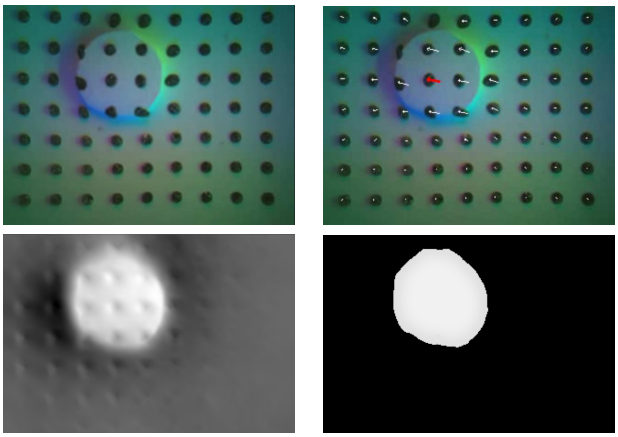


Figure 2.5: **Upper Left** Original Image of a circular indent **Upper Right** Motion flow using marker tracking **Lower Left** Depth map generated using photometric stereo algorithm. **Lower Right** Threshold of the depth map to determine the contact area.

against the elastomer, the membrane deforms to match the shape of the object's surface. Observed from behind, the membrane displays a relief replica of the surface. This relief is illuminated from three sides by red, green, and blue light, respectively. A camera is positioned directly below the gel, and can capture images of the deformation at 25 Hz, as can be seen in Figure 2.5. When optimized for resolution, the sensor's spatial resolution can achieve 1 µm to 2 µm [118]. For compact GelSight devices designed for robotic fingers, the spatial resolution typically ranges from 30 µm to 100 µm. [53, 118] There are different modalities the GelSight provides. For example, a photometric stereo algorithm can be employed to reconstruct the surface's depth map [113], shown in the lower-left panel of Figure 2.5. The GelSight's own code repository provides a method for estimating depth d_j for each pixel using neural networks and FFT-based Poisson solvers. This method, however, is not very accurate, and its output is only proportional to the depth with an unknown coefficient. Contact area can be estimated by obtaining a binary mask $d_j > d$ and then dilating it by D pixels (Figure 2.5, lower right). The central (non-peripheral) contact area is estimated by eroding the contact mask by E pixels.

One of the features of GelSight is the presence of easily distinguishable reflective dots (further: markers) on the surface of the elastomer. These markers can be easily tracked by classical computer vision methods, and their displacement yields the motion-flow field shown in the upper-right panel of Figure 2.5. It has been empirically determined that the

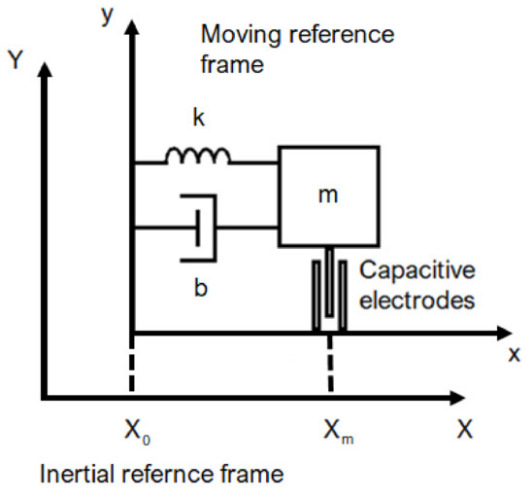
displacement of these markers is a good approximation for the shear force acting on the gel, as referenced by multiple researchers [121, 119, 69]

The main limitation of the GelSight for this work is its temporal resolution. At its native 25 Hz frame rate, the sensor can only resolve signals below the 12.5 Hz Nyquist limit, while the stick-slip vibrations described in Section 2.1.4 rise in frequency with slip velocity and quickly exceed this bound. The GelSight can therefore reveal that slip is occurring, but not measure the vibration itself. Capturing that signal requires a sensor with a bandwidth several orders of magnitude higher, which motivates the second sensing modality.

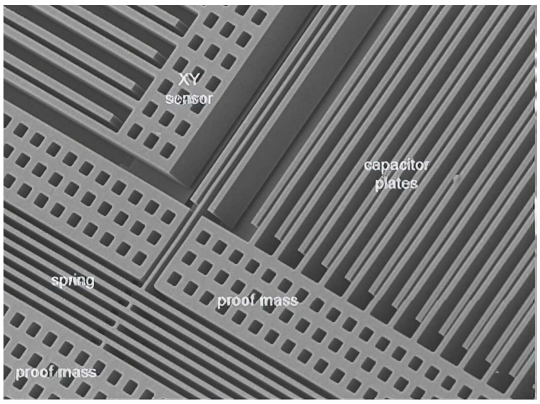
2.2.2 Accelerometer

The second sensing modality is an accelerometer, whose working principle is illustrated in Figure 2.6. The accelerometers used in this work are MEMS (micro-electro-mechanical systems) devices: a small proof mass is suspended on silicon springs inside the sensor housing, and an external acceleration deflects the mass relative to the frame. This deflection is read out capacitively through comb electrodes, whose capacitance change is proportional to the displacement and therefore to the applied acceleration [64, 106]. Because the proof mass reacts to any inertial excitation of the housing, an accelerometer mounted near the contact acts as a broadband vibration sensor: structural oscillations caused by stick-slip propagate through the gripper and appear directly in the acceleration signal.

The accelerometer used in this work is the ADXL355, a three-axis MEMS accelerometer designed for low-noise vibration measurement [3]. In this thesis, the ADXL355 will be referenced as the *vibration sensor*. Its sensing elements are fully differential: the x- and y-axes use lateral comb structures as in Figure 2.6b, while the z-axis uses a teeter-totter plate that tilts over fixed electrodes. The ADXL355 uses differential measuring, meaning the proof mass sits between two capacitors as seen in Figure 2.6(a). This has the upside that any drift due to temperature or age affects both capacitors equally, and is as such automatically compensated. With a noise density of $25\ \mu g/\sqrt{\text{Hz}}$, the sensor is precise for a consumer-grade device. For comparison the accelerometer used in the iPhone, has a noise density of $218\ \mu g/\sqrt{\text{Hz}}$ [17, 102], almost an order of magnitude higher.



(a) Mechanical model.



(b) SEM image of the microstructure.

Figure 2.6: Working principle of a capacitive MEMS accelerometer. (a) A proof mass m is suspended by a spring k and damper b inside the sensor housing (the moving reference frame). An acceleration of the housing displaces the mass relative to the frame, and this displacement is read out as a capacitance change at the electrodes. (b) Scanning electron microscope image of a fabricated sensing element, showing the perforated proof mass, the suspension springs, and the interdigitated capacitor plates. Both adapted from D'Alessandro et al. [18].

3 Related Work

Quantitative studies of friction can be traced all the way back to Leonardo da Vinci in 1493 [44]. The ensuing five centuries have produced a mature body of literature, but one that for the longest time still presumes to know material parameters and surface geometry. This thesis will take a learning based approach, with the following section detailing the research done in this regard.

3.1 Slip Frequency

The frequencies that are relevant when working with slip phenomena are dependent on the material and speed involved. It is theorized that humans can detect frequencies from $40\,\text{Hz}$ to $800\,\text{Hz}$ [46, 60]. This fact is not translatable directly to robotics, as for one skin is a very different material to the polymers and metals used in robots, and we don't know if that is the optimal detection range, or merely a biological limit. No consensus exists on which frequencies are relevant for slip. Authors often reason backward from their sensor's coverage, simply declaring it sufficient, which biases the reported ranges toward what their hardware happens to measure.

Yin et al. [117] found that for some materials and slip speeds, detection of vibration signals at $800\,\text{Hz}$ and above is necessary. Using pressure conductive rubber and an eddy electrode, Teshigawara et al. [105] found frequency components exceeding $1\,\text{kHz}$ during the incipient slip phase.

3.2 Vibration Sensor Alternatives

Accelerometers are not the only sensors that can be used for measuring slip vibration, and for this thesis different alternatives were explored. In the following section an overview of the different possibilities is given, including how they are used in the literature, strengths and limitations. A summary of all sensors can be found in Table 3.1

3.2.1 Vision-based tactile sensors

The working principles of vision-based tactile sensors have been discussed in depth in Section 2.2.1. In most cases, the low frame rate prevents these sensors from picking up relevant slip vibration signals. Even if higher-frame rate visuotactile cameras were available, the resulting massive data volume would make onboard processing impractical. For comparison, the GelSight Mini with its 320 x 240 resolution would produce about $0.46\,\mathrm{GB/s}$ at $2\,\mathrm{kHz}$.

There are cameras which can detect slip events at the required frequency: event-based cameras. These cameras don't have a fixed frame rate; instead, they capture brightness changes (events) in the scene rather than frames. Each brightness change triggers a signal, which allows high temporal contrast sensitivity that captures events in microseconds [93, 74]. This suffices to capture slip vibrations, while keeping the data rate under $1\,\mathrm{MB/s}$.

But even Funk et al. [28], who used such a sensor, were only able to detect vibrations up to $498\,\mathrm{Hz}$. While this does suffice to characterize the $40\,\mathrm{Hz}$–$300\,\mathrm{Hz}$ frequency range that is most often cited as being relevant for slip, it still falls short of being able to detect frequencies larger than $1\,\mathrm{kHz}$ relevant for some incipient slip. In addition, event based cameras are too cost prohibitive for most use cases.

3.2.2 Piezoelectric sensors

Piezoelectric materials generate an electric charge when mechanical stress is applied. Since their frequency response is limited only by the mechanical resonance of the device, piezoelectric elements can be designed to operate at very high frequencies, with surface and bulk acoustic wave devices reaching into the gigahertz range [116]. The sensitivity of a piezoelectric element decreases as its resonance frequency, and thus its usable bandwidth, increases. As a result, sensor designs trade bandwidth against signal strength. Piezoelectric

sensors used in robotics are therefore typically designed for bandwidths in the kilohertz range, which retains high sensitivity while comfortably covering the frequencies relevant for slip detection.

Piezoelectric sensors come in a variety of form factors. Specifically for dynamic forces like vibrations, piezoelectric polymer film or ceramic diaphragms are often used. The main advantage of these are their affordability and possibility of mounting them essentially anywhere. Polymer film has the advantage here, with the tradeoff of being more receptive to noise, temperature, and other external factors. Both can't measure static forces, and are very sensitive to changes in their mounting configuration. Because of their small size and as such high natural resonance frequency, their measurement range spans roughly 1 Hz to 10 kHz for polymer film and 10 Hz to beyond 10 kHz for ceramic diaphragms. Piezoelectric diaphragms are the transducers in contact microphones, which are often specially designed to be more sensitive and noise resistant in specific frequency ranges.

If one wants to measure lower frequencies or measure static forces, piezoelectric pressure sensors have to be used. These are mounted in a way that a static pressure also produces a static signal, and their larger size enable lower frequencies to be detected, with the tradeoff that the frequency range is often only from 0 Hz to 2 kHz.

Piezoelectric transducers are a classical slip detection method. Dornfeld and Handy [22] detected slip from acoustic emission as early as 1987, establishing that the stick-slip transition has frequencies well above the frequencies a force sensor resolves. Howe and Cutkosky [42] instead used a polymer film on a digit as a high frequency stress sensor, and showed that slip features can be derived from the dynamic stress changes captured by the sensor. Tremblay and Cutkosky [107] used the same principle to estimate friction from incipient slip during a manipulation task, and Fujimoto et al. [27] embedded polymer film in an artificial finger skin to detect incipient slip. Fishel et al. [25] also showed that a dedicated micro-vibration sensor in a biomimetic fingertip is robust enough to detect slip. The frequency content these works exploit affirms the frequency bands described in Section 3.1. The frequency components beyond 1 kHz that Teshigawara et al. [105] report during incipient slip are covered comfortably by a piezoelectric element and not at all by a frame-based visuotactile sensor.

Gloumakov et al. [31] also present a promising approach, modelling the contact as a stick-slip friction oscillator, they recover the dominant frequency of a capacitive tactile signal by a windowed DFT and show that it rises monotonically with pull speed between 10 mm/s to 100 mm/s for cherry and basswood. For acrylic no consistent trend emerges at all, because adhesion between the surface and the silicone skin violates the assumptions of the model, which is precisely the case in which a learned mapping is preferable to

a fitted one. Their sensor can also electrically couple its tactile pixels (taxels) to trade spatial resolution for sampling rate, about 30 Hz in the individual mode against 500 Hz to 2000 Hz when coupled, and the faster mode prevented an average of 10° of additional rotation before the slip was arrested ($p < 0.001$).

The main downside with piezoelectric sensors is the difficulty in mounting them, making it difficult to reproduce setups that utilize them. The sensing area is often only a few square millimeters, so the element must sit precisely where the relevant vibrations occur. A piezoelectric element also only responds to the stress transmitted into it, and the mounting determines the contact stiffness between the element and the structure, which together with the sensor mass forms a mechanical resonance that filters the signal. A loose mount lowers this resonance and attenuates the higher frequencies, so both the measured amplitude and the frequency content of a slip event can shift substantially depending on how and where the element is fixed. A given characterization is therefore valid only for its specific mounting and is hard to reproduce across grippers, making it less ideal for learning based approaches.

3.2.3 Laser Doppler Velocimeter

The laser Doppler velocimeter is worth mentioning, since this sensor directly measures what this thesis is trying to infer: slip velocity. However, the sensor's only use case has been in the work which created it, namely Morita et al. [80]. A laser Doppler velocimeter (LDV) measures the velocity of a surface optically, by irradiating it with two crossed laser beams and evaluating the Doppler shift of the scattered light. Morita et al. [80] miniaturized this principle into a 2.8 mm $\times$ 2.8 mm $\times$ 1 mm MEMS chip that fits into a gripper fingertip, and used it for slip-based grasping force control. The main advantages are that the measurement is contactless, so the friction conditions at the measurement point do not influence the reading, and that the sensitivity and bandwidth are high: the sensor output oscillates one cycle per 1.62 µm of target displacement, and with a 20 kHz sampling rate a slip decision was made every 50 ms at a velocity resolution of 32.5 µm/s.

The disadvantages currently make its widescale deployment difficult. Detection reliability depends strongly on the optical properties of the target: while slip on an aluminum block was detected during 100% of the slipping time, the rate dropped to 47% for wood and 37% for white acrylic due to lower signal-to-noise ratios, and a polished transparent acrylic surface produced no usable signal at all. Furthermore, the sensor only works within a narrow sensitive distance of 0.9 mm to 1.5 mm from its surface, and the FFT-based processing imposes a trade-off between response time and velocity

resolution. Conventional LDVs also remain far too large for fingertip integration, with the smallest commercial device measuring $28\,\text{mm} \times 18\,\text{mm} \times 60\,\text{mm}$. Considering that entry-level, normal-sized vibrometers start at 3000 dollars, small-scale variants are still not commercially viable.

3.2.4 Barometric Sensors

Barometric sensors are sealed MEMS pressure dies, that are cast into an elastomeric fingertip causing contact forces to be transmitted to the die as a pressure change. TakkTile-type sensors follow this principle [104], and slip detection with them has been demonstrated [57]. These systems are limited by the polling rate of the digital pressure readout and by the stiffness of the elastomer, which low-pass filters the vibration before it reaches the die, placing their usable bandwidth at the lower edge of the slip band.

Table 3.1: Sensing modalities for detecting slip-induced micro-vibrations (∼0.1–1 kHz band) at a robotic gripper fingertip; see [26, 12, 95] for comprehensive surveys.

Modality	Advantages	Limitations
Piezoelectric polymer film (PVDF) *∼1 Hz–10 kHz*	Very high dynamic sensitivity; thin and conformable, mountable at the contact surface above compliant pads; self-generating (no excitation); established in stress-rate and incipient-slip sensing [42, 27].	No static response; requires high-impedance charge/voltage buffer; pyroelectric cross-sensitivity to thermal transients; fragile metallization; output depends strongly on lamination and preload [26].
Piezoelectric ceramic diaphragm (PZT) *∼10 Hz–>10 kHz*	Extremely low cost; high charge output (relaxed front-end requirements); mechanically robust [95].	Stiff and bulky relative to a fingertip; strongly resonant (colored) response complicates spectral features; response highly sensitive to mounting boundary conditions.
MEMS accelerometer on finger structure *0–1.6 kHz (consumer); to 11 kHz (ADXL1002)*	Outside the force path (no fingertip redesign); trivial mounting; measures structure-borne slip vibration, a principle established by skin-acceleration sensing [41] and accelerometer-instrumented slip nibs [107]; digital variants ease integration.	Also transduces actuator gear-mesh and base vibration, requiring band-selection or motion gating; consumer parts limited to ∼4 kHz; sensitivity decreases with distance from contact.
High-rate sampling of piezoresistive force/pressure sensor *0-∼2 kHz*	Single transducer covers static force and slip band (digital crossover), as demonstrated by the BioTac's AC/DC pressure channels [25] and dynamic tactile arrays [36]; signal originates in the force path where slip occurs; MEMS variants are calibrated and linear.	Lower AC sensitivity than piezoelectric elements; compliant fingertip pads low-pass the vibration before it reaches the die; bridge thermal noise bounds detectable micro-slip amplitude; needs kHz ADC and analog anti-aliasing.
Acoustic emission / contact microphone *∼100 Hz–>20 kHz*	Very high sensitivity to stick–slip acoustic emission, exploited since early AE-based slip detection [22]; wide band captures high-frequency precursors of gross slip; inexpensive.	Susceptible to airborne ambient noise and actuator acoustics; requires careful acoustic coupling and isolation; limited literature on calibration/transferability across grippers [95].
Barometer embedded in elastomer (TakkTile-type) *0–∼0.5 kHz*	Robust, overload-tolerant, cheap [104]; combined static pressure and dynamic event detection, with demonstrated slip capability [57]; conformal contact surface built in.	Bandwidth marginal for the upper slip band; requires elastomer casting; sensitivity depends on cast geometry (unit-to-unit variation) [104].
Vision-based tactile skin *Frame-rate limited (∼30–120 Hz)*	Dense spatial contact field; detects *incipient* slip directly from partial stick–slip displacement/shear field rather than vibration proxies [119, 20, 48].	Cost, volume, and computational load incompatible with a low-cost servo gripper; temporal resolution below the vibration band; illumination and gel wear [118].
Actuator-intrinsic sensing (baseline) *≲50 Hz (bus-limited)*	Zero added hardware; useful for contact detection and gross load changes.	Bus latency and internal filtering exclude the slip-vibration band; a 258.5:1 geartrain attenuates and masks fingertip micro-events; load signal is a control-effort estimate, not a measurement [94].

3.3 Learning Based Slip Detection

Combining visuo-tactile sensing with learning based approaches seems to be where most of the current research is being done. Most work is also on slip prevention, with research often focusing on incipient slip detection to prevent objects from sliding in the gripper in the first place.

According to google scholar the most cited paper in this domain is “Slip Detection with Combined Tactile and Visual Information” by Li et al. [68]. They use a multilayer LSTM network to fuse a visuotactile and external camera image using pre-trained image encoders. They achieve 88% slip detection across a wide array of objects.

Gao et al. [30] follow a similar approach, but replace the recurrent backbone with a multi-scale temporal convolution network. Their CNN-MSTCN passes the external camera image through a frozen ResNet34 and the XELA tactile array through a small three-layer CNN, reduces both to 64-dimensional features, and runs each through its own multi-scale temporal convolution branch before concatenating them into a shared fusion branch. Across 50 objects, of which 10 are held out for testing, they report 96.96% accuracy and an F1 score of 97.06%, against 93.69% for their reimplementation of the network of Li et al. [68]. Though their fusion approach is not as good as the numbers might suggest, the vision branch on its own already reaches 96.00%, while the tactile branch on its own falls to 75.51% at a recall of 51.67%. Fusing both modalities therefore buys just under one percentage point over the camera by itself. This is likely due to the fact that in their lift-and-hold trials, a slipping object is simply an object that visibly moves, and the tactile stream is left with little to contribute.

The limitation both works share is the binary label. Wang et al. [110] looks at slip labeling further, training an ensemble of five GRU classifiers on the tangential pillar forces of a PapillArray sensor, an array of nine silicone pillars sampled at $1\,\mathrm{kHz}$. Their ground truth definition is the most rigorous in this literature: incipient slip is the state in which at least one pillar is already sliding while at least one other still sticks. They detect it within roughly $10\,\mathrm{ms}$ of its onset, at 95.6% accuracy offline and 96.8% across 216 trials on a real robot fitted with a different variant of the sensor. Even so, they conclude that the binary output is itself the bottleneck and call for replacing it with a continuous warning signal, since slip confined to a small portion of the contact may still leave enough friction to hold the object. Chen et al. [13] and Jawale et al. [49] implement exactly that and regress a continuous slip velocity instead. Chen et al. estimate sliding velocity from fingertip tactile signals alone across 28 objects, reaching a mean squared error below $0.5\ (\mathrm{cm/s})^2$ while

generalising to unseen shapes and textures. They also managed to use this slip velocity prediction to control slip, showing that 0.5 (cm/s is enougth for slip velocity control. Jawale et al. extract the divergence, curl and contact area of the marker deformation field of a GelSight Mini and feed them to a three-layer LSTM, obtaining a mean absolute error of $0.26\,\mathrm{cm/s}$ at an R^2 of 0.75 over slip speeds from $0.8\,\mathrm{cm/s}$ to $6.7\,\mathrm{cm/s}$. Both label their ground truth with an external optical tracker, motion capture in the former case and ArUco markers in the latter, and both degrade sharply on compliant objects, where Jawale et al. report as little as 43.71% detection accuracy on a sponge. This is the closest to what this thesis is trying to set up, though Chen et al. used a BioTac sensor, which is not visuotactile and can poll in the kilohertz range.

Taken together, the works separate along three properties: whether the target is a continuous velocity or a binary label, whether the input reaches the vibration band, and whether the tactile stream is visuotactile. None of them combines all three. What remains open is the pairing of a visuotactile stream with a dedicated high-bandwidth vibration channel, regressed onto a continuous slip velocity rather than a label, which is the combination Chapter 5 sets out to build.

3.4 Data Fusion

Section 3.2 established the current research into the different sensor modalities. Each modality comes with its own frequency and representation, which makes fusion inherently challenging. This section establishes the academic backing behind the architectures used later in Chapter 5. The goal is to show what different approaches exist, even if they are not in the field of slip detection or do not use vibration or visuotactile signals.

3.4.1 Fusion Structure

When to fuse the data? In contemporary literature there are three different approaches: Early, late, and intermediate [92]. None of the three types is inherently better and their performance depends on the use-case.

Early Fusion

Also called feature-level fusion, early fusion combines the signal vectors into a single feature vector before it is passed into the neural network. This often comes with a whole set of challenges, such as dealing with different representations, frequencies, and data types.

As Sebe et al. [99] and Poria et al. [90] argue, early fusion is very computationally efficient, and especially useful for usecases where the modalities have similar polling rates, like multispectral imagery [108]. In addition, this method essentially forces cross-attention between the different modalities. Downsides, depends on the implementation details, but according to Ramachandram et al. [92] the fact that no feature extraction happens on any individual stream can degrade performance.

Late Fusion

Late fusion, also called decision-level fusion takes the opposite approach. Each modality gets its own MLP "branch" and the branches are combined using a decision function. The decision can be max/average-fusion, Bayes' based or any other function that can transform the branch encoding and turn them into an output.

Late fusion is popular when analyzing Human interaction, used to fuse audio, image, and text data like in the cases of emotion recognition [55], sentiment analysis [90], gesture recognition [115] and action recognition [101], but not limited to that domain [75, 92].

The main downside is that there is very little cross-attention between the modalities, with only the decision function having access to all modalities. A shortcoming that intermediate fusion fixes.

Intermediate Fusion

Intermediate fusion adapts the branch philosophy and enables cross talk by inputting the output embedding of each branch into an MLP. Modalities don't have to be fused all at once, and Neverova et al. [83] showed that fusing correlated modalities first produced better results than fusing them at once. Intermediate fusion is popular in medical settings like modality fusion in medical devices like MRI and PET fusion [71] and medical image

retrieval [11], but has also been successfully deployed for speech classification [84] and robotic grasping [67].

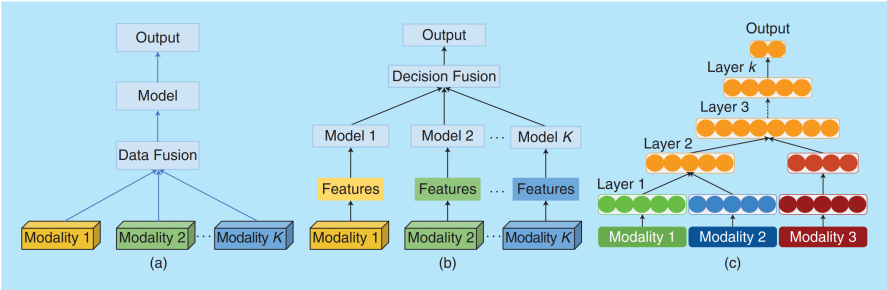


Figure 3.1: Different fusion structures, taken from Ramachandram et al. [92].

Each type of fusion comes with its own tradeoff, and while intermediate fusion is usually the safest choice, this thesis will implement networks utilizing each type.

3.4.2 Transformer Based architectures

The main reason transformer based architectures are considered in this thesis are that they are both, good at dealing with different frequency data, and token representation lends itself to multimodal fusion.

Perceiver IO [47] is one of the more known examples of the ”tokenize everything, fuse with transformer” philosophy. They fuse image, audio, and 3d point cloud data using a learned latent array with cross-attention similar to GPT 2 [91] and improved the mean average precision of audio event classification by almost 60% compared to using only a single modality.

The main method for dealing with variable frequency modalities is encoding the tokens of each sensor modality. This approach has been shown to work effectively in both Kazemi et al. [56] who used it on various asynchronous time-series data, and Horn et al. [40] who added a continuous time embedding to each measurement from medical data.

Sparsh-X [37] is a neural network that works on multisensory touch representation, combining visuo-tactility, contact-audio, motion and pressure. Sparsh-x served as an inspiration for the transformer based implementations in this thesis. Sparsh-X uses raw sensor inputs with each modality tokenized independently: batch embeddings for the

image and audio streams, chunked embeddings for the accelerometer and pressure signals. Each modality is passed through a few tokenization layers.

Each token branch is passed individually through a transformer. For cross-modality ”bottleneck tokens” [81] are used, which are tokens that are shared between different transformers. In the last step all tokens are concatenated and passed into a fusion transformer.

Sparsh-X also showed that contact-audio based inference was 36% more accurate than visuotactile based inference classifying surface properties. Though with manipulation tasks, including visuotactile sensing improved the success rate from 20% to 83%.

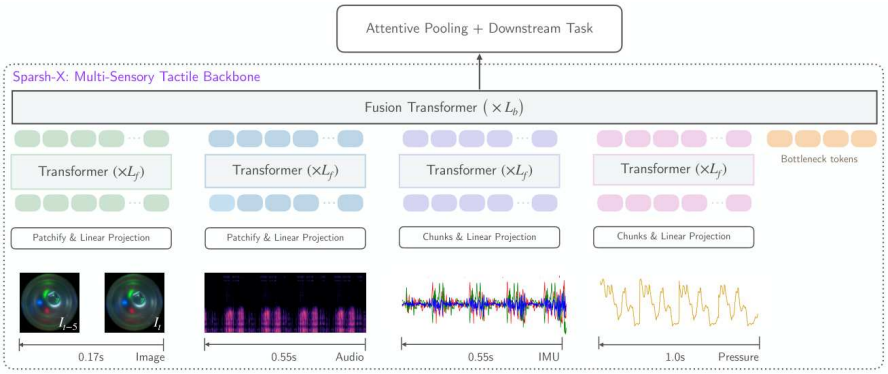


Figure 3.2: Sparsh-X neural architecture, taken from Higuera et al. [37]

4 Experimental Platform

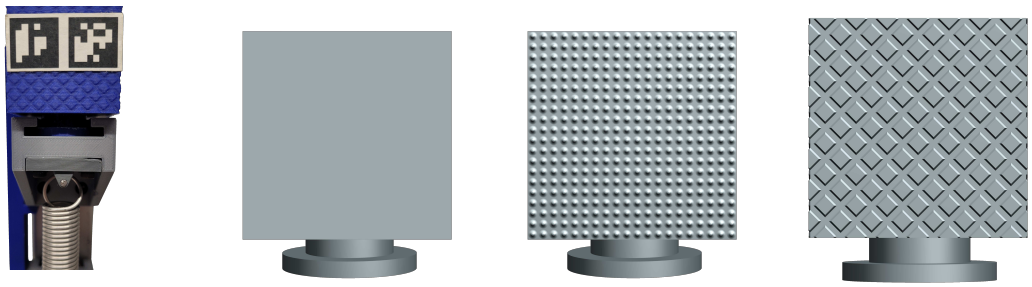
The test bench was designed to provide a low-cost, reproducible setup for generating labeled slip data. The test bench simulates a robotic gripper gripping and lifting an object and it was built in collaboration with List et al. [70] in a previous project. To isolate the slip and vibration signals, we opted to keep the hand itself stationary, and instead manipulate the object inside the gripper to simulate movement.

The test bench has to accurately:

1. Induce forces on the gripped object.
2. Apply a controlled, repeatable grip (normal) force.
3. Measure the velocity of the object and record it as the independent ground truth of slip velocity.
4. Correlate object movement to the vibration sensor and GelSight image.
5. Isolate slip signals from external and self-generated vibrations and time-synchronise all sensor streams.

4.1 Test Object

Slip vibration and the signals gathered by visuotactile sensors depend heavily on the gripped object's surface: glass offers little texture for the visuotactile sensor to infer from, while slick surfaces can bury slip vibrations in noise. To accommodate objects of varying topology and surface, the test object couples to the test bench through a form-fit connection: a round footing on the object mates with a receptacle in the carriage, allowing toolless insertion and removal. Objects can be 3D printed with the male fitting directly, or



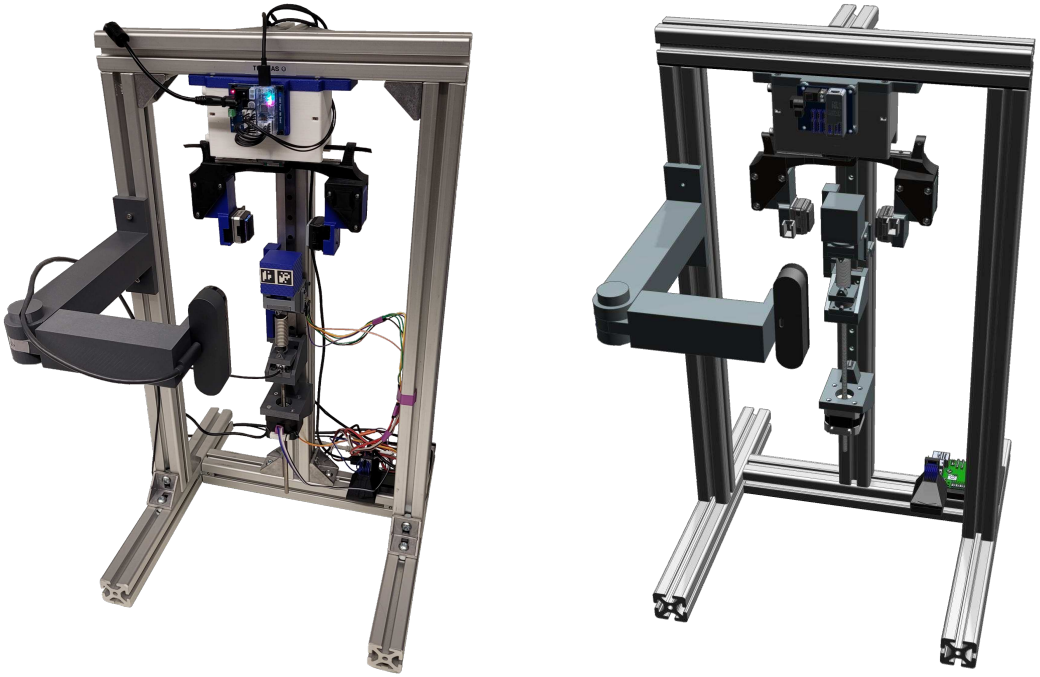
(a) Form-fit carrier mount. (b) Smooth surface. (c) Studded surface. (d) Squares surface.

Figure 4.1: The form-fit carrier mount (a) and the two test objects used in this thesis, with smooth (b), studded (c), and squares (d) contact surfaces [70]. The smoothe surface was not used.

fitted with a male-to-receptacle adapter. For this thesis, two cubes with different surface textures (Figure 4.1) were used with this carrier mount.

4.2 Frame and Mounting

The frame of the test bench is constructed from commercially available aluminium item profiles. This allows flexible positioning of sensors and actuators, as well as easy expandability. Silicone feet reduce vibrations from surrounding sources, though the vibration sensor is sensitive enough to register small disturbance steps even through the dampening feet. The profiles are arranged in a way to make the whole system rigid and torsion resistant. A MGN12H linear guide rail guides the sled that the object is mounted on. During usage, the rail must be kept properly lubricated, as a sled running on an unlubricated rail produces substantial vibrations in the same frequency range as slip.



(a) Physical prototype. (b) Annotated CAD model.

Figure 4.2: Test bench overview: (a) physical prototype and (b) annotated CAD model with labelled components [70].

4.3 Actuators

The test bench has two main actuators, the linear NEMA 17 motor responsible for inducing a pulling force on the test object, and the gripper. In the following, both will be explained.

4.3.1 Gripper

The gripper is an actuated Universal Manipulation Interface (UMI) Gripper by Helmut et al. [34], which in turn is based on the UMI by Chi et al. [14]. It's driven by a DYNAMIXEL XL430-W250-T servo motor.

The UMI comprises a parallel jaw with tendon- and spring-actuated fingers. In its released state two springs push the fingers into their open position. To close the gripper, the motor winds in a belt attached to the fingers, closing them. Control is entirely position driven.

Communication is done via a half-duplex serial bus at $50\,\text{Hz}$. The gripper is directly connected to the PC. The reasoning is that the slip computation also happens on the PC, and this reduces the latency from signal to actuation compared to first having to send the signal to the Raspberry Pi.

The major downside to this motor is that it does not have any voltage sensors and as such no impedance control. The motor does present a "load" readout. The load value is essentially the position error of the motor. As the fingers press down on an object, the expected position deviates from the actual position, which is reported as the current load. A grip is established by increasing the Pulse Width Modulation (PWM), effectively the power, of the gripper, until a desired load is reached. Then the position is maintained.

4.3.2 Linear Motor

The purpose of the linear motor is to apply a force to the test object. The linear motor in the test bench is a NEMA17 stepper motor, driven by a TMC2208. The motor pulls down a platform attached to the test bench via a sled. On the sled is a mounting point for a spring, which in turn is attached to the test object. The maximum pulling force is determined by the strength of the spring.

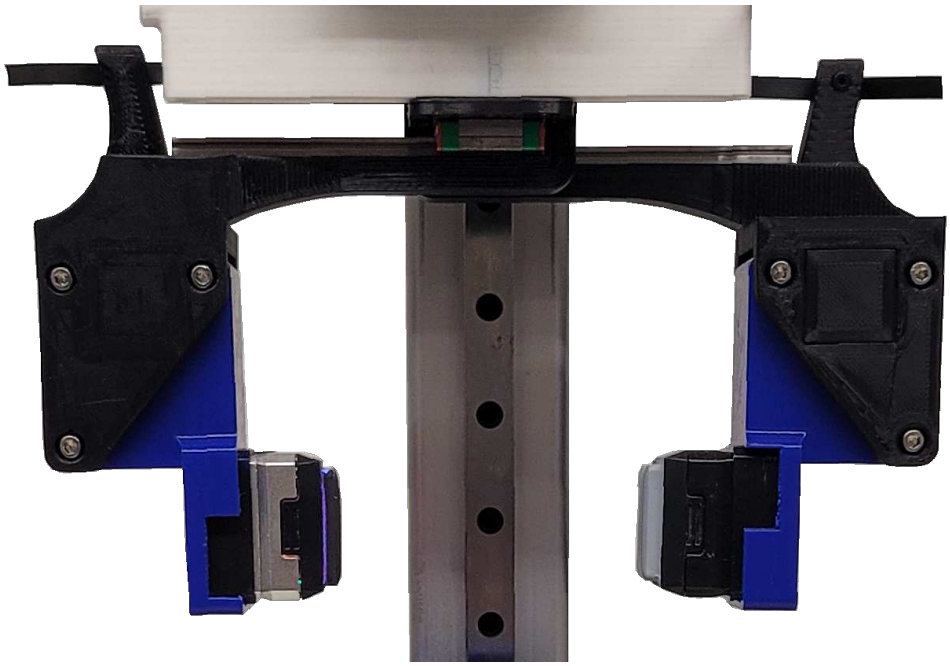


Figure 4.3: The fingers of the UMI with the GelSight mounted in the right finger [70].

4.4 Sensors

4.4.1 Vibration Sensors

The test bench has two vibration sensors. The primary vibration sensor is mounted on the gripper and collects the data for inference. The second *vibration noise sensor* is mounted away from the gripper, on the base of the test bench and serves to collect the noise floor and external disturbances.

In this work both sensors operated at their maximum output data rate of $4\,\mathrm{kHz}$. This choice follows from the sensor's filter chain rather than from the Nyquist criterion alone: the internal digital low-pass corner sits at approximately one quarter of the output data rate, so only at $4\,\mathrm{kHz}$ does the resulting $1\,\mathrm{kHz}$ passband leave the slip-relevant vibration band of up to several hundred hertz unattenuated [3]. Sampling at the full rate also preserves anti-aliasing headroom for post-processing. The cost is a larger integrated noise floor, though the ADXL355 has a very low noise density of $25\,\mu\mathrm{g}/\sqrt{\mathrm{Hz}}$ (typical, per axis,

$\pm 2\,g$ range), even with its higher polling rate. Compared to the 12.5 Hz Nyquist limit of the GelSight, the vibration sensor extends the observable band by nearly two orders of magnitude and covers the stick-slip vibration regime described in Section 2.1.4.

Filtering Method

Before either signal is passed to the network, the noise sensor is used to remove the shared disturbance floor from the primary sensor's signal. The scheme follows the classical adaptive noise cancelling architecture of Widrow et al. [112]: a primary sensor observes signal plus disturbance, a reference sensor observes a correlated, ideally signal-free version of the same disturbance, and an estimate of the disturbance is subtracted from the primary channel. This is implemented as a two-stage filter, a fixed calibrated filter followed by a slow online correction.

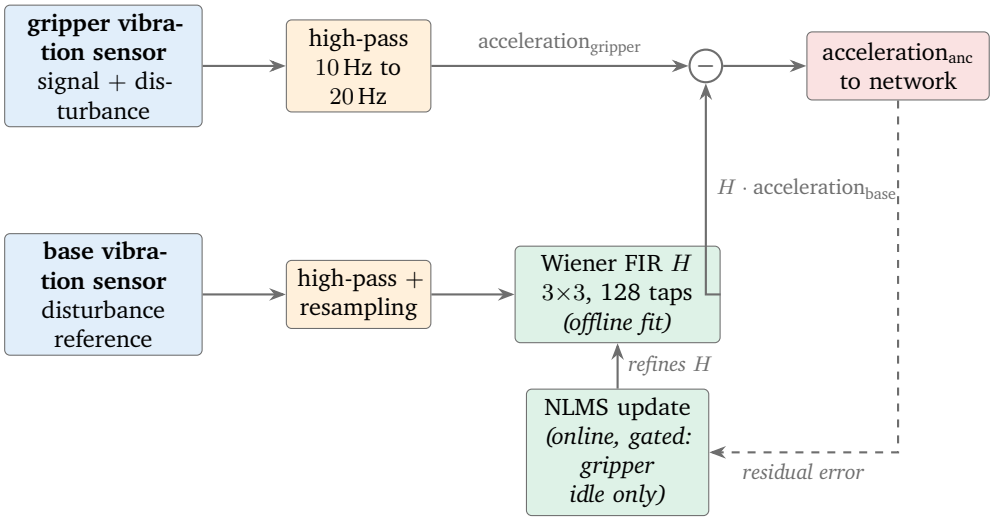


Figure 4.4: The two-stage disturbance-cancellation pipeline.

Before either filter stage is designed, the frequency bands over which the two sensors are usefully correlated are identified with a magnitude-squared coherence estimate,

$$C_{gb}(f) = \frac{|S_{gb}(f)|^2}{S_{gg}(f)\,S_{bb}(f)}, \tag{4.1}$$

computed with Welch's averaged periodogram method [111] over roughly one-second segments, where S_{gb} is the cross-power spectral density between the gripper and base

channels and S_{gg}, S_{bb} their respective auto-spectral densities. $C_{gb}(f)$ is evaluated separately for the gripper at rest, for the stepper motor running with no object grasped, and during a slip event, which shows which bands carry cancellable, shared disturbance rather than slip information that happens to correlate between the two mounting points. The companion frequency-domain estimate $H(f) = S_{gb}(f)/S_{bb}(f)$ gives a first look at the structural transfer function of the rig between the base and the gripper before a time-domain filter is fit.

In the first stage, a causal multiple-input multiple-output Wiener FIR filter [33] is fit offline. The filter is a 3×3 matrix of per-axis-pair FIR filters, so that cross-axis coupling introduced by the rig's structure can be captured in addition to the direct per-axis path. Its taps are estimated by ridge-regularised least squares, regressing each gripper axis on lagged samples of all three base axes using calibration recordings taken with the gripper unloaded, so that the fit cannot absorb any slip-related content. The tap count is swept from 64 to 512 and fixed at the knee of the held-out variance-reduction curve; 128 taps per axis pair was sufficient in this setup. The compensated signal is then

$$\text{acceleration}_{\text{anc}} = \text{acceleration}_{\text{gripper}} - H \cdot \text{acceleration}_{\text{base}}. \tag{4.2}$$

H is validated by checking, on held-out no-grasp recordings, the reduction in decibels within the coherent bands identified above, and by confirming that the slip-band spectrogram of a known slip event is left unchanged by the filter.

Should the coupling between the two sensors drift within a session, for instance due to thermal changes in the mounting structure, a second stage refines H online with a normalised least-mean-squares (NLMS) update [33], initialised from the offline fit of the first stage. This adaptation is gated to periods in which the gripper is idle, i.e. no grasp or motion is commanded, so the filter never adapts to the slip signal itself, which would otherwise be misidentified as correlated noise and partially cancelled. It is this compensated signal, rather than the raw primary-sensor reading, that is passed downstream to the network.

4.4.2 GelSight

The GelSight is mounted in the right finger of the UMI actuator, described in Section 4.3.1. The GelSight Mini is mounted on one jaw via a custom 3D-printed mount that can be rotated. This way, the GelSight can be positioned in any position relative to the linear actuator.

The details of the working principles behind the GelSight can be found in Section 2.2.1, while the preprocessing algorithms used can be found in Section 5.2.1.

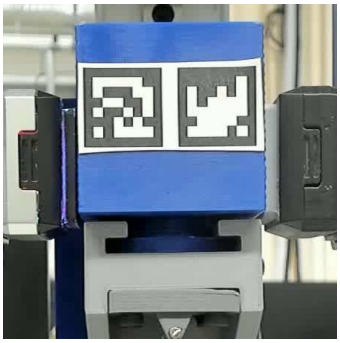
4.4.3 Webcam

The webcam, a Logitech BRIO, is used to determine the current position and velocity of the object. The webcam is also directly connected to the PC. For this, ArUco markers mounted to the test object are used. For increased accuracy, and as a fallback if one does not get detected, two markers are used.

The spatial resolution of the webcam-based tracking is determined by back-projecting a single pixel onto the plane in which the marker moves. With the calibrated focal lengths $f_x = 1143.4\,\text{px}$ and $f_y = 1144.2\,\text{px}$ and a working distance of $Z_c = 176\,\text{mm}$, this yields a nominal resolution of approximately $0.154\,\text{mm}$ per pixel in both image directions. Because the marker position is estimated from ArUco corners refined to sub-pixel accuracy and averaged over all detected corners, the smallest reliably detectable in-plane motion is finer than this single-pixel bound, on the order of a few tens of micrometres. It should be noted that this resolution scales linearly with the working distance, since $\Delta X \propto Z_c$, and that it applies only to motion within the plane at distance Z_c; out-of-plane (depth) motion is not observable from a single marker centroid.



(a) Webcam on articulated arm



(b) Webcam frame with ArUco markers

Figure 4.5: Webcam setup [70].

4.4.4 Force Torque Sensor

The force torque sensor mainly serves to qualitatively validate the ground truth as provided by the camera, and is not used in any training data. While it can't determine slip velocity, it can validate if slip is happening. The sensor we used is a Resense Hex21. It is a 6-axis F/T sensor with a $1\,\text{kHz}$ sample rate. As seen in Figure 4.6, it can either be mounted directly at the object, or at the motor. In both cases, it can work as a secondary vibration sensor, with the mounting next to the motor picking up and quantifying the motor noise, while the position next to the test object can pick up slip vibrations. In addition, it can also accurately determine when slip starts and stops. When the linear actuator moves down, the pulling force on the object increases. As soon as it slips, the force decreases, as the spring is relaxed again. So any negative acceleration in the force is a sign of slip, while any zero acceleration is the object sticking.

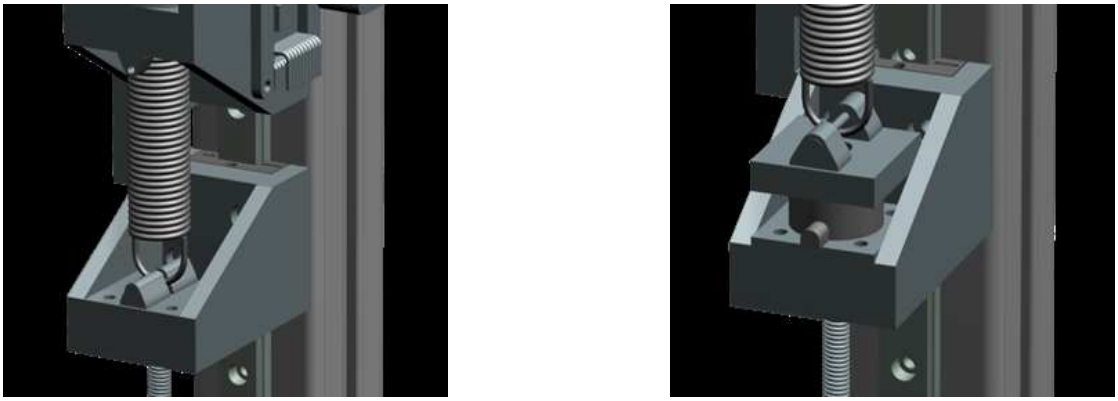


Figure 4.6: Force transmission setup without the force torque sensor (left) and with the HEX21 sensor mounted between actuator and test object (right) [70].

4.5 Synchronisation

A major problem with the test bench at the start of the thesis was sensor synchronisation. Using the USB interface, the latency between sensors and the PC was usually around $2\,\text{ms}$, but could be as high as $30\,\text{ms}$ during stutters. USB is a host-polled, shared bus: a peripheral cannot initiate transmission but must wait for the host controller to allocate it a slot within the bus's frame schedule, introducing scheduling jitter, while serial-class traffic

becomes unpredictable when the bus is shared with isochronous devices such as cameras. This was a problem, as the occasional 30 ms desync would mean that the camera images would become desynced by a frame, and in the case of higher frequency data like the force sensor running at 1000 Hz, the data would be misaligned by 120 samples. For this reason the new test bench uses Ethernet to connect the Raspberry Pi to the computer. This allows synchronisation via Precision Time Protocol (PTP, IEEE 1588) over the direct Ethernet link. Due to a defect with the motherboard of the host PC, hardware PTP timestamping was not possible, which would have netted a sub-1 µs time accuracy. With the currently implemented software timestamping the timing difference is around 30 µs. Even in the worst observed case, due to the duplex nature of Ethernet, the clock offset remains well below the 250 µs sample period of the fastest sensors, so consecutive samples of two 4000 Hz vibration sensors can still be matched unambiguously.

4.6 Wiring and Connections

The following will explain the wiring and connection setup of the test bench. The hardware control is split between the Host PC and a Raspberry Pi 5 (RP5). Both are connected via Ethernet, and synchronised via Precision Time Protocol as explained in Section 4.5. The RP5 runs the ROS 2 Jazzy Jalisco middleware and interfaces with all peripherals:

- Two ADXL355 vibration sensors via SPI
- TMC2208 stepper driver via UART
- Servo motor via PWM (GPIO)
- Resense HEX21 via USB
- Logitech BRIO webcam via USB
- GelSight Mini via USB

Figure 4.7 shows the complete wiring and connection diagram for the test bench.

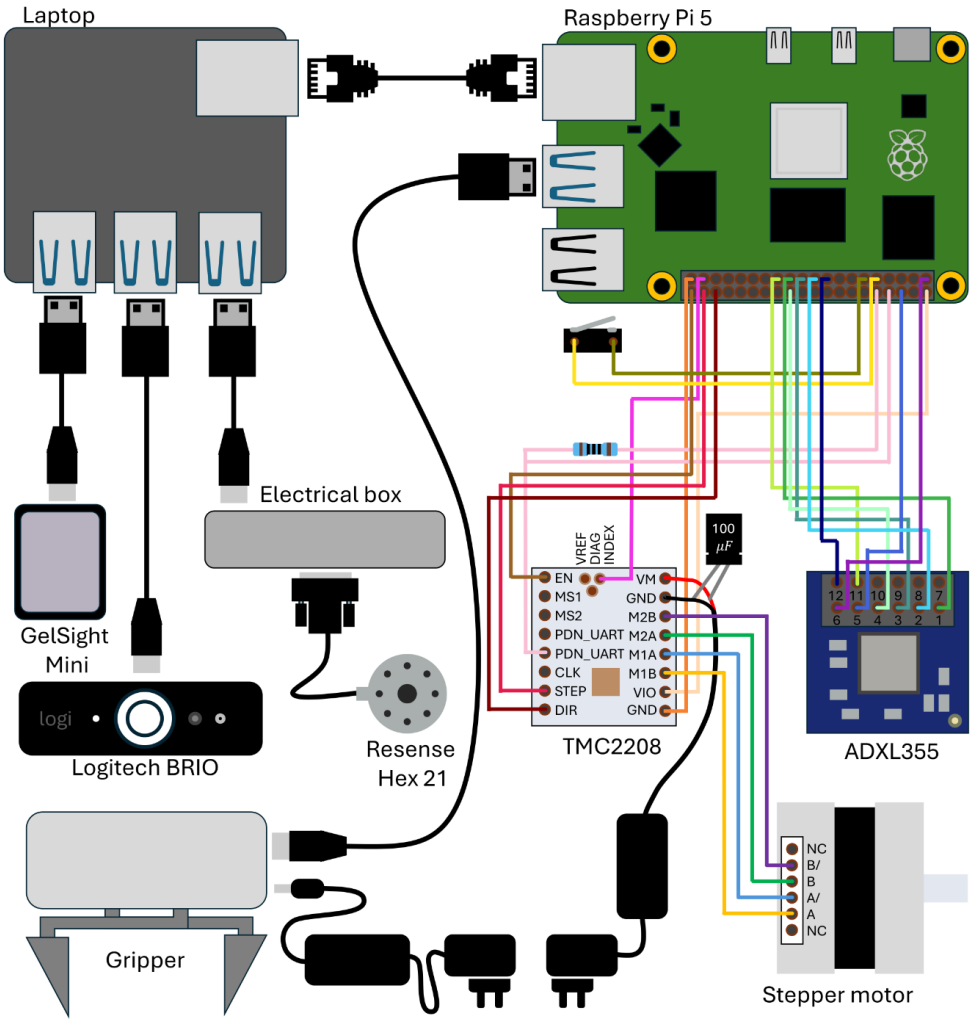


Figure 4.7: Complete wiring diagram showing connections between the Raspberry Pi 5, stepper driver (TMC2208), servo, ADXL355, and all USB peripherals [70]. Both ADXL355 sensors connected to the same pins, with the exception of the Chip Select. The second ADXL355 is not depicted.

Table 4.1: Pin mapping between TMC2208 and Raspberry Pi GPIO.

TMC2208 Pin	Function	Raspberry Pi GPIO Pin
VM	Motor Supply Voltage	External 19.5 V DC, 3.3 A Power Supply
GND	Ground	External 19.5 V DC, 3.3 A Power Supply
M2B	Motor Coil 2 B/	(Pin 2 – Stepper Motor)
M2A	Motor Coil 2 B	(Pin 3 – Stepper Motor)
M1A	Motor Coil 1 A/	(Pin 4 – Stepper Motor)
M1B	Motor Coil 1 A	(Pin 5 – Stepper Motor)
VIO	Logic Supply Voltage	5 V (Pin 2)
GND	Ground	GND (Pin 39)
EN	Enable Motor Outputs	GPIO 21 (Pin 40)
PDN_UART	UART Communication	UART TX (Pin 8) & UART RX (Pin 10)
STEP	Step-Signal Input	GPIO 16 (Pin 36)
DIR	Direction-Signal Input	GPIO 20 (Pin 38)
Index	Index Output	GPIO 26 (Pin 37)

Table 4.2: Pin mapping between ADXL355 and Raspberry Pi GPIO. Only difference for the two ADXL sensors is the Chip Select Pin.

ADXL355 Pin Number	ADXL355 Function	Raspberry Pi Pin
1	Chip Select (CS)	SPI0 CS0 (Pin 24)
1	Chip Select (CS)	SPI1 CS0 (Pin 26)
2	Master Out Slave In (MOSI)	SPI0 MOSI (Pin 19)
3	Master In Slave Out (MISO)	SPI0 MISO (Pin 21)
4	Serial Clock (SCLK)	SPI0 SCLK (Pin 23)
5	Digital Ground (GND)	GND (Pin 6)
6	Digital Power (VDD)	3.3 V (Pin 1)
7	Interrupt 1	Not Connected
8	Not Connected	Not Connected
9	Interrupt 2	Not Connected
10	Data Ready	Not Connected
11	Digital Ground (GND)	GND (Pin 25)
12	Digital Power (VDD)	3.3 V (Pin 17)

4.7 Data Acquisition

Data acquisition is automated and synchronized using PTP as described in Section 4.5 to ensure well labelled data. The data acquisition basically consists of a sequence of actions that repeatedly pull on an object and record the slip data. One pass through the sequence is called a run, and every run produces one recording. Figure 4.8 shows the sequence.

A run starts with the gripper open and the carriage moved against a limit switch, so that the motor knows its zero position, and from there to the start position. The gripper then closes on the object until the target grip force is reached and holds that position. Before any pulling starts, the pulling force and the marker position are read once and stored as the baseline of the run, and the test speed is set. This baseline is what later tells apart the object sitting still from the object having moved.

The recording is started, and the motor pulls the object downwards in steps. After every step the script checks whether the object has slipped or whether the carriage has reached its lower limit switch. As long as neither is the case, the next step follows. Once one of them occurs, the recording is stopped, the gripper opens, and the GelSight is recalibrated on the now unloaded gel before the next run begins. Recalibrating between runs matters because the gel keeps a small amount of deformation after being loaded, which would otherwise carry over into the following recording.

Between runs, three settings are varied: the surface of the test object (Section 4.1), the grip force, and the speed of the pull. Every combination is recorded several times, so that the data covers both surfaces that slip smoothly and surfaces that slip in jerks, at different loads and speeds. During a run the speed is randomized between every pull action.

Each recording holds all sensor streams with the timestamps described in Section 4.5. They are afterwards converted into a single file per run, in which the GelSight frame is the reference: every frame is stored together with the roughly 160 acceleration samples that fall into its 40 ms period, the force reading, and the object velocity derived from the ArUco markers. That velocity is the label of the frame. The camera image and the marker position themselves are not passed to any network, so the ground truth cannot leak into the input. In total, 290 runs were recorded, amounting to about three hours of recording time.

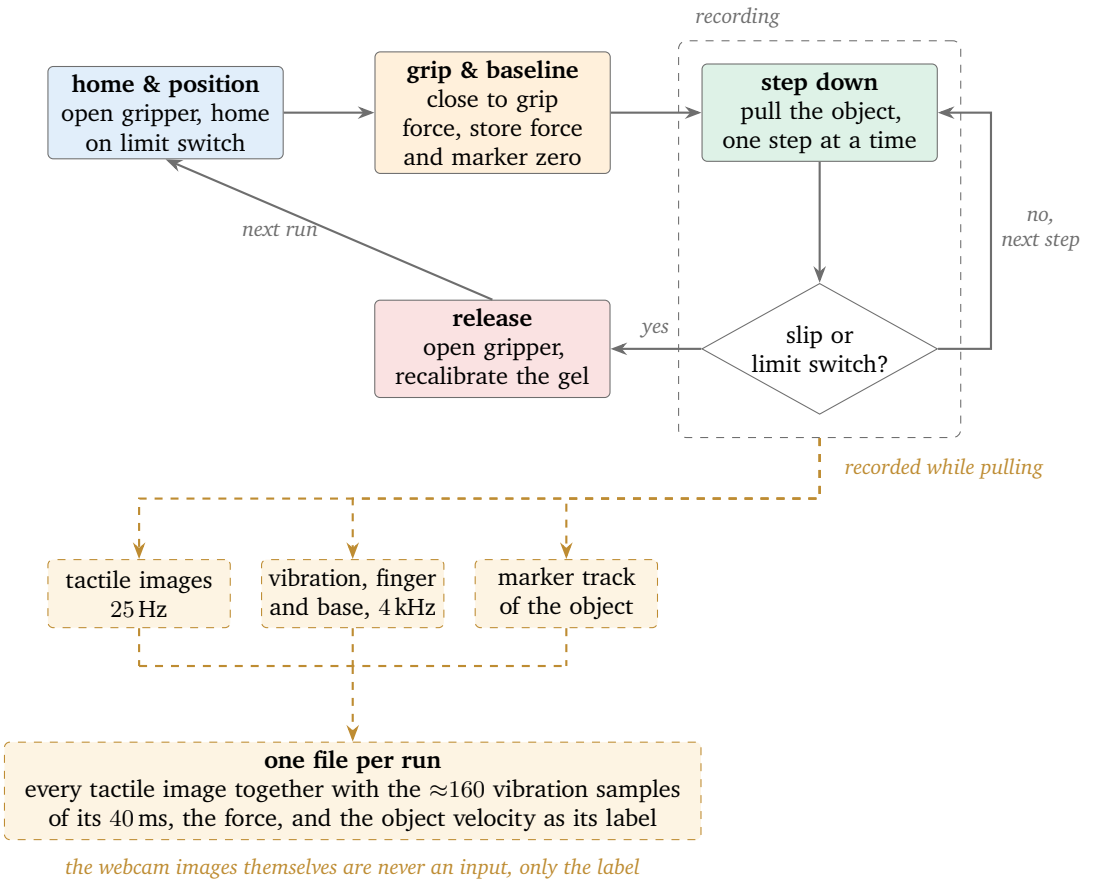


Figure 4.8: One acquisition run and the data it produces. The bench homes itself, grips the object and takes its baselines before any load is applied. Only the pull is recorded, and the run ends either when the object slips or when the carriage reaches its lower limit switch. The four recorded streams are afterwards merged into one file per run, with the tactile image as the reference.

5 Methodology

Estimating continuous slip velocity from multimodal robotic sensing requires a methodology that combines synchronized tactile, acceleration, and visual measurements with a physics-informed experimental design and neural architectures capable of modelling their complementary temporal information. The methodology is split into five parts:

1. Task formulation: defining the slip velocity target
2. Data and Representations: how is the Input structured and transformed to expose task-relevant temporal and cross-modal structure.
3. Network preliminaries: the building blocks, the shared output formulation, and the branch encoders that every architecture is assembled from.
4. Base network architectures: why and how different modelling and fusion paradigms can be used to estimate slip velocity.
5. Themed architectures: variants that each target one specific shortcoming of the base networks.

The test bench itself, and how the multimodal data are acquired, synchronized and labelled, is described in Chapter 4.

5.1 Task formulation

How to estimate the slip velocity of a gripped object from visuotactile and acceleration data? A fundamental problem with that problem statement is, that neither slip velocity, nor visuotactile and acceleration data are entirely objective measurements: the image shows a deforming gel rather than the contact itself, the accelerometer measures the vibration of the finger structure rather than the motion at the interface, and slip velocity

is a property of a contact that, during incipient slip, is neither fully sliding nor fully stuck. This section therefore fixes the input and output pair that every architecture in this chapter shares (Section 5.1) and the ground truth slip velocity against which they are all trained and evaluated (Section 5.1).

Data Input and Output

The input consists of the visuotactile image data from the GelSight, and the (x, y, z) acceleration data. Since the acceleration data is recorded at $4000\,\text{Hz}$ and the GelSight records at $25\,\text{Hz}$, one frame is assigned 160 acceleration measurements. The combination of the two is one sample, which is labelled with its *slip velocity*.

Ground Truth Slip Velocity

In this thesis, slip occurred when:

1. The object was moving relative to the gripper
2. The vibration amplitude exceeded that of normal operation by 10%

The *slip velocity* is the speed of the movement of the object relative to the gripper, as determined by an external camera.

As discussed in Chapter 2, trying to find a binary classification of what even constitutes slip is an exercise in futility. Most related work defined slip as the relative movement of the object to the gripper. Some [29] went further and defined it as the relative movement of object and the gel. The former often estimates the start too early, as the compliant gel shears elastically under a growing tangential load, displacing the object relative to the gripper before any sliding occurs at the contact interface. The latter has a tendency to predict slip too late: during incipient slip, sliding is confined to a small annulus at the contact periphery while the central stick region keeps the gel moving with the object, so measurable relative motion between object and gel only emerges once the stick region collapses and gross slip is already underway.

As incipient slip is a gradual process, there is no “correct” way to define slip. In this thesis slip occurs when the object moves more than 2mm during a pull event. The 2mm are the maximum downward elasticity of the GelSights gel. This has a tendency to estimate slip too early, though this is a trade-off that is made to be able to properly cover the incipient

slip. Data analysis shows that at the speeds used in this thesis, the difference between slip relative to the gripper occurring, and slip relative to the gel occurring is about 100ms, or about 2.5 frames.

5.2 Data Representation

A central part of this thesis is researching whether utilizing different data representations is beneficial.

Research indicates that transforming raw sensor data into physics-informed modalities, like spatial entropy of marker displacements or FFT for acceleration data, consistently yields better accuracy [43, 58, 6].

The following will list the different representations used and their working principles.

5.2.1 Tactile Representations

Three tactile representations were used in this thesis: The raw visuotactile image, the marker flow, and the depth representation.

Raw Tactile Images

Raw GelSight RGB frame track, read directly from the recorded HDF5 dataset as can be seen in Figure 5.1. The main consideration is the resolution of the image. The native resolution of the GelSight is 3264 x 2448, which is too high for virtually all use cases. As such, most applications downscale this to 320x240 as recommended by the manufacturer.

Instead of only inputting a single frame, consecutive frames $t-1$ and t are stacked on the channel axis, so the first convolution directly sees displacement between frames.

The idea is that the ability to see motion is more immediate in the input data, instead of having to learn that behavior.

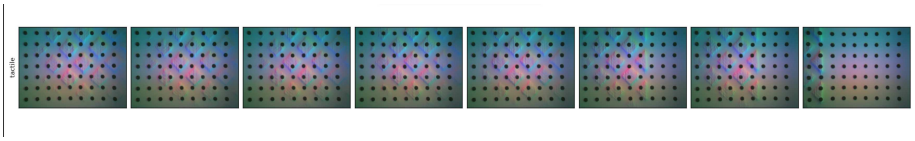


Figure 5.1: Example GelSight frames used as tactile input.

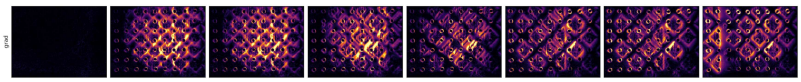


Figure 5.2: Depth reconstruction from the GelSight gradient signal.

Depth

Per-frame depth maps, as can be seen in Figure 5.2 are reconstructed from the GelSight images using the gradient-integration network provided by the manufacturer, yielding a height map of the gel surface. The result is a 320x240 matrix in which each entry specifies the depth of the gel at that pixel. The network requires a calibration baseline of 40 frames to determine the reference state of the undisturbed gel. In this thesis, a pre-recorded sequence of the unloaded gel serves as this baseline. Recording a fresh baseline immediately before each grasp would marginally improve accuracy, but was omitted to reduce the setup time per trial.

Marker Flow

Tracks the per-frame Lucas-Kanade marker displacement. Lucas-Kanade builds on the brightness constancy assumption, which states that a moving point keeps its intensity, giving the optical flow constraint

$$I_x u + I_y v + I_t = 0, \tag{5.1}$$

where I_x, I_y, I_t are the spatial and temporal image derivatives and (u, v) is the sought displacement. Since Equation 5.1 is one equation with the two unknown displacements u and v, the constraint is collected over a small window around each marker and solved by least squares:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \left(A^\top A\right)^{-1} A^\top \mathbf{b}, \quad A = \begin{pmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ \vdots & \vdots \end{pmatrix}, \; \mathbf{b} = -\begin{pmatrix} I_t(\mathbf{p}_1) \\ \vdots \end{pmatrix}, \tag{5.2}$$

with $\mathbf{p}_i$ the pixels inside the window [72]. Instead of a dense field, the flow is thus only computed at the marker positions. This is a less data-heavy but less accurate method than dense flow. Per frame pair it takes on average 0.15 ms to calculate all 63 marker displacements, making it approximately 73 times quicker than dense flow.

5.2.2 Vibration Representations

Slip is at its core a vibratory process, and the accelerometer is the only stream that resolves it within a single frame interval: at $4000\,\text{Hz}$ it returns roughly 160 measurements in the time the GelSight produces one image. Where a visuotactile representation requires a convolutional pass over a 240×320 image, a window of vibration data can be reduced with a Fourier transform, or with plain sums and averages, at a fraction of the compute and a fraction of the size.

Base Acceleration

Raw 3-axis ADXL355 acceleration track, read directly from the recorded dataset. For each frame the timestamp of the current and next frame is taken, and all acceleration measurements in that time frame are bundled. Frequencies below $50\,\text{Hz}$ are removed with a high-pass filter, as those frequencies contain signals from gravity and movement of the gripper.

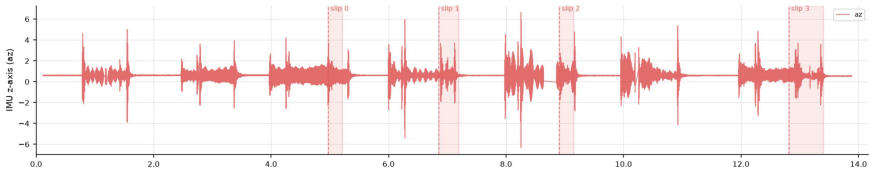


Figure 5.3: Raw acceleration signal used as a vibration sensor input representation, with the slip events marked.

Acceleration Short-Time Fourier Transform

Short-Time Fourier Transform (STFT) is a transform used to determine the frequencies present in set window of a given signal. In this thesis it is used to deconstruct the vibration

signal into its frequencies components. As the vibration signal at the time of recording is digital, and as such discretized, the time-discrete version of STFT is used:

$$\textbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-i\omega n}$$

For training, the frequency spectrum is divided into 128 bins, starting at $50\,\text{Hz}$ and going up to $2000\,\text{Hz}$, in steps of about $15\,\text{Hz}$.

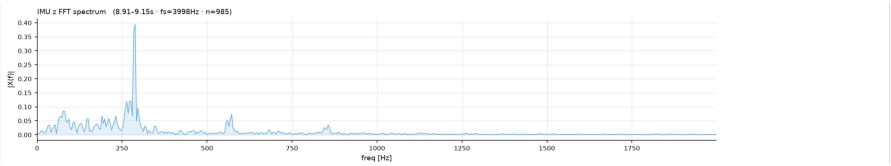


Figure 5.4: Frequency-domain representation of the vibration signal during a slip event.

Acceleration Summary

For a windowed acceleration axis signal $x = (x_1, \ldots, x_T)$ of length T, we extract four scalar descriptors per channel, capturing amplitude, oscillation rate, and spectral shape. Throughout, $\bar{x}$ is the window mean, $\tilde{x}_t = x_t - \bar{x}$ the mean-centered signal, and ε a small constant guarding against division by zero.

Root-mean-square (RMS) amplitude, the window's overall energy:

$$\text{rms}(x) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} x_t^2 + \varepsilon}$$

Zero-crossing rate, the fraction of adjacent samples that straddle zero and thus a proxy for oscillation rate, with $\text{sgn}(\cdot)$ the sign and $\mathbb{1}[\cdot]$ the indicator (one when its argument holds, zero otherwise):

$$\text{zcr}(x) = \frac{1}{T-1}\sum_{t=2}^{T} \mathbb{1}[\text{sgn}(\tilde{x}_t) \neq \text{sgn}(\tilde{x}_{t-1})]$$

Spectral centroid, the amplitude-weighted mean frequency (the spectrum's center of mass). Let $X_f = |\text{DFT}(\tilde{x})_f|$ be the magnitude spectrum for bins $f = 0, \ldots, F-1$ ($F = \lfloor T/2 \rfloor + 1$), with normalized frequency $\nu_f = f/(F-1) \in [0, 1]$:

$$\text{centroid}(x) = \frac{\sum_f X_f \, \nu_f}{\sum_f X_f + \varepsilon}$$

Peak frequency, the mode of that spectrum at $f^\star = \arg\max_f X_f$:

$$\text{peak}(x) = \nu_{f^\star}$$

Applied independently to each axis, this yields a $(3, 4)$ feature matrix (RMS, ZCR, spectral centroid, and peak frequency per axis), flattened to a 12-dimensional summary of the window's vibration content.

5.3 Network Preliminaries

Slip-velocity estimation is a multimodal, multi-rate learning problem. Both of these present a challenge when designing architectures. How to correlate signals that have fundamentally different representations and encode different information?

5.3.1 Architecture Overview

In total eight architectures were designed. Three "base" variants, general architectures that were constructed using trial and error using best practices. And five themed variants, that each try to solve a specific problem. Some of the themed architectures are inspired by research which is named in their description. The three base networks of Section 5.4 are:

`Fusion` uses a standard MLP architecture to perform late fusion by encoding each modality independently in its own branch before combining the resulting embeddings in a shared prediction trunk.

`Transformer` significantly shortens the time until tokenization with a larger cross attention transformer trunk. In addition, it extends the transformer idea to asynchronous sensor streams by tokenizing each modality at its native rate and encoding the real age of every token relative to the prediction time.

Hybrid combines the idea of Fusion, which also uses branches, but transforms the output of each branch into a shared token space and fuses them with a transformer.

The five themed architectures of Section 5.5 are: **RateCore** separates slow visuo-tactile sensor stream from a fast accelerometer-based prediction pathway, addressing the mismatch between the GelSight frame rate and the temporal scale of slip spike events.

Strip tests how much of the performance of the multimodal models remains when the problem is reduced to a minimal causal time-series model using raw vibration and compact marker-motion statistics.

Twin investigates whether cross-modal self-supervised pretraining can use synchronized but weakly labelled visuo-tactile and inertial recordings more effectively before supervised fine-tuning.

Event changes the output parameterization to model the brief incipient slip phase separately from the subsequent slip-velocity plateau, with the intention of being better able to predict slip before it happens.

Evidence replaces fixed learned fusion with a weighted combination of modality-specific predictions, to increase the robustness against the case where a sensor stream is present but unreliable.

5.3.2 Time Alignment

The acceleration sensor and the GelSight run at different timing frequencies, so they have to be aligned, both during training and during inference. During training the timestamp of the first GelSight image is used to find the matching vibration signal measurement. During inference, the timestamp of the slip-velocity label is used instead.

5.3.3 Time encoding

This is only relevant for the token based architectures. The method implemented in this thesis is based on Time2Vec by Kazemi et al. [56]. The window anchor is always the first measurement in a window. For each token, the temporal distance d_t to the window anchor is determined.

$$\omega_k = \frac{2\pi}{T_k}, \qquad T_k = T_{\text{min}} \left(\frac{T_{\text{max}}}{T_{\text{min}}} \right)^{\frac{k-1}{n-1}}, \qquad k = 1, \ldots, n \tag{5.3}$$

$$\phi(dt) = W \begin{bmatrix} \sin(\omega_1\, dt) \\ \vdots \\ \sin(\omega_n\, dt) \\ \cos(\omega_1\, dt) \\ \vdots \\ \cos(\omega_n\, dt) \\ dt \,/\, T_{\text{max}} \end{bmatrix} + b, \qquad W \in \mathbb{R}^{d_{\text{model}} \times (2n+1)} \tag{5.4}$$

Where $\phi(dt)$ is the time embedding, `min_period`/`max_period` are $T_{\text{min}} = 0.002, T_{\text{max}} = 2.0$, $n = 4$ is the time frequency count, and W, b are the learned weight and bias.

5.3.4 Double Head architecture

This thesis adopts a "two-head" formulation rather than a single "velocity-only" regressor, motivated by the zero-biased distribution of slip velocity: the overwhelming majority of samples correspond to static contact, while the remainder follow a continuous and comparatively noisy distribution over slip magnitude. A single regression head trained end-to-end on this mixture is forced to trade off resolution between the dominant near-zero mode and the sparser slipping regime, systematically biasing magnitude estimates towards zero and degrading sensitivity precisely where detection matters most. The target is decomposed into a discrete occurrence variable and a conditional magnitude variable, following the classical hurdle-model formulation for zero-inflated data.

A classification head predicts the gate probability $p_\theta(\text{slip} \mid x)$ via a logit trained with confidence-weighted binary cross-entropy against a consensus pseudo-label, while a regression head predicts $\mathbb{E}[\log(1+v) \mid \text{slip}, x]$ via a Huber loss masked to samples that the consensus labels as confidently slipping. Thus, the magnitude head is not trained on ambiguous or near-zero targets.

Concretely, consider a training batch of B samples indexed by i. Each sample carries a log-velocity target $y_i = \log(1 + v_i)$, a binary consensus slip label $g_i \in \{0, 1\}$, a label confidence $c_i \in [0, 1]$, and a class weight s_i. The label g_i comes from the offline slip segmentation of the marker field, and c_i is that segmentation's own confidence wherever it agrees with a second, independent label obtained by thresholding the marker-derived

speed (at $v_{\text{hi}} = 5\,\text{px/s}$, or at the midpoint of the hysteresis band $[v_{\text{lo}}, v_{\text{hi}}] = [2, 5]\,\text{px/s}$ when the speed lies inside it). Where the two sources disagree, the confidence is capped at $c_{\text{amb}} = 0.3$. The class weight is $s_i = 1.5$ for slipping and $s_i = 1$ for static samples. The network emits the gate logit z_i and the log-velocity estimate $\hat{y}_i$. Regression residuals are scored with the Huber loss $\ell_i = H_\delta(\hat{y}_i - y_i)$, where $H_\delta(u) = \frac{1}{2}u^2$ for $|u| \le \delta$ and $\delta\left(|u| - \frac{1}{2}\delta\right)$ otherwise, with $\delta = 0.3$ measured in log space, so that residuals beyond roughly a $35\,\%$ relative velocity error are penalized linearly rather than quadratically and single mislabelled frames cannot dominate the gradient. In addition this encourages guessing velocity spikes. In early runs all architectures ignored the spikes, as guessing them wrong incurred a too large loss penalty.

The training objective combines three terms,

$$\begin{aligned}\mathcal{L} \;&= \frac{\sum_i s_i\, g_i c_i\, \ell_i}{\max(\sum_i g_i c_i,\; 1)} && \text{(magnitude, confident slip only)}\\ &+ \lambda_{\text{floor}}\, \frac{1}{B} \sum_i \ell_i && \text{(floor, all samples)}\\ &+ \lambda_{\text{gate}}\, \frac{\sum_i c_i\, [-g_i \log \sigma(z_i) - (1 - g_i) \log(1 - \sigma(z_i))]}{\max(\sum_i c_i,\; 1)} && \text{(gate, all samples)}\end{aligned} \tag{5.5}$$

with σ the sigmoid, $\lambda_{\text{floor}} = 0.05$, $\lambda_{\text{gate}} = 0.3$, and all sums running over $i = 1, \ldots, B$.

The magnitude term averages ℓ_i over confidently slipping samples only, using the product $g_i c_i$ as a soft mask. The mask vanishes for every static sample and shrinks to at most c_{amb} for samples the two label sources disagree on, so that a slip-labelled sample with a near-zero target cannot dominate the regression. The clamped denominator keeps the term finite for batches that contain no confident slip at all. The gate is trained on the same confidence weights, but over all samples, so ambiguous frames contribute at reduced weight instead of being dropped. The floor term applies a small regression loss to every sample. This ensures that the velocity head still receives a training signal when the slip classifier incorrectly predicts that a slipping sample is static, preventing unreliable velocity estimates.

Combining the three terms additively allows the classification and regression heads to be optimized jointly while retaining separate supervision for slip occurrence and slip magnitude. The sum is deliberate: multiplying the two heads into a single expected velocity $p_\theta(\text{slip} \mid x) \cdot \hat{v}$ would reduce the gate to a scaling factor on the regression output and reintroduce the zero-inflation pathology the hurdle formulation is meant to avoid. All

architectures share this two-headed design and are trained with the same loss and the same weights, so that benchmark differences reflect the architectures rather than their supervision.

5.3.5 Branch Encoders

The Branch encoder is a fundemental building block for most of the architectures that will be presented in the following. Each sensor modality, tactile, depth, marker, and vibration, is turned into a fixed-size embedding by a dedicated branch encoder before any fusion takes place. All branch encoders follow the same overall pattern: modality-specific input is passed through a small stack of convolutional or recurrent layers suited to that modality's structure, then reduced to a single embedding vector via pooling and a linear projection. Figure 5.5 gives an overview of this structure, and the remainder of this section describes each branch encoder in turn.

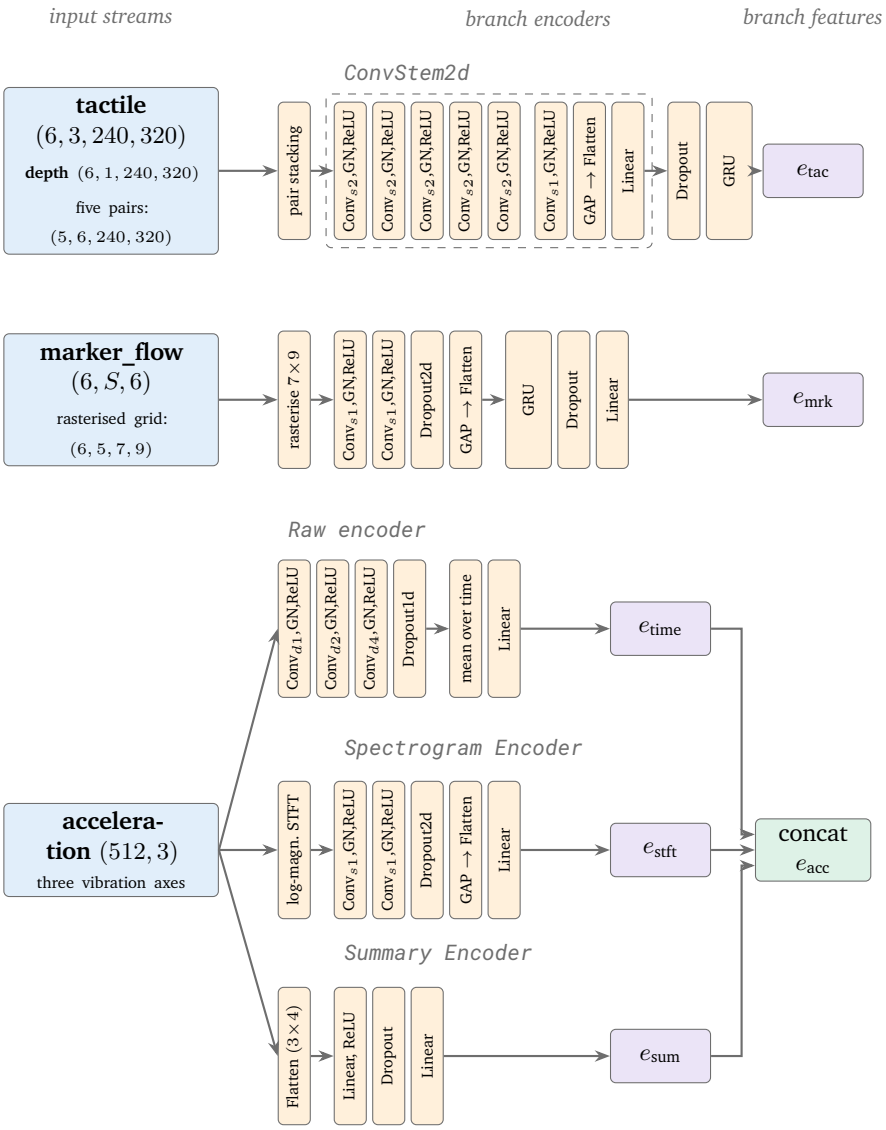


Figure 5.5: Internal structure of the branch encoders. Each row runs from an input tensor through its layer stack to a branch embedding. `ConvStem2d` is shared across the five frame pairs, and the per-frame CNN of the marker branch is shared across the six frames. GN denotes GroupNorm, GAP global average pooling.

ConvStem2D

`ConvStem2d` is a resolution-agnostic convolutional operation built from a stack of convolutions to achieve a given output size. ConvStem2D repeatedly halves the input's spatial size with stride-2 convolutions, followed by a `ReLU` and normalization, until the grid is small enough, with the exact number of stages computed from the given input resolution. This combination of `Conv2D` and `ReLU` serves to achieve both feature extraction and spatial reduction. This stack is followed by one stride-1 convolution, global average pooling, and a linear projection that reduce the resulting feature map to a fixed-size embedding vector.

Gated Recurrent Unit

Introduced by Cho et al. [15], a Gated Recurrent Unit (GRU) is used to selectively remember past inputs. As such it serves a similar function as the Long Short Term Memory (LSTM) unit, with the main difference being that GRU combines short and long term memory into one state. This thesis uses the standard PyTorch implementation:

$$
\begin{aligned}
r_t &= \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \\
z_t &= \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \\
n_t &= \tanh(W_{in}x_t + b_{in} + r_t \odot (W_{hn}h_{t-1} + b_{hn})) \\
h_t &= (1 - z_t) \odot n_t + z_t \odot h_{t-1}
\end{aligned}
$$

Here x_t is the input at step t (in the tactile branch, the ConvStem2d embedding of the t-th frame pair) and h_{t-1} the previous hidden state, with $h_0 = 0$. The sigmoid σ produces the reset gate r_t, controlling how much of h_{t-1} enters the tanh-squashed candidate state n_t, and the update gate z_t, controlling how much of h_{t-1} survives into the new state h_t. The matrices W_{i*} and W_{h*} project input and previous state with biases b_{i*}, b_{h*}, where $* \in r, z, n$ names the target quantity, and $\odot$ is element-wise multiplication.

The thesis uses a derivative function `ConvGRUCell` that replaces every matrix multiplication with a 3x3 convolution, with the intention of better encoding spatial features.

Tactile and Depth Encoder

Both the tactile and depth branch take 240×320 images as input. Every input is an image pair concatenated along the channel axis. For the tactile RGB images each of the six images with dimension $(3, 240, 320)$ is paired with its subsequent image into a $(6, 240, 320)$ tensor. The resulting five image pairs are stacked into a final $(5, 6, 240, 320)$ tensor for the tactile encoder and a $(5, 2, 240, 320)$ tensor for the depth encoder. The idea is to input a large enough window to find not only correlations between consecutive pairs, but also across the sequence of pairs. The stack of frame pairs is then passed through a `ConvStem2d`, which encodes every pair independently.

Marker Encoder

The marker branch represents GelSight dot motion as a spatially aligned displacement field on a grid. Each grid cell corresponds to one marker location and contains the marker's horizontal and vertical displacement and velocity, $[dx, dy, vx, vy]$. A fifth channel indicates whether a valid marker measurement is present at that location, resulting in a $(5 \times 7 \times 9)$ input grid for each time step.

This input is first passed through a per-frame CNN for spatial encoding, consisting of two 3×3 convolutional layers with GroupNorm and ReLU activations. Unlike the tactile branch, no downsampling is required, since the marker grid is already small. A linear projection layer produces the marker encoder embedding.

Raw Vibration Signal The raw vibration time-series branch takes a three-channel sequence, corresponding to the measured vibration axes. The sequence is passed through a stack of three one-dimensional convolutional layers for temporal encoding. Each convolution uses a kernel size k of 5, with increasing dilation d factors of 1, 2, and 4, which means that the Conv1d operation looks at k adjacent samples that are d samples apart, allowing the encoder to capture motion patterns over progressively wider time ranges. Each layer is followed by GroupNorm and a ReLU activation. Dropout is then applied to the temporal feature maps, after which global average pooling aggregates the sequence into a single feature vector. A linear projection layer produces the time-series vibration embedding.

Spectrogram The vibration spectrogram branch represents the same three vibration channels in the time-frequency domain as calculated by a Short-Time Fourier Transform (STFT). This input is passed through a small CNN consisting of two 3×3 convolutional layers, each followed by GroupNorm and a ReLU activation. The first convolution maps the three input channels to the configured feature width, while the second doubles that width. Dropout is applied to the resulting feature maps, and adaptive global average pooling reduces the spatial time-frequency representation to a single vector. A final linear layer projects this vector into the spectrogram-branch embedding.

Vibration Summary The vibration summary branch uses a compact window-level representation (explained in Section 5.2.2) with shape (3×4), which is flattened into a 12-dimensional vector. This vector is processed by a small MLP consisting of a linear layer, ReLU activation, dropout, and a second linear projection layer. This produces a learned embedding of the summary statistics without explicit temporal recurrence or convolution.

The outputs of the time-series, spectrogram, and summary branches are concatenated to create the full vibration representation. If one vibration representation is unavailable, its learned absent vector replaces only that sub-branch while the remaining vibration encoders continue to contribute features.

Every architecture in Sections 5.4 and 5.5 builds on the branch encoders described above, but each one adapts them to its own fusion mechanism.

5.4 Base Network Architectures

This section presents the three base architectures: `Fusion`, `Transformer`, and `Hybrid`. All three base architectures reuse the same branch encoders (Section 5.3.5) to turn raw sensor streams into embeddings, and all three decode a single fused representation into the same gated (hurdle) slip-velocity and slip-gate heads (Section 5.3.4), so differences in performance are due to how each network fuses its inputs rather than to what it encodes or predicts.

What changes between them is the fusion point and mechanism. `Fusion` is the most classical: each branch is encoded to completion, and only the resulting embeddings are concatenated and passed through a shared MLP, so it's a classic example of the late fusion architecture introduced in Section 3.4.1. `Transformer` instead tokenizes every stream

at its own native rate and fuses at every layer through self-attention, with a Fourier embedding giving each token its exact age; this lets a single prediction token attend jointly over raw, asynchronous, multi-rate sensor data. `Hybrid` keeps this token-and-attention mechanism but decouples it from any particular sensor stream: it tokenizes a configurable list of representations, and drops the continuous time embedding in favor of a fixed learned position per token in addition to passing the inputs through more layers before they are fully tokenized.

5.4.1 Late-Fusion Network (`Fusion`)

The Late-Fusion network is the most classic of the neural networks, using CNN layers and other modality-specific encoders to encode the different data types, and then using an MLP to evaluate the concatenated representations. The name is derived from the fact that most of the processing occurs before the different data streams are concatenated. This has the advantage of allowing the network to learn strong internal representations of each input, with the risk of over-prioritizing one representation.

Fusion Head

The late-fusion network uses the branch encoder as described above with the following changes:

The tactile branch is passed into a number of GRU layers (see Section 5.3.5). The GRU layers are intended for learning longer-range temporal dependencies in the lower-dimensional CNN output, and its final hidden state forms the branch output. While inputting images directly into a GRU would also serve to relate features temporally, using a flattened ($240 \times 320 \times 3$) input would require roughly 266 million parameters in the input-to-hidden weights of the first layer alone, at a hidden size of 384, and would discard the translation equivariance and weight sharing of the convolutional stem.

For the marker branch, the resulting grid features are globally pooled into a feature vector for each time step, which is then passed through a GRU for temporal encoding. Finally, dropout and a linear projection layer produce the marker-branch embedding used by the fusion network.

Fusion Tail

After each branch encoder has produced its embedding, a branch dropout layer drops each modality with a chance of 15% during training. This is intended to prevent the neural network from over-relying on a single modality. If a modality is dropped, a learned "absent" vector is inserted in its place.

The embeddings of the different modalities are concatenated and put through a shared MLP. This MLP is followed by two separate linear heads, each producing one of the outputs described in Section 5.3.4.

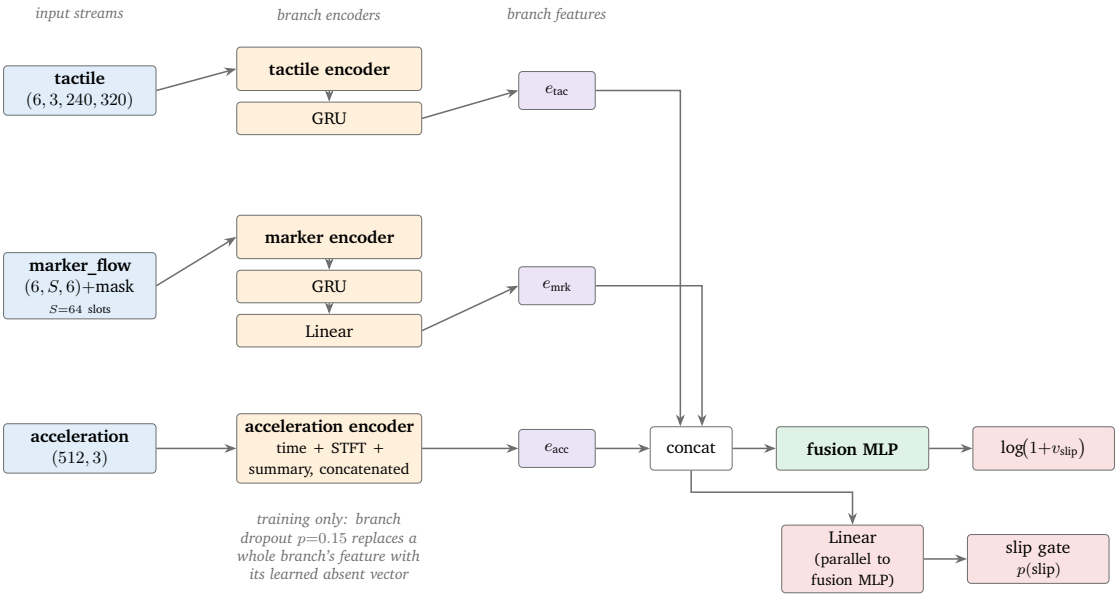


Figure 5.6: `Fusion`: late fusion of the five branch encoders (internal structure in Figure 5.5) through a shared fusion MLP into the gated (hurdle) slip-velocity and slip-gate heads (Figure 5.7). The `large` and `small` presets share the same architecture and differ only in capacity as noted in Table 5.1.

5.4.2 Continuous-Time Transformer (`Transformer`)

One of the major advantages of transformers is their ability to deal with multi-rate data. To make use of this fact, the embedding produced by each branch is passed through a

Table 5.1: `Fusion` preset comparison.

Parameter	**small**	**large**
Tactile branch		
frame_dim (ConvStem2d out)	64	192
gru_hidden	96	384
gru_layers	1	2
Vibration sensor sub-branches		
vib_time_width	32	128
vib_stft_width	16	64
vib_summary_width	16	64
Marker branch		
marker_width	32	64
marker_out	32	64
marker_hidden	32	64
Fusion trunk / head		
fusion_hidden	(128, 64)	(768, 384, 128)
head_dropout	0.1	0.2
Total		
Parameter count	$\leq 0.3\,\mathrm{M}$	$\sim$6–8 M

Fourier time embedding function as explained in Section 5.3.3. The base version has three branches: GelSight image pairs, marker flow, and acceleration data.

Transformer research is predominantly conducted in natural language processing, where the input and the target share the same token space. A tokenizer maps text onto tokens, which an embedding maps to vectors. The transformer layers refine these into new representations, and an output “de-tokenizer” maps them back onto the same token vocabulary. De-tokenization is approximately the inverse of tokenization, making the whole pipeline symmetric. This symmetry does not hold for the problem that this thesis addresses. The inputs and outputs are fundamentally different, and as such the tokenizer and de-tokenizer are fundamentally different as well.

Transformer Head

Each stream is tokenized at its own rate, and every token is built only from measurements that are older than the prediction time. A token embedding is the sum of three parts: the output of the stream's tokenizer, a learned embedding of the stream it came from, and the Fourier embedding of its age. The encoders used inside the tokenizers are the ones described in Section 5.3.5 (Figure 5.5).

The tactile stream is served as six frames. Consecutive frames are stacked into pairs of six channels and passed through the `ConvStem2D` of Section 5.3.5, followed by a linear projection. This yields five tokens, one per frame pair, and each token is encoded with the time of its newer frame.

The marker flow is served per marker and for each of the six frames, the marker encoder embeds every tracked marker and pools the set with a masked mean and maximum, and a linear layer projects the result. This yields six tokens, one per frame, each carrying the time of its frame.

For the vibration data, a total of 512 measurements are input into the network, and divided into 16 tokens with each token encoding 32 measurements. Each token gets encoded with the time of the newest measurement contained in it.

The reconstructed depth track can optionally be added as a fourth stream. It is tokenized exactly like the tactile stream, with one channel per frame instead of three, and it reuses the tactile frame times. It is disabled by default and is evaluated as an ablation in Chapter 6.

Transformer Body

The body is a transformer encoder that operates on the concatenated tokens from the sensors together with a learned prediction (PRED) token of age zero. Because every stream's tokens and the PRED token share a single sequence, standard self-attention already lets the PRED token attend to all 27 tokens at every layer; no separate cross-attention mechanism is needed. 4 layers with 4 heads each are used, with GELU activations inside pre-norm residual blocks. After the final layer, the PRED token's output is taken as the fused representation and passed through a LayerNorm before being decoded by the two prediction heads (Figure 5.7).

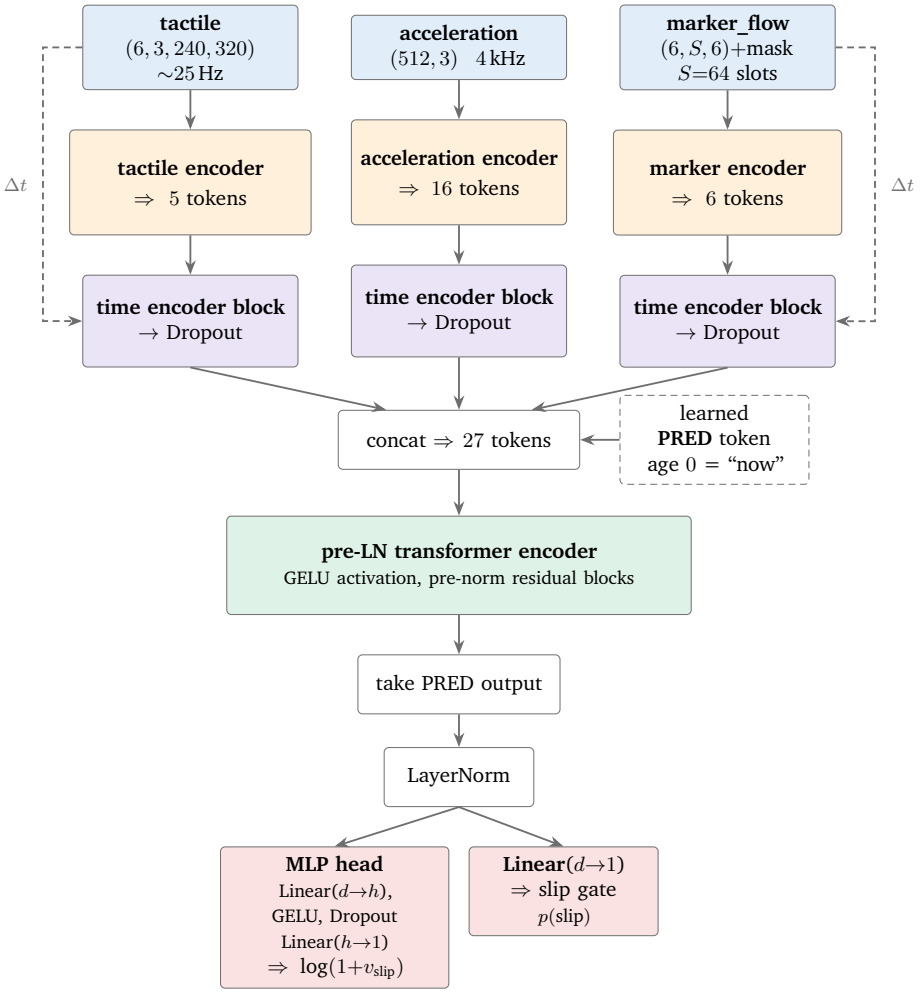


Figure 5.7: `Transformer`: a continuous-time token-fusion transformer. Each sensor stream is cut into causal tokens at its native rate by its tokenizer (internal structure in Figure 5.5); every token carries its real age Δt through a Fourier time embedding, so no resampling or uniform-spacing assumption is needed. A learned prediction token with age zero attends over all 27 tokens and is decoded into the slip velocity and the slip gate $p(\text{slip})$.

5.4.3 Hybrid Representation-Fusion Network (`Hybrid`)

The Hybrid network is also token based, but instead of one tokenizer per raw sensor stream, it takes a list of representations as input. Each representation gets its own tokenizer, and adding or removing one is a single line in the config. This makes the Hybrid network the fastest for testing different representation combinations, it can mix learned representations, like the tactile images, with engineered ones, like the acceleration FFT (Section 5.2.2) or the depth map (Section 5.2.1).

Hybrid Head

The architecture supports up to six representations: tactile frame pairs, depth, marker flow, and three views of the acceleration data (raw, FFT, and summary statistics). Throughout Chapter 6, only three of these, tactile frame pairs, marker flow, and the raw acceleration signal, are used by default; depth and the two remaining acceleration views are added only for the ablations of Section 6.5. Each representation is turned into tokens by its own encoder, reusing the encoders of Section 5.3.5 (Figure 5.5). The main differences are extra layers that are selectively appended or prepended. The tactile encoder is followed by five `Conv1d` layers and yields 5 tokens, one per frame pair, and the depth encoder is built the same way, also yielding 5 tokens. The marker flow is first rasterized into a grid before the encoder pools all markers of a frame into a single token. The acceleration views yield 8 tokens for the raw signal, and 1 token each for the FFT and the summary statistics. Here the raw signal is first input into a `Conv1d` layer stack before being processed by the encoder. In total this gives 21 tokens.

As in the fusion network, a whole representation can be dropped during training with a chance of 15%, so the network learns not to rely on any single one. If dropped, a learned ”absent” token takes its place. This also acts as a fallback if a representation is missing at inference time, for example when no depth reconstruction is available for a sample.

Hybrid Body

The Hybrid network reuses the PRED token from the Continuous-Time Transformer (Figure 5.7), but here it only gets a learned position, not a Fourier time embedding. This means the network assumes a fixed number of tokens per representation instead of a continuous timeline. The PRED token and the 21 representation tokens are passed through

a pre-norm transformer encoder ($d = 128$, 4 layers, 4 heads), and the PRED token's output is decoded by the same two prediction heads as the other networks (Section 5.3.4).

Unlike the fusion and transformer networks, the Hybrid network only has one size preset. This is because it is meant for ablations, where changing the network size between runs would make the comparisons harder to interpret.

Future Slip Prediction

One thing that this thesis also attempts to test was if it's possible to predict future slip. The architecture used for this was the `Hybrid` architecture because of its good performance. For the future slip prediction, the network was left unchanged, only the training method was adapted.

A prediction horizon Δ was defined, that defines how far into the future slip velocity is predicted. During training, when data with the associated timestamp t was input, the object velocity associated with the data closest to $t + \Delta$ is used as the ground truth lable. samples whose label would fall past the end of a recording, or across a gap in the acquisition, are dropped rather than approximated.

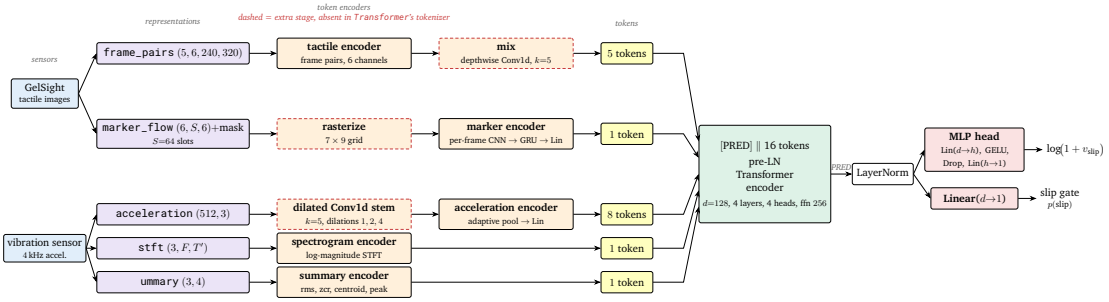


Figure 5.8: `Hybrid`: hybrid fusion of a transformer based architecture and late fusion. Each branch is mapped to its own encoder. In contrast to `Transformer`, `Hybrid` has a heavier tokenizer, extra components are marked by the dotted outline. The resulting 21 tokens are augmented with representation-type and positional embeddings and fused by a pre-LN transformer encoder. The PRED output feeds the slip velocity head and the slip gate. Unlike the PRED token of Figure 5.7, this token carries no age embedding, only the learned positional embedding shared with the other tokens.

5.5 Themed Architectures

The themed architectures presented in this section each set out to solve a specific problem that was encountered while designing the base architectures, and serve to evaluate whether integrating a solution into a base architecture could improve it in that respective way. The problems are:

- **Rate-Decoupled Core Network** (Section 5.5.1): how can we forgo having to wait for the low-frequency visuotactile image feed?
- **Stripped-Down Single-Stream Network** (Section 5.5.2): how much of the accuracy is actually owed to the learned multimodal representations, and how much of it can a handful of physically motivated scalars provide on their own?
- **Cross-Modal Self-Supervised Pretraining** (Section 5.5.3): how can the abundant unlabelled sensor recordings be exploited by pretraining encoders using unsupervised learning?
- **Parametric Decomposition of Slip Breakaway Event** (Section 5.5.4): how can the brief breakaway spike be modelled, to prevent it being averaged into the velocity plateau that follows it?
- **Evidence-Weighted Fusion Network** (Section 5.5.5): how can a sensor stream that is present but unreliable be down-weighted, when the fusion weights are fixed at training time?

Each of them keeps the branch encoders of Section 5.3.5, the two-headed output of Section 5.3.4, and the training recipe of the base networks wherever the mechanism under test does not require otherwise, so that a difference in the results can be attributed to that mechanism rather than to the encoders or the recipe.

5.5.1 Rate-Decoupled Core Network(`RateCore`)

Speed is imperative when implementing slip-control. In most cases there are only a few hundred milliseconds of headroom for both inference and reaction before a grasp fails. During the tests, only 2-3 visuo-tactile frames showed slip during a slip event. This means a substantial part of slip happens, before the first slip frame is even captured, meaning that neural networks relying on only slip data are biased toward late inference.

However, with preliminary tests showing that the vibration signals are sufficient at detection if slip is occurring, but struggling to precisely infer how fast the object was slipping, a sensible question became: ”can we use the vibration signal to produce a fast, but inaccurate estimation, that is refined by visuo tactile data as it arrives”

This is the intention behind the rate decoupled core neural network, which utilizes a fast causal core on every accelerometer tick, which is refined(conditioned) by a slow context encoder. This neural network is inspired by Feichtenhofer et al. [23], who introduced dual-clock architectures separating a slow semantic stream from a high-frequency temporal stream, Perez et al. [88], who proposed modulating deep features through affine conditioning (γ, β), and Oord et al. [86] as well as Bai, Kolter, and Koltun [7], who established causal dilated 1D convolutions for streaming temporal inference.

The fast core processes the three-axis vibration signal through a causal dilated convolutional stack. Successive layers double their dilation factor ($d = 1, 2, 4, \dots$), expanding the temporal receptive field to capture both sharp transient stick-slip spikes and broader vibration decay envelopes without downsampling.

To integrate slow visuotactile semantics into the high-rate vibration stream, intermediate feature maps $\mathbf{h}$ are scaled and shifted via Feature-wise Linear Modulation (FiLM) [88]. A linear projection maps the context embedding into channel-wise scale ($\boldsymbol{\gamma}$) and shift ($\boldsymbol{\beta}$) vectors:

$$\text{FiLM}(\mathbf{h}) = \boldsymbol{\gamma} \odot \mathbf{h} + \boldsymbol{\beta}, \tag{5.6}$$

where $\odot$ denotes element-wise multiplication along the channel dimension. $\boldsymbol{\gamma}$ acts as a scalar, weighing the importance of components of the encoding, while $\boldsymbol{\beta}$ shifts the baseline feature activation. The intention is that together they can shift and scale the acceleration data embedding to better amplify already existing vibration velocity signals.

The main constraint of this approach is the dependency of the visuotactile branch on the acceleration branch. The FiLM projection only provides conditioning, and if the causal 1D stack produces poor encoding, the visuotactile data can’t ”save it”.

In essence, this architecture bridges the multi-rate timing gap by using slow visuotactile context to continuously calibrate a causal vibration analyzer operating at the sensor’s native rate.

Nevertheless, for typical stick-slip dynamics, this rate-decoupled scheme provides the gripper controller with immediate, calibrated velocity estimates tens of milliseconds before the next camera frame arrives, which can be crucial for effective slip-control.

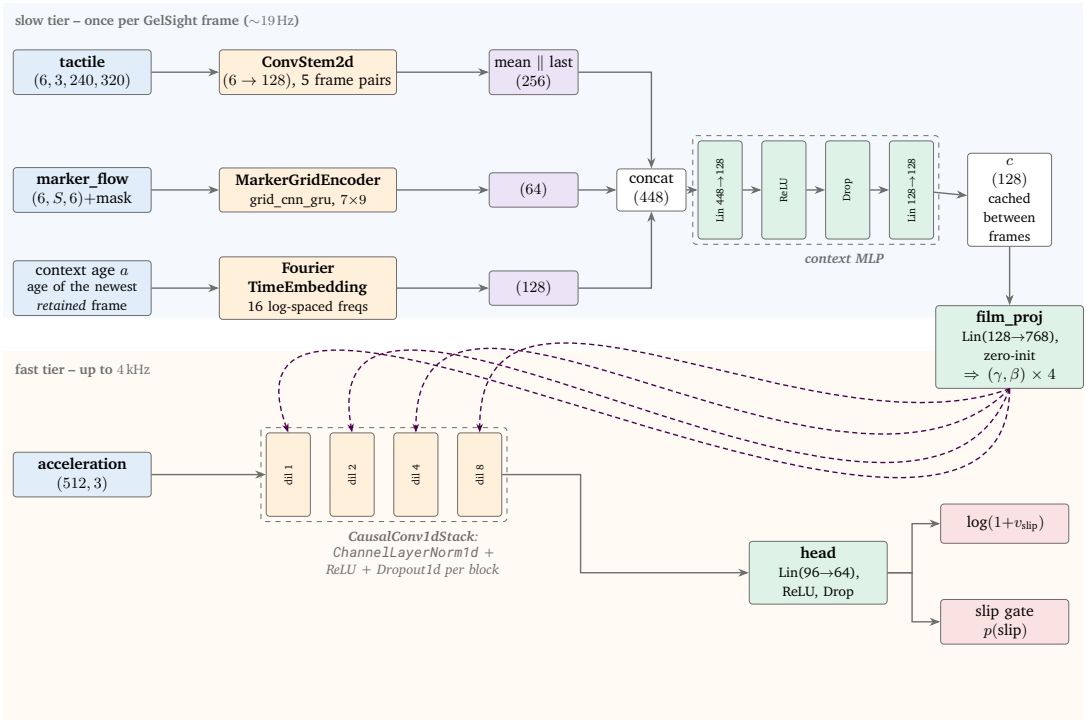


Figure 5.9: `RateCore`: a slow context tier (tactile, `marker_flow`, optionally depth), refreshed once per GelSight frame, conditions a fast causal acceleration core-tier that can be evaluated on every accelerometer tick, in practice as fast as the hardware allows.

5.5.2 Stripped-Down Single-Stream Network (`Strip`)

The philosophy behind the strip neural network was to construct a highly parameter-efficient architecture that leveraged the previous work that was done into physics informed and analytic analysis of visuotactile slip. As such only the raw vibration signal and the marker position are taken as an initial input and one large tensor is built by stacking different vibration and visuotactile marker-movement representations.

For the vibration signal, 512 raw measurements are stacked with their L1 norm, resulting in an array of shape (512, 4).

The marker position branch is more complex, being decomposed into four groups of scalars, all computed per frame from the marker slots and their validity mask, without any learned parameters.

The marker featurizer extracts a compact 20-channel representation per frame without learned parameters by fitting a first-order affine deformation model [100] to the Lucas-Kanade marker displacements [72]:

$$\mathbf{d}_i \approx \mathbf{A}(\mathbf{p}_i - \bar{\mathbf{p}}) + \mathbf{b}, \tag{5.7}$$

where $\mathbf{p}_i$ is the marker coordinate, $\bar{\mathbf{p}}$ the centroid of all valid markers, $\mathbf{b} = [b_x, b_y]^\top$ the bulk translation vector, and $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ the spatial deformation gradient. The parameters $(\mathbf{A}, \mathbf{b})$ are solved per frame in closed form via linear least-squares.

The resulting 20 features are grouped into four physical categories: The first is the **Coverage & Magnitude (6 channels)**. The correctly-detected marker fraction, the mean, maximum, and standard deviation of displacement magnitudes $\|\Delta\mathbf{p}\|$, and the mean and maximum velocity magnitudes $\|v\|$ are used as an approximation for shear force as described by Yuan et al. [119].

The **Bulk Translation (4 channels)** is calculated using the translation vector $\mathbf{b}$ for displacement and velocity, capturing rigid sliding of the contact patch as described by Dong et al. [20].

The **Kinematic Invariants (8 channels).** Following Koenderink and van Doorn's flow decomposition [63, 61], $\mathbf{A}$ is decomposed into four coordinate-independent invariants for both displacement and velocity:

$$\text{div} = a_{11} + a_{22}, \quad \text{curl} = a_{21} - a_{12}, \quad \text{shear}_n = a_{11} - a_{22}, \quad \text{shear}_s = a_{12} + a_{21}. \quad (5.8)$$

These correspond to normal indentation/expansion (div), torsional twist (curl), asymmetric stretch (shear_n), and pure shear strain (shear_s) as shown by Koenderink et al. [62].

Non-Affine Residuals (2 channels) is calculated using the root-mean-square (RMS) error of the affine fit for displacement and velocity:

$$\text{RMS}_{\text{res}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{d}_i - (\mathbf{A}(\mathbf{p}_i - \bar{\mathbf{p}}) + \mathbf{b})\|^2}. \quad (5.9)$$

Because stick-to-slip transitions initiate non-uniformly at the contact periphery while the center remains stuck, this residual peaks sharply during incipient slip as described by Chen et al. [12].

All channels are compressed via $\text{asinh}(\cdot)$ as an alternative to normalization layers. The tail of the network is comprised of a CausalConv1dStack followed by ReLu and dropout layers that finally pass into the MLP head.

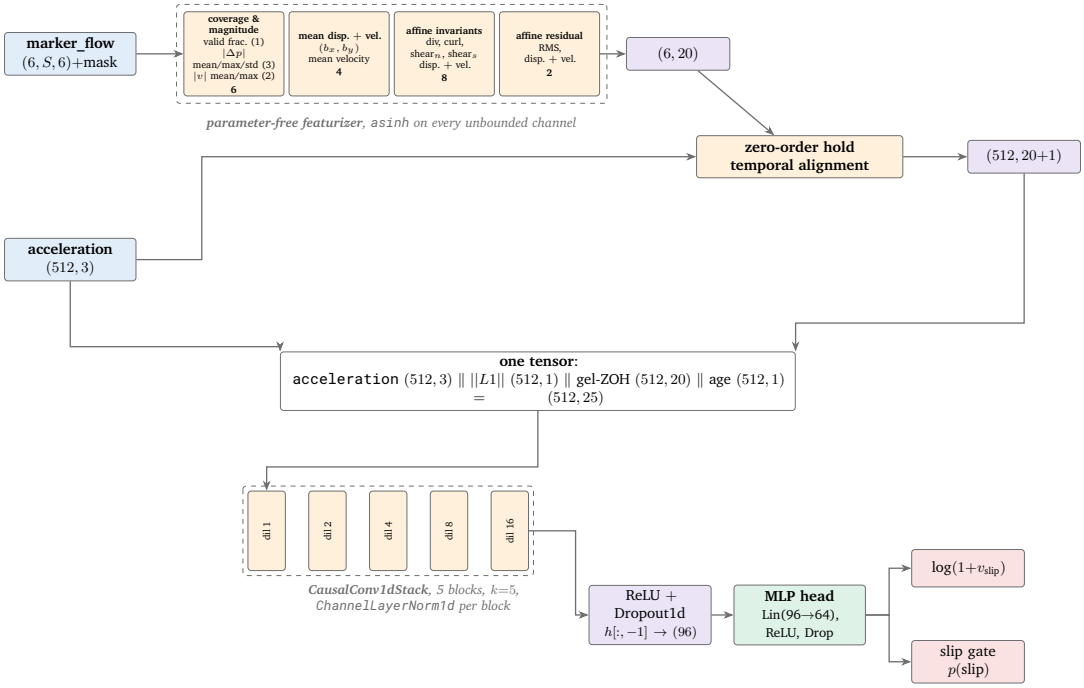


Figure 5.10: `Strip`: the marker field is reduced to $\sim$20 parameter-free physical scalars per frame (including the four affine invariants of the displacement and velocity fields), zero-order held onto the ADXL grid, and read by a single causal Conv1d stack.

5.5.3 Cross-Modal Self-Supervised Pretraining (`Twin`)

In slip detection, raw sensor data is easy to generate abundantly, while labels are much harder to generate and sparser. The idea with the `Twin` neural network is to build twin self supervised encoders, that learn the relationship between the sensor streams, and are fine-tuned using the labeled data.

While the sensor streams are already time synchronized, temporal correlation alone does not provide the semantic correspondence between modalities. The twin encoders learn how features across different streams, like optical marker velocity and vibration frequency shifts, physically relate to each other, learning their respective representation of contact dynamics.

During pretraining, each of the two encoders compresses its input into an embedding $z \in \mathbb{R}^{128}$. The embedding is passed into different heads, each one with a different goal metric. Depending on how well the embedding predicts the goal metric, each head produces its own loss which is summed to produce the gradient used in backpropagation, which tunes the encoder to better encode the relevant information.

Self-supervised visuotactile encoder

The visuotactile encoder is a `ConvStem2d` (Section 5.3.5) that tries to estimate the RMS vibration envelope and the position of a set of masked markers from the current marker position. Its output is an embedding of size 128.

Vibration Energy Envelope Head The visuotactile encoder tries to learn the **root-mean-square (RMS) vibration energy envelope**, which is a measurement that quantifies the instantaneous mechanical power that is dissipated through the sensor [2]. Visualized, it draws the smooth amplitude contour around the signal. In this thesis the following formula is used:

$$e_k = \log\left(1 + \sqrt{\frac{1}{M}\sum_{m=1}^{M} \|\mathbf{a}_{k,m}\|^2}\right), \quad k = 1, \ldots, 16, \tag{5.10}$$

where $\mathbf{a}_{k,m}$ is the acceleration vector formed from the x, y, and z components of sample m within bin k. The 512 sample window is divided into $k = 16$ bins, with $M = 32$ samples each.

Marker Mask Head In addition, to encourage the visuotactile encoder to learn spatial coherence, the current marker flow is passed into a masked embedding, where 30% of the markers are hidden. The visuotactile encoder has to estimate the mean motion of the hidden markers.

Self-supervised Vibration Encoder

The vibration encoder is a `CausalConv1dStack` (Section 5.3.5) that projects 512 acceleration samples ($128\,\text{ms}$ at $4\,\text{kHz}$) into a 128-dimensional embedding vector.

Marker Kinematics Head In the reverse direction, the inertial embedding z_{vib} is trained to predict the mean displacement and velocity vectors of the optical markers in the most recent tactile frame:

$$\bar{\mathbf{s}} = \left[\overline{\Delta x},\, \overline{\Delta y},\, \bar{v}_x,\, \bar{v}_y\right]^\top \in \mathbb{R}^4. \tag{5.11}$$

This teaches the 1D vibration encoder to map high-frequency vibration signals directly to motion flow on the visuotactile sensor.

Pretrain objective

The joint pretraining loss is a weighted sum of the four self-supervised terms:

$$\mathcal{L}_{\text{SSL}} = 1.0\,\mathcal{L}_{\text{env}} + 0.5\,\mathcal{L}_{\text{marker}} + 0.5\,\mathcal{L}_{\text{mask}} + 0.1\,\mathcal{L}_{\text{NCE}}. \tag{5.12}$$

With the fourth term being calculated via symmetric InfoNCE loss, to enable cross-Modal contrastive alignment:

Cross-Modal Contrastive Alignment To align the two distinct modalities in a common geometric latent space, normalized linear projections of z_{gel} and z_{vib} are optimized using a symmetric InfoNCE loss as described in Oord et al. [85]:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{2}\sum_{i=1}^{B}\left(\log\frac{\exp(\tilde{z}_{\text{gel},i}^\top \tilde{z}_{\text{vib},i}/\tau)}{\sum_{j=1}^{B}\exp(\tilde{z}_{\text{gel},i}^\top \tilde{z}_{\text{vib},j}/\tau)} + \log\frac{\exp(\tilde{z}_{\text{vib},i}^\top \tilde{z}_{\text{gel},i}/\tau)}{\sum_{j=1}^{B}\exp(\tilde{z}_{\text{vib},i}^\top \tilde{z}_{\text{gel},j}/\tau)}\right), \tag{5.13}$$

where $\tau = 0.1$ is the temperature parameter and weighing factor ($w_{\text{NCE}} = 0.1$).

The result of the pretraining step is two encoders with an output embedding of size 128 each.

Fine-Tuning

The encoders are fine-tuned using an appended MLP, consisting of a linear layer, ReLU layer, and a dropout layer. Fine-tuning trains the pipeline end-to-end, meaning that the encoder weights are not frozen and are updated.

5.5.4 Parametric Decomposition of Slip Breakaway Event (`Event`)

Slip events are sudden very high energy events. They are characterized by a sudden velocity spike, as the object transitions from a state of static friction to a state of dynamic friction, which diminishes once the potential energy stored in tension between the two surfaces has been released. The following architecture is the attempt to give a neural network the tools to properly model this slip event.

The following output reconstruction formula is used:

$$\hat{v}(t) = v_{\text{plateau}} + \underbrace{p_{\text{spike}} \cdot A \cdot \underbrace{\exp\left(-\frac{1}{2}\left(\frac{t-\Delta t}{w}\right)^2\right)}_{\text{Gaussian pulse}}}_{velocityspikeestimation} \tag{5.14}$$

Where v_{plateau} is the steady-state sliding velocity after the slip spike, $p_{\text{spike}} \in [0, 1]$ is the predicted probability that a breakaway event is occurring, A is the excess peak amplitude above v_{plateau}, Δt is the temporal offset of the peak relative to the current anchor frame ($t = 0$), and w is the Gaussian pulse width governing the duration of the transient.

The new head gives the velocity plateau and spike its own parameters, so committing to a spike no longer means corrupting the plateau estimate. If it predicts no event, the head collapses back to the previous plateau regressor, so the change adds a capability without risking the baseline. The Gaussian forces the spike to be short-lived, so it can't become a second plateau.

The neural network is mainly marker based, using the branch-encoder explained in Section 5.3.5 to encode a tactile marker flow, vibration, vibration STFT, and vibration summary.

The encoded feature vectors are concatenated and passed through a small fusion MLP to reduce it to a 128 sized vector. This vector is now further reduced to the spike head for estimating the different variables from Equation 5.14. Each variable gets its own function:

Softplus for the unbounded non-negative magnitudes v_{plateau} and A as both are velocities, and need positivity without a ceiling. For A the non-negativity is also what keeps the bump from cancelling the plateau instead of adding to it.

Sigmoid for the bounded quantities p_{spike} and w, used raw where the natural range is $[0, 1]$ and affinely rescaled onto $[5, 50]$ ms for the width. The width is the one parameter needing a two-sided bound: zero would divide by zero, and an unbounded width would let the transient flatten into a second plateau.

Tanh for Δt, the only quantity that is bounded and signed, since the peak can fall either side of the anchor.

The head is initialized so that the network starts as a plain plateau regressor, with the spike probability at 0.12 and the amplitude near zero. Should the network infer no event, the reconstruction reduces to v_{plateau} and the architecture degenerates to the behaviour of the preceding networks, with the aspiration being that the added parameterization allows spikes to be estimated better than other networks.

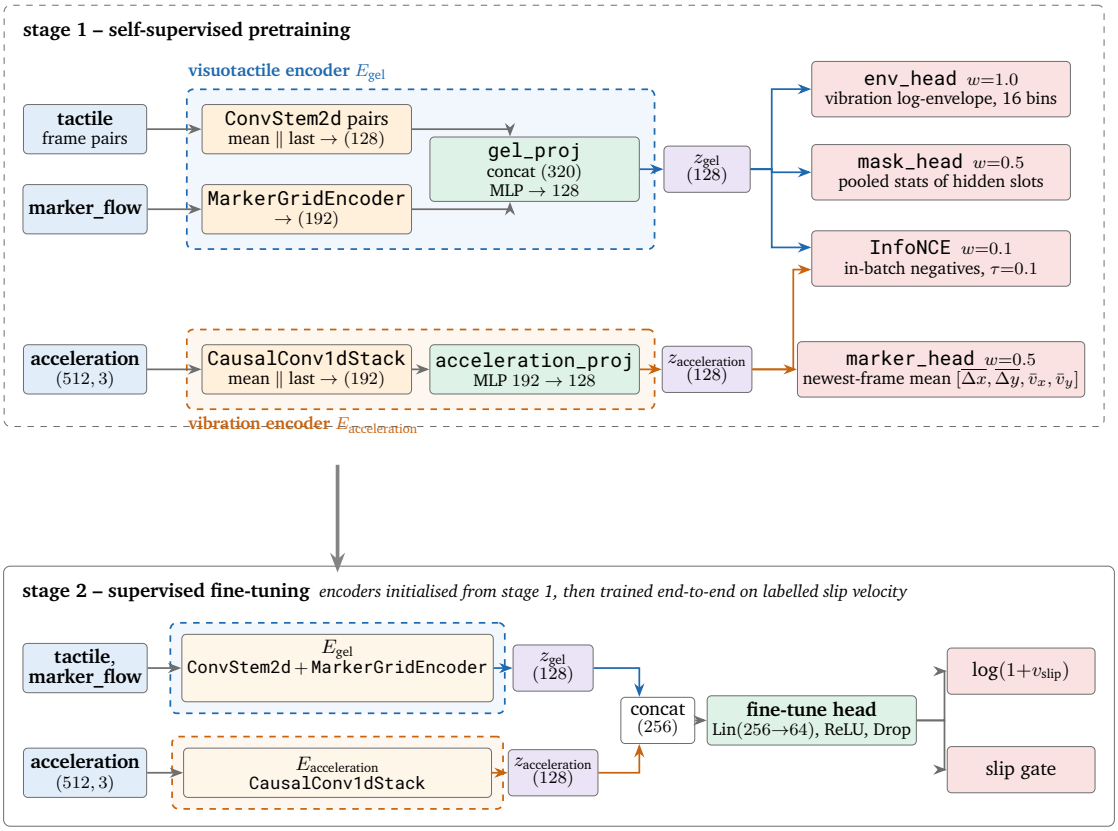


Figure 5.11: `Twin`: two small encoders produce z_{gel} and z_{vib}. Stage 2 (solid) fine-tunes the standard gated (hurdle) head on top; stage 1 (dashed) pretrains the encoders with four pretext heads whose targets are all computed from the same batch, so no labelled velocity is needed for this stage.

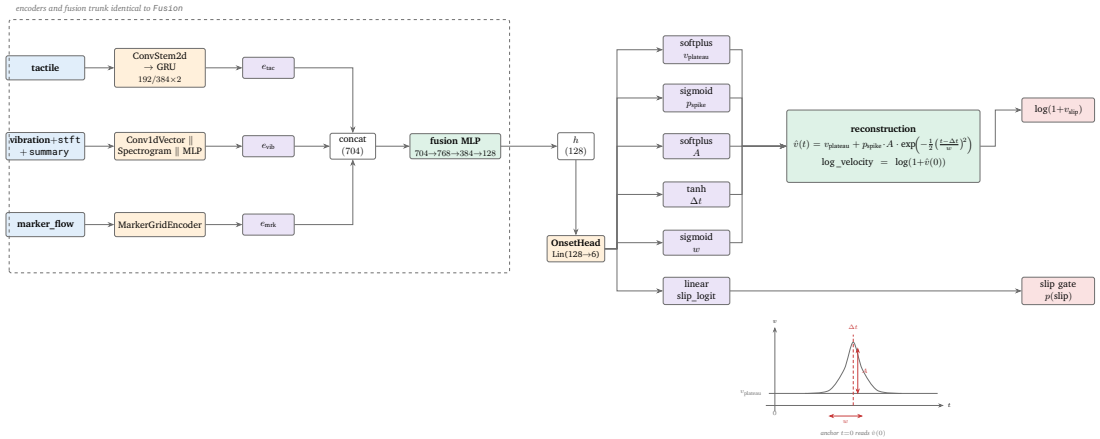


Figure 5.12: `Event`: identical branch encoders and fusion trunk to `Fusion` (Figure 5.6); only the head changes. A plateau velocity and a parametric spike bump are reconstructed at the anchor time into the same $\log(1+v)$ quantity every network predicts, so the training recipe and the leaderboard are unaffected.

5.5.5 Evidence-Weighted Fusion Network (`Evidence`)

This architecture is intended to provide robustness against wrong or poisoned sensor streams. This might be external light fluctuations for the GelSight, or vibration from external sources for the ADXL355.

The branch encoders of Figure 5.5 again form the base of this neural network. What differs now is that each branch gets its own regressor head. Each branch predicts its own log-velocity prediction, its own log-variance, and its own slip-gate logit. This idea is taken from Hinton [38]. Fusion of the sensor streams is then done arithmetically rather than learned:

$$\lambda_i = \exp(-s_i) \cdot \text{present}_i, \qquad \mu^* = \frac{\sum_i \lambda_i \mu_i}{\sum_i \lambda_i}, \qquad s^* = -\log \sum_i \lambda_i \tag{5.15}$$

Each branch outputs a mean and a log-variance instead of a point estimate, as such when multiplying their Gaussians, the product is another Gaussian. Completing the square in the exponent collects all heads of the target into a single quadratic term, whose coefficient and centre are the pooled precision $\sum_i \lambda_i$ and the precision-weighted mean of Equation 5.15. The fusion rule is treating the branches as independent experts on the same quantity, with absence entering as $\lambda_i = 0$. Because the Gaussian negative log-likelihood of a branch is minimised at $s_i = \log((\mu_i - y)^2)$, each branch is driven to report its own squared error as its variance, so a stream that cannot explain the current sample and has a high variance has its weight in the final fusion reduced.

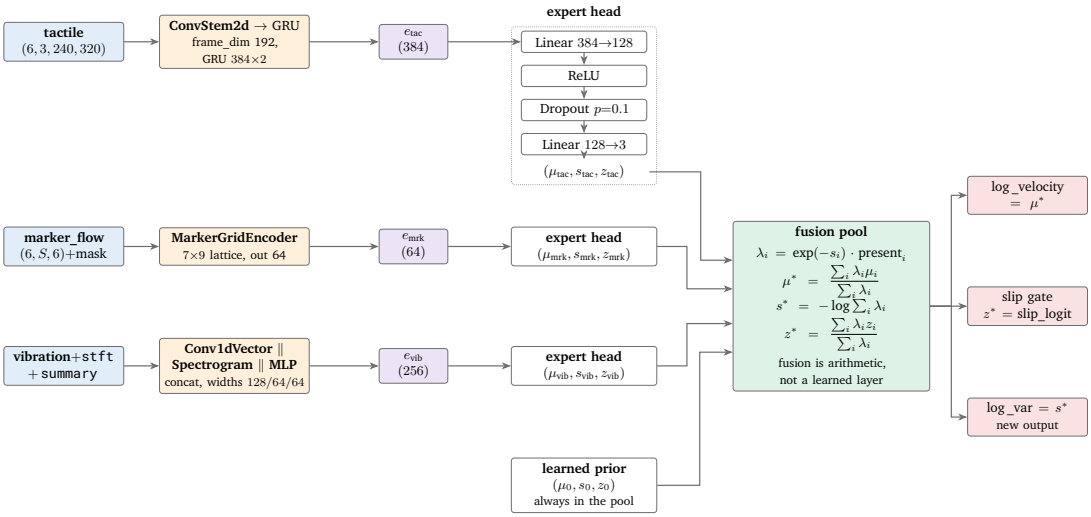


Figure 5.13: `Evidence`: the same branch encoders as `Fusion` (Figure 5.6), each ending in its own expert head (μ_i, s_i, z_i). A learned prior expert keeps the pool from being empty; the pooled mean, gate and log-variance are a closed-form inverse-variance combination, not a learned layer.

6 Results

The following section describes the evaluation of the neural networks described in Sections 5.4 and 5.5.

In summary, all neural network architectures performed similarly, within a 17 % margin of each other. The best performing architecture was the Hybrid architecture, but each neural network has its own strengths.

The structure of this chapter is as follows: Starting the chapter will be a section on comparing the overall accuracies of the models. Following that is an analysis section on where the error originates from. The following three sections look at how variations in different parameters, namely object surface, network size, and input data, affect the accuracy of different neural networks.

Since timing is important for slip-control, a section on latency and onset-time is presented. Lastly, a section on the limitations shows what inaccuracies could have been caused by design choices.

6.0.1 Baseline

The baseline is the total average over all validation sets. For slip frames, it is $8.3\,\mathrm{mm/s}$, for no-slip frames it is 0.61 mm/s and for all frames it is $2.38\,\mathrm{mm/s}$. The no-slip baseline is the noise floor of the velocity measurements.

6.1 General Results

Data was generated as described in Section 4.7. 290 recordings were used, totaling about 3 hours of recording time. The training/test/validation split was 70 %/15 %/15 %.

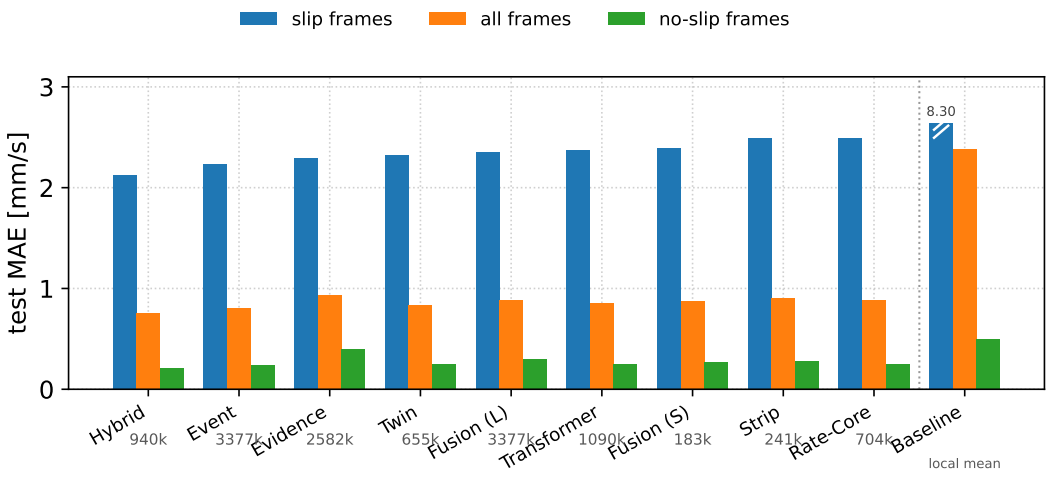


Figure 6.1: Slip / all / no-slip test MAE, nine networks, sorted by slip-frame MAE. Each bar is labelled with its network's parameter count. The tenth column is the predict-mean baseline.

When looking at the results, the base comparison metric is the Mean Absolute Error (MAE). The MAE describes how much the inferred measurement differs from the actual measurement across all object surfaces in the whole data corpus. The error describes the difference between a velocity measurement as determined by the ArUco marker on the object, compared to the inferred slip velocity, both given in mm/s.

The second metric is the Area Under the Receiver Operating Characteristic curve (AUROC) of the slip gate, the classification head introduced in Section 5.3.4. Sweeping the decision threshold on the gate probability $p_\theta(\text{slip} \mid x)$ traces out the true-positive rate against the false-positive rate, and the area under that curve is the probability that a randomly drawn slip frame is scored higher than a randomly drawn no-slip frame. A value of 1.0 means perfect separation and 0.5 means the gate carries no information. Unlike gate accuracy, the AUROC does not depend on a chosen threshold and is not inflated by the fact that only 28.5 % of frames are labelled as slip.

The results can be seen in Table 6.1. Of the eight architectures introduced in Chapter 5, Fusion is evaluated at two sizes (Large and Small), giving nine trained networks in total; the rest of this chapter refers to individual rows of Table 6.1 as "networks" and to the eight

Table 6.1: All architectures ranked by slip-frame MAE. Inputs: T tactile frames, D depth, F marker flow, A vibration sensor, S vibration spectrogram, U vibration summary. *fwd* is the median forward pass latency; *e2e* adds the GelSight frame period and the per-frame input cost a deployed system pays, and *Rate* is the loop rate it implies (both defined in Section 6.7).

Network	Params [M]	Inputs	MAE_{slip}	MAE	$MAE_{no\text{-}slip}$ [mm/s]	RMSE	Gate AUROC	fwd [ms]	e2e [ms]	Rate [Hz]
Hybrid	0.94	T+F+A	2.13	0.76	0.21	2.68	0.999	8.7	62.2	16
Event	3.38	T+A+S+U+F	2.23	0.80	0.24	2.74	0.999	12.9	66.3	15
Evidence	2.58	T+A+S+U+F	2.29	0.93	0.39	3.26	0.999	13.7	67.2	15
Twin	0.66	T+F+A	2.32	0.84	0.24	3.39	0.999	9.2	62.6	16
Fusion (L)	3.38	T+A+S+U+F	2.35	0.88	0.29	3.50	0.999	12.7	66.2	15
Transformer	1.09	T+A+S+U+F	2.37	0.85	0.25	3.43	0.999	10.0	63.5	16
Fusion (S)	0.18	T+A+S+U+F	2.40	0.87	0.27	3.71	0.999	7.1	60.6	17
Strip	0.24	A+F	2.49	0.91	0.27	3.70	0.999	2.5	55.9	18
Rate-Core	0.70	T+F+A	2.49	0.89	0.25	4.00	1.000	9.6	63.0	16

underlying designs as “architectures”. For most of the results section, the focus will be on the slip-MAE, so how accurately an architecture was able to describe the slip events. Since all architectures use a two-headed approach, with one head inferring if slip is happening, the no-slip error is not that important since a controller can rely on the AUROC signal instead.

The results show that the Hybrid architecture is the best performer with a slip-MAE of 2.13 mm/s. Though all architectures performed similarly, with even the least performant network, RateCore, only performing 17 % worse with a slip-MAE of 2.49 mm/s. To put this into perspective Chen et al. achieved an MAE of 5mm/s [13].

Hybrid has some advantages over Fusion and Transformer that could explain its better performance. Fusion weights each branch by a fixed weight as the branches are fed into the trunk. This weighting is fixed after training, which is very likely suboptimal. For example, during incipient slip a higher weight on the visuotactile image might be beneficial for detecting shear forces, while during steady slip the vibration data might carry more information. In `Hybrid`, the mixing coefficients are computed from the token contents at inference and can therefore follow this shift from frame to frame.

Transformer also has this advantage, but here the cost of the continuous-time Fourier embedding might be losing it performance. This embedding is most valuable when the token spacing varies a lot, and it is likely that the variance in the vibration and visuotactile data is not large enough. The Transformer tokens therefore have to compress more

information into the same space as those of Hybrid, without the situation that would make the embedding useful ever occurring.

Regarding detecting if slip is happening, all neural network architectures performed essentially perfectly. When looking at Figure 6.3, it becomes apparent that detecting slip is essentially a solved problem, being learned within even the first few epochs.

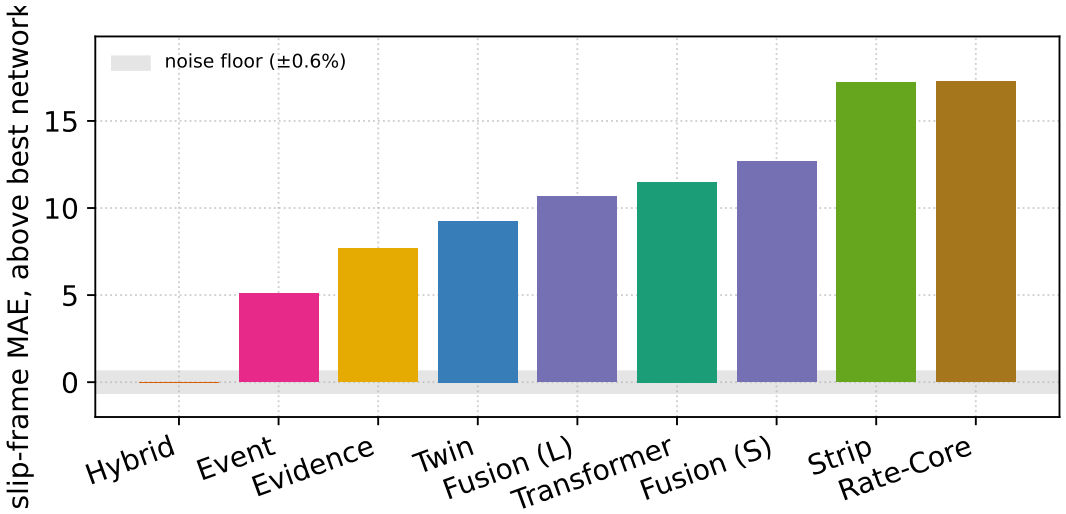


Figure 6.2: Slip-frame MAE relative to the best network.

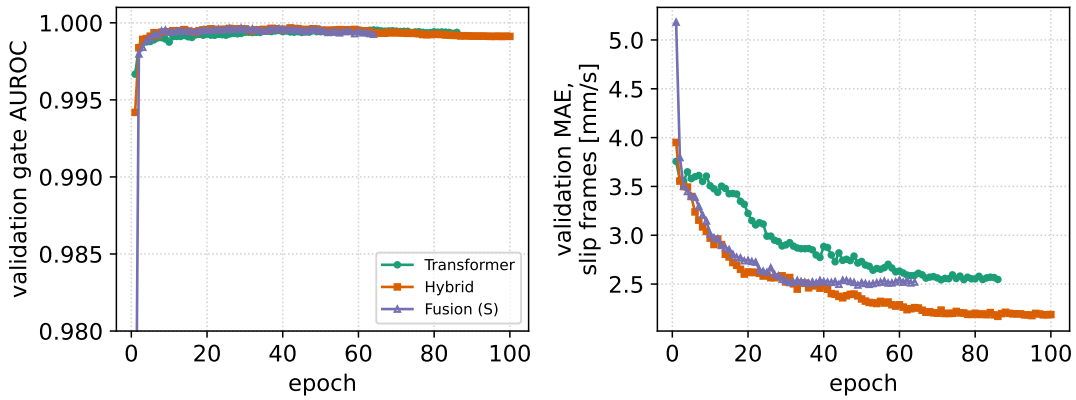


Figure 6.3: Validation gate AUROC (left) and validation slip-frame MAE (right) against epoch, Transformer / Hybrid / Fusion (S).

6.2 Error Sources

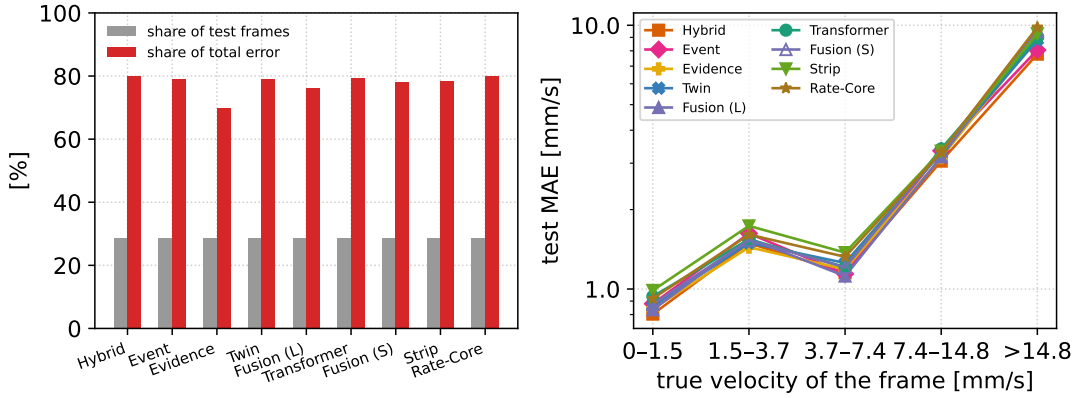


Figure 6.4: Left: share of test frames that are slipping against the share of total error they cause. Right: test MAE against the true velocity of the frame, one line per network, log scale.

As alluded to previously, the most difficult areas to predict are the instances where slip occurs. In addition, MAE scales with slip velocity. This can be seen in Figure 6.4, which

shows that as slip velocity increases, so does the MAE in an exponential manner.[1] Slip frames make up only 30,% of the test corpus, but they carry between 70 and 80 % of all absolute error. The remainder falls on the no-slip majority, where the error is small but not uniform: all architectures sit between 0.21 and 0.23 mm/s. slip and no-slip MAE also don't correllate, `Evidence` has the third-best slip-MAE of the nine networks but the worst overall MAE, entirely because of how it behaves on frames where the object is not moving, predicting spikes where no slip was occurring at all.

Most of the error during slip frames stems from two phenomena, both visible in Figure 6.16: low-frequency vibrations and slip-velocity spikes. The oscillations, in the 10 Hz to 30 Hz band, are a result of the natural slip vibration and of motor vibrations, and all architectures average over them rather than tracking them, which accumulates error.

Being unable to track these vibrations is not detrimental to slip-control. At those frequencies the signal is of little use to a controller: at the end-to-end latencies reported in Figure 6.12, the architectures close a loop at 10 Hz to 18 Hz, so a grip controller could not act on a 30 Hz component even given a perfect estimate of it. The averaging therefore performs the low-pass step the control loop would have to apply itself, and this share of the error is better read as a bandwidth mismatch than as a deficiency of the estimator.

The second, and more detrimental to slip velocity control, are the velocity-spikes, or ruptures, as described in Section 2.1.2. In the data slip velocity peaks usually happen within the 36ms of slip-onset, but can also occur later.

Within the slip band, the error is worst at the fast end in absolute terms and worst at the slow end in relative terms. Above 14.8 mm/s the MAE reaches about 9 mm/s, while in the slowest bin, below 1.5 mm/s, it is about 0.9 mm/s and therefore of the same order as the velocity that is being estimated. The two ends have different causes. At the slow end, the ground-truth jitter of Section 6.0.1 is an absolute quantity that does not shrink with the velocity being measured, so it takes up an ever larger share of the error as the target approaches zero. At the fast end, the cause is the training target of Section 5.3.4 rather than the network: the regression head fits $\log(1 + v)$ under a Huber loss masked to slip frames, and the logarithm compresses exactly this part of the range, so that the difference between 15 and 30 mm/s is small in log space and a large error there costs the optimizer little. The loss also averages over all slip frames alike, and the fastest frames are a small minority of them, so their error is averaged away in the very quantity the

[1] The decrease around 25–50 px is most likely due to two factors. One is that around that slip velocity the natural frequency of either the object or the GelSight was excited, which made sliding smoother, and the y-scale of the graph is logarithmic, so this reduction is in reality only about 1 mm/s deep.

checkpoint is selected on. That the shape of the curve is the same for all nine networks supports this. Lifting the fast end therefore calls for a different regression target, such as linear or square-root space, or for weighting the loss by velocity magnitude.

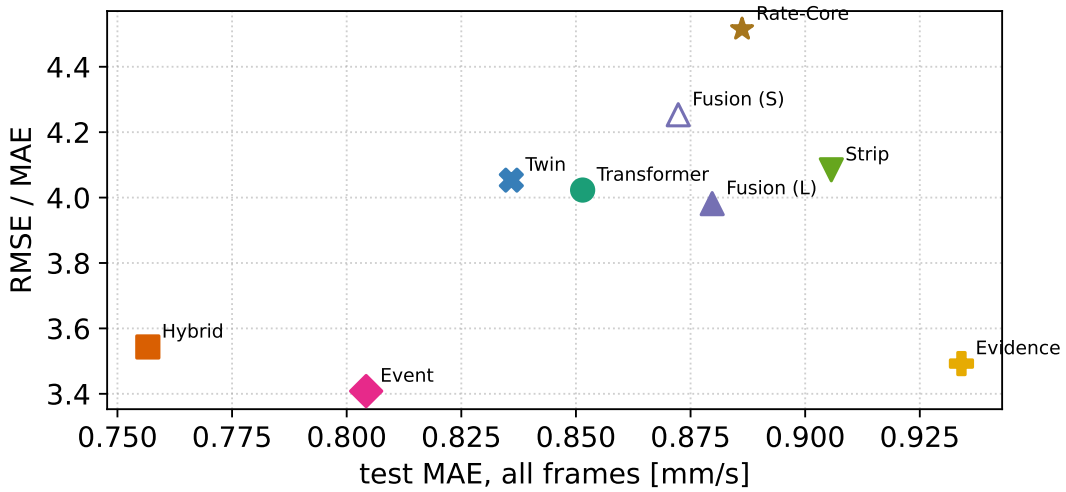


Figure 6.5: RMSE / MAE against overall test MAE, one point per network.

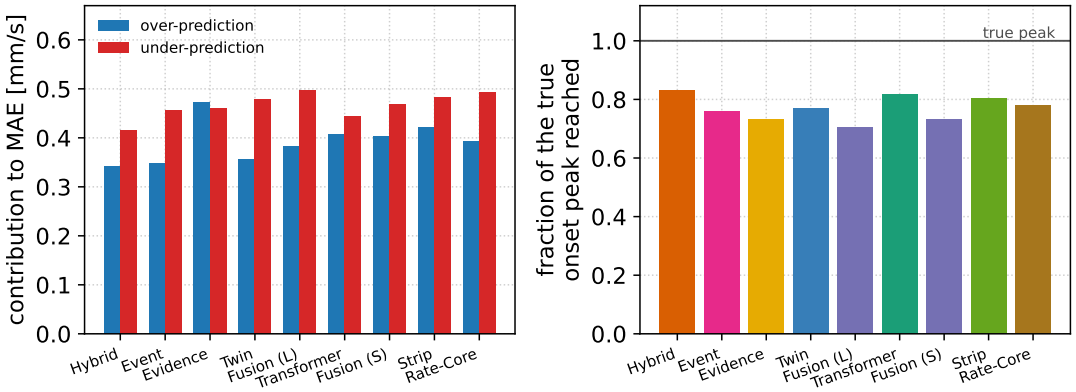


Figure 6.6: Left: contribution of over-prediction and under-prediction to MAE, per network. Right: fraction of the true velocity peak reached, per network.

6.3 Unseen Surface Test

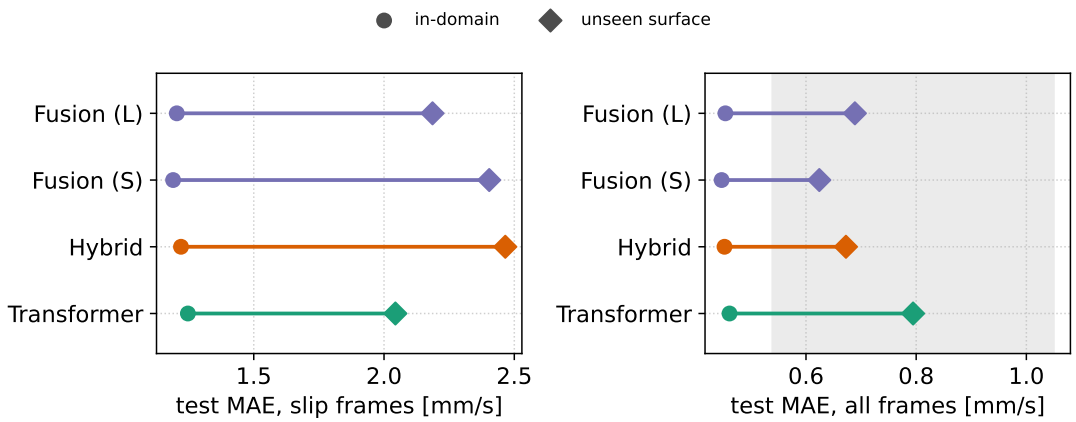


Figure 6.7: In-domain to unseen-surface, one line per network. Left: slip frames. Right: all frames.

Table 6.2: In-domain and unseen-surface test performance, four networks.

	in-domain		unseen surface		
Network	MAE_{slip}	MAE [mm/s]	MAE_{slip}	MAE	Gate AUROC
Transformer	1.25	0.46	2.04	0.79	0.988
Hybrid	1.22	0.45	2.47	0.67	0.989
Fusion (S)	1.19	0.45	2.40	0.62	0.988
Fusion (L)	1.20	0.45	2.19	0.69	0.981

The surfaces from Figure 4.1 were used in an unseen surface test. In the unseen surface test, the neural networks were only trained on the dotted surface, and evaluated on the diamond surface.

All architectures perform worse, with general MAE rising by 40–72 % and slip-MAE increasing by 64–102 %. The results are visualized in Figure 6.7. Slip detection generalizes far better; even in the worst case with the Fusion network, the detection rate only drops by 0.9 %.

Interestingly `Transformer` generalized the best on slip frames, while generalizing the

worst on all frames. `Hybrid` in turn generalized the worst on slip frames, but performed better on all frames, a similar story to the `Fusion` architectures. This inconsistency is hard to interpret, and requires more research to pin down to a cause.

6.4 Sensor Ablation

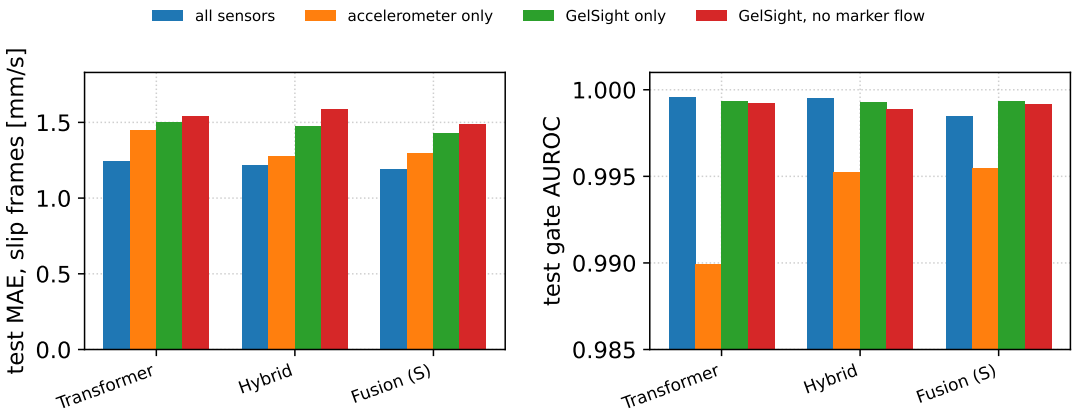


Figure 6.8: Ablation test for the three base architectures Left: slip-frame MAE. Right: gate AUROC, same input sets, on a truncated 0.985–1.000 axis; the bars stand on that baseline, not on zero.

For the sensor ablation, an earlier corpus of 190 items was used. This corpus did not include the same slip velocity range as the full later corpus, and as such the overall MAE is significantly lower. For each ablation, the respective input branches were removed. For `Hybrid` and `Transformer` this meant that those tokens now contained "missing" values. The ablation test was only done on the three main architectures: `Transformer`, `Hybrid`, and `Fusion Small`. `Fusion Small` was used since it showed very similar performance to its larger counterpart, and using the small one also tested how sensor ablation impacted smaller networks. The tests that were run were: all sensors, vibration-sensor-only, GelSight-only with marker flow, and GelSight-only without marker flow.

The results showed that for slip velocity, the vibration sensor is more important than visuotactile input. Removing the visuotactile input degraded performance of slip-MAE by 9.2 % for `Fusion`, 4.9 % for the `Hybrid`, and 16 % for the `Transformer` architecture.

Dropping the vibration sensor stream was more costly, with `Fusion` degrading by 20 %, `Hybrid` by 21 %, and the `Transformer` by 20 %. Further dropping the marker flow exacerbates this to 25 % for `Fusion`, 30 % for `Hybrid`, and 23 % for `Transformer`.

For detecting if slip is occurring, this relation is somewhat inverted, with GelSight-only ablations outperforming vibration-sensor-only, though the difference in absolute terms is less than 1 %. This result is also intuitive, as if slip is happening, it is very apparent in the tactile image and marker flow, while harder to discern from the base motor noise in the vibration signal.

This confirms the hypothesis that most of the information about slip velocity is encoded in the slip vibrations, while information about incipient slip and if the object is slipping is better detected using visuotactile data. The more nuanced takeaway is that all architectures still perform decently with only one sensor modality. Especially when only using vibration data, all three networks still achieve good performance, and future work could look into vibration only slip detection, using inexpensive accelerometers.

Table 6.3: Sensor ablation: three base architectures, with four different input sets.

Network	Inputs	MAE_{slip} [mm/s]	Gate AUROC
Transformer	all sensors	1.25	1.000
	accelerometer only	1.45	0.990
	GelSight only	1.50	0.999
	GelSight, no marker flow	1.54	0.999
Hybrid	all sensors	1.22	1.000
	accelerometer only	1.28	0.995
	GelSight only	1.48	0.999
	GelSight, no marker flow	1.59	0.999
Fusion (S)	all sensors	1.19	0.998
	accelerometer only	1.30	0.995
	GelSight only	1.43	0.999
	GelSight, no marker flow	1.49	0.999

6.5 Depth Input

Tests with all the base architectures showed that adding depth degraded performance rather than improved it. The results can be seen in Table 6.4, where `Fusion Large`

gets worse by 4.4 %, `Transformer` gets worse by 1.9 %, and `Hybrid` gets worse by 0.7 %. This difference is relatively small in terms of error, but as will be seen again later in Section 6.7, the real cost is in latency, with the end-to-end latency increasing by 57 % (37.6 ms), 49 % (31.7 ms), and 56 % (35.4 ms) for the `Fusion Large`, `Transformer`, and `Hybrid` architectures, respectively. Slip detection shows no change in performance across the three models.

The reason for the degraded performance is not clear, although it might be the case that depth is only usable with the larger parameter sizes. The parameter size tests in Section 6.6 show that the sizes of the base `Fusion`, `Transformer`, and `Hybrid` architectures were already plateaued, with `Hybrid` even overfitting. Even if a larger parameter size could bring improvement, the increased latency is not worth the trade-off of the most likely marginal improvement.

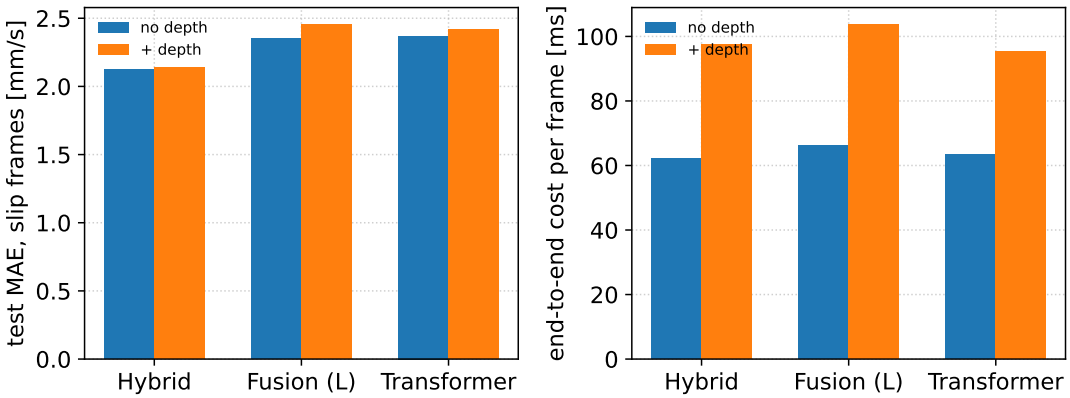


Figure 6.9: `Hybrid`, `Fusion (L)` and `Transformer`, with and without a `depth` branch. Left: slip-frame test MAE. Right: end-to-end cost per frame.

Table 6.4: Depth-input ablation. Each network is scored against its own no-depth default from Table 6.1.

	MAE_{slip} [mm/s]			e2e cost [ms]		Gate AUROC	
Network	no depth	+ depth	Δ [%]	no depth	+ depth	no depth	+ depth
Hybrid	2.13	2.14	+0.7	62.2	97.6	0.999	0.999
Fusion (L)	2.35	2.46	+4.4	66.2	103.8	0.999	0.999
Transformer	2.37	2.42	+1.9	63.5	95.2	0.999	1.000

6.6 Parameter Size

For this test, different-sized implementations of the base architectures were trained on the whole test corpus. Scaling was done by width alone: at every rung the tactile CNN stem, the per-modality encoder branches (vibration, marker, spectrogram) and the fusion trunk or head were narrowed together, while the input set, branch topology and dropout wiring stayed identical.

The size ladder started with about 1 % of the base architecture's parameter size and went up to about 20 %. The exact numbers depended on the implementation specifics of the architecture. The ladder was chosen as most of the MAE reduction happened in those first 20 %, better illustrating the trend. The Transformer carries one further rung above its base configuration, at roughly twice its parameter count, to check whether that architecture was still capacity-limited at the top of the ladder.

The results can be seen in Figure 6.10 and Table 6.5. For completeness, the themed architectures are also included in Figure 6.10.

These results show that architecture design beats parameter size, with Hybrid Small beating Fusion Large by 10 % with 13.5× fewer parameters. Increasing parameter size also has strongly diminishing effects, with 20 % of the parameters already providing 90 % of the performance, as can be seen in Figure 6.11. Hybrid as well as Transformer even lose some performance at a larger parameter size, most likely overfitting on the training data.

This result becomes significant in conjunction with the next Section 6.7 on latency, as for designing a slip-controller latency and performance has to be balanced against the parameter size and the associated inference latency.

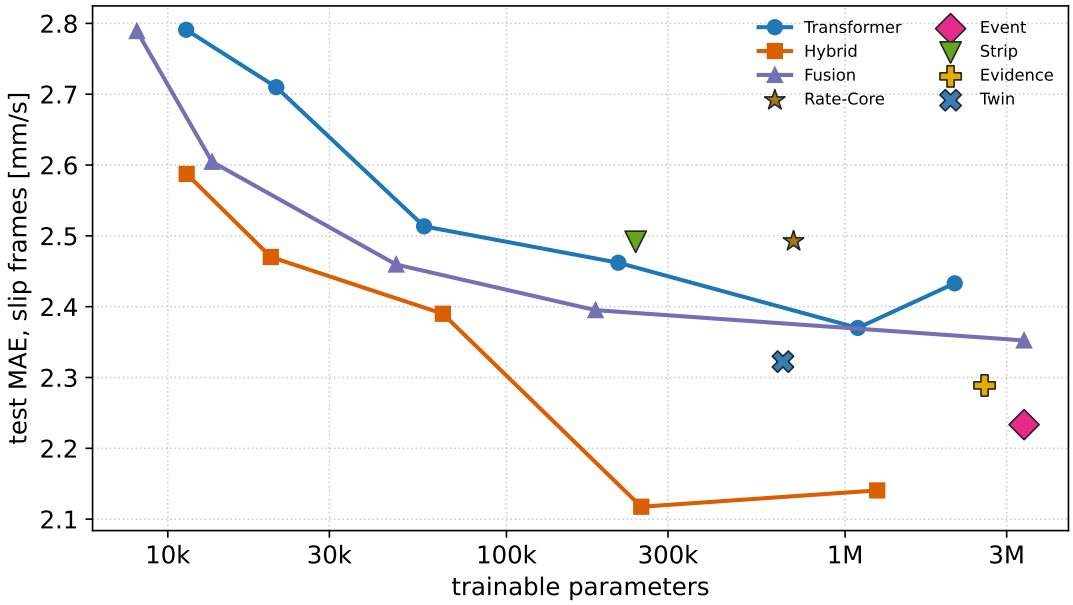


Figure 6.10: Slip-frame test error against parameter count.

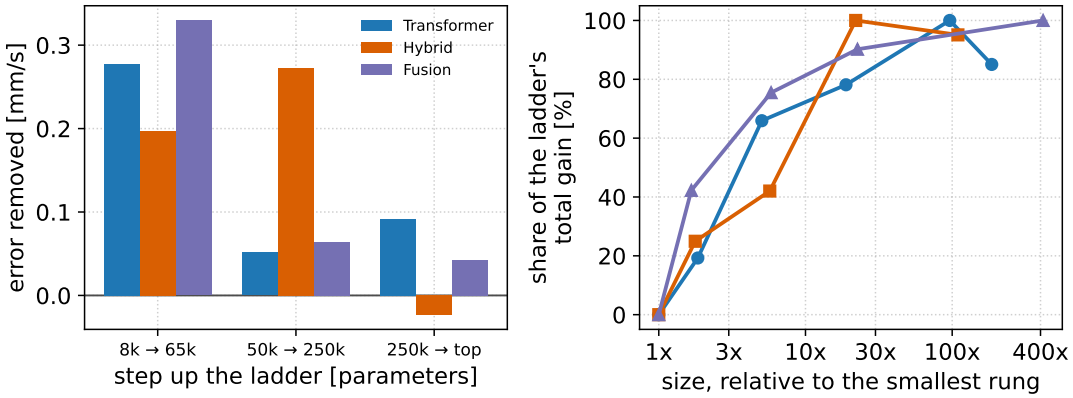


Figure 6.11: Left: error removed by each step up the ladder. Right: cumulative share of each ladder's total gain, against size relative to its own smallest rung.

Table 6.5: The full size ladder: three architectures, six rungs for the Transformer and five each for the Hybrid and Fusion.

Ladder	Rung	Params	MAE_{slip} [mm/s]	Latency [ms]
Transformer	nano	11k	2.79	6.1
	micro	21k	2.71	6.0
	tiny	57k	2.51	6.6
	small	214k	2.46	7.2
	base†	1,090k	2.37	10.0
	large	2,107k	2.43	8.3
Hybrid	nano	11k	2.59	7.7
	micro	20k	2.47	7.8
	tiny	65k	2.39	7.9
	small	250k	2.12	9.0
	base†	1,244k	2.14	15.6
Fusion	nano	8k	2.79	6.2
	micro	14k	2.60	6.0
	tiny	47k	2.46	6.3
	small†	183k	2.40	7.1
	large†	3,377k	2.35	12.7

6.7 Latency

Latency was measured on a Ryzen 5 5600X. There are two types of latency in the test setup: network latency and end-to-end latency. Network latency is the inference time of the respective architecture, while end-to-end latency is the time from when the first timestamp arrives at the PC to the time that the first result is produced. Latencies were measured by running a forward pass over 760 samples, repeated three times with the average taken. Since latency varied by up to 9% across different days, the latency test conducted alongside each respective test group was used for the corresponding results.

Picking up from the parameter size relationships in the previous Section 6.6, parameter size stays roughly flat within the same architecture and only changes with the top rung of the size ladder within the same architecture.

Across architectures, this relationship does not hold, with virtually all neural networks that have a single CNN image stem having an inference time within 10%. End-to-end computation is where most of the difference becomes apparent. The time from the first

timestamp to the data being loaded into the neural network is on average 52.6 ms across all eight architectures, none of which use depth by default (Section 6.5). Figure 6.12 adds a tenth bar, `Hybrid` with depth input, on top of the normal no-depth `Hybrid`, so the depth cost stays visible: as alluded to in Section 6.5, depth computation is expensive, adding 34.9 ms on average.

The lowest-latency architectures are, as expected, the `Strip` and `Rate-Core` fast tier. `Strip` only uses marker flow and vibration signal, so no image data is involved. This makes the forward pass very efficient, taking only 2.5 ms, 80% faster than the fastest base architecture, `Fusion S`.

While the `Rate-Core` slow tier was more in line with the other architectures in terms of latency, `Rate-Core` was specifically designed for fast inference. With its minimal `Conv1D` stack, it delivered on this design intention, achieving latencies of 1 ms.

Across most architectures, inference time is dominated by the shared CNN image stem, so networks converge to similar latency regardless of head design, while end-to-end latency is shaped mainly by data loading and, if enabled, the costly depth computation (+34.9 ms). Only architectures that drop image input entirely (`Strip`) or use a minimal-depth network by design (`Rate-Core`) manage to outperform this, achieving sub-3 ms inference.

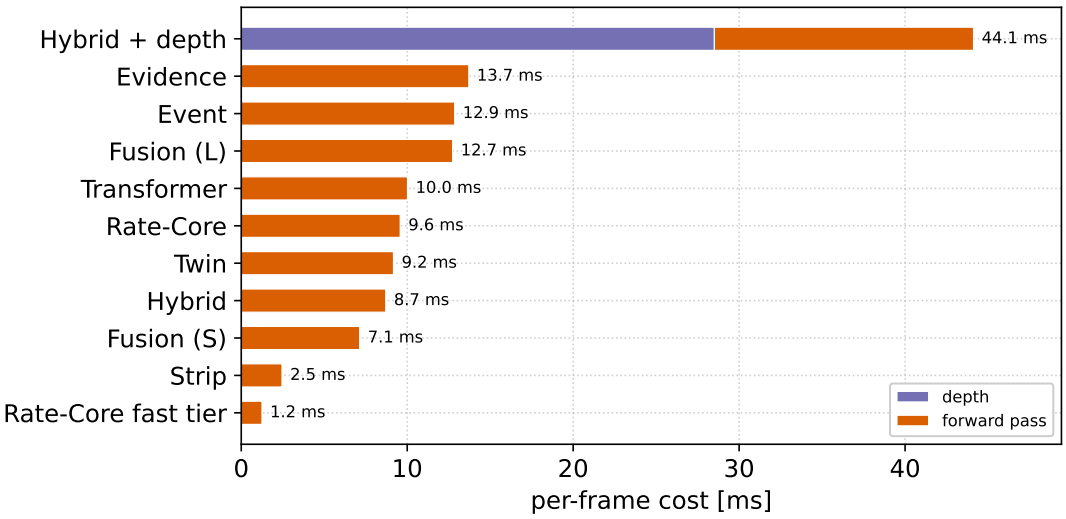


Figure 6.12: Latency for each architecture, plus a tenth bar, Hybrid with depth (Section 6.5), added since none of the nine default networks use depth. All numbers are rounded down.

6.8 Label Efficiency (Twin)

The fundamental question that Twin was intended to answer was if pre-training the encoder could increase the training efficiency.

The results point towards pre-training not significantly increasing the label efficiency, as can be seen in Figure 6.13 and Table 6.6.

For this test, Twin was pre-trained on 10, 30, and all 203 files, and then evaluated against this architecture trained on the same training corpus. For the pre-trained net, the pre-train and train corpus were always the same size. Even using the whole corpus for pre-training, the advantage over from-scratch training was only 4 %. And this is despite the fact that the pre-trained model sees twice as much data as the scratch-trained network: once during pre-training and once during normal training.

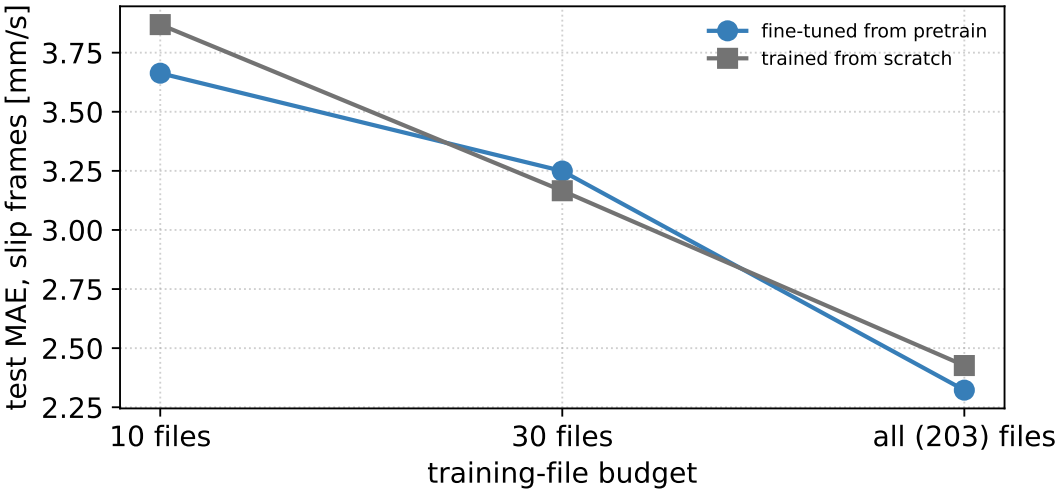


Figure 6.13: Twin, fine-tuned from the self-supervised checkpoint against trained from scratch, at each training-file budget.

Table 6.6: Twin label-efficiency sweep: fine-tuned vs. scratch, three training-file budgets.

Training	Files	MAE_{slip} [mm/s]	MAE [mm/s]	Gate AUROC
fine-tuned	10	3.66	1.27	0.995
	30	3.25	1.14	0.997
	all (203)	2.32	0.84	0.999
from scratch	10	3.87	1.66	0.994
	30	3.17	1.15	0.998
	all (203)	2.43	0.87	0.999

6.9 Slip Velocity Spike

The slip velocity spike can be seen in Figure 6.14 and is caused by the first dissipation of energy after static stick changes to dynamic stick-slip. On average, this happens about 36 ms after the start of a slip event. Its detection and size estimation are crucial, as a controller would have to actively regulate against it, as for regulating slip velocity the gripper has to react to this spike without re-engaging static friction. Figure 6.14 shows that all architectures tend to underpredict the spike magnitude, especially at lower speeds,

since predicting a spike is risky for the neural network: predicting it in the wrong location incurs a costly loss penalty.

This thesis considers a velocity peak as correctly inferred if the neural network correctly predicts a velocity that is at least 70% of the actual slip velocity, but also no larger than the actual spike. This metric is called recall@70 (R@70). The peak ratio is how large the predicted peak was compared to the actual peak.

The performance of the different architectures is shown in Table 6.7 and illustrated in Figures 6.14 and 6.15. Hybrid is again the best performing architecture, with a recall@70 of 72 % and a peak ratio of 83 %. The worst performance was by Fusion Large, which was interestingly substantially outperformed by its smaller variant, most likely due to overfitting, showing that increased parameter size is not correlated with performance.

Event, as described in Section 5.5.4, was intended to combat this problem by including a model of the slip velocity spike in its output head. Unfortunately the results are not promising, with it performing worse than most of the other architectures. The reason for this is not entirely clear: it is one of the largest networks by parameter size, and could be suffering from the same overparameterization as Fusion Large.

Predicting the slip spike velocity is where the prediction-based networks performed the poorest. This is to be expected, as predicting the spike magnitude would require the network to extract the potential stored energy from the visuotactile and vibration data.

Table 6.7: Velocity-peak tracking and false-spike rate, all nine networks.

	onset peak			false spike [%]	
Network	peak MAE [mm/s]	ratio	recall@70	in-event	no-slip
Hybrid	7.54	0.830	0.717	5.9	1.2
Event	8.01	0.759	0.641	7.3	1.4
Evidence	9.55	0.731	0.582	7.0	4.3
Twin	8.35	0.769	0.693	6.6	1.5
Fusion (L)	10.17	0.703	0.502	7.4	1.7
Transformer	7.24	0.816	0.709	7.3	1.5
Fusion (S)	9.47	0.732	0.590	8.1	1.7
Strip	9.01	0.804	0.665	8.6	2.1
Rate-Core	9.07	0.779	0.681	7.5	1.5

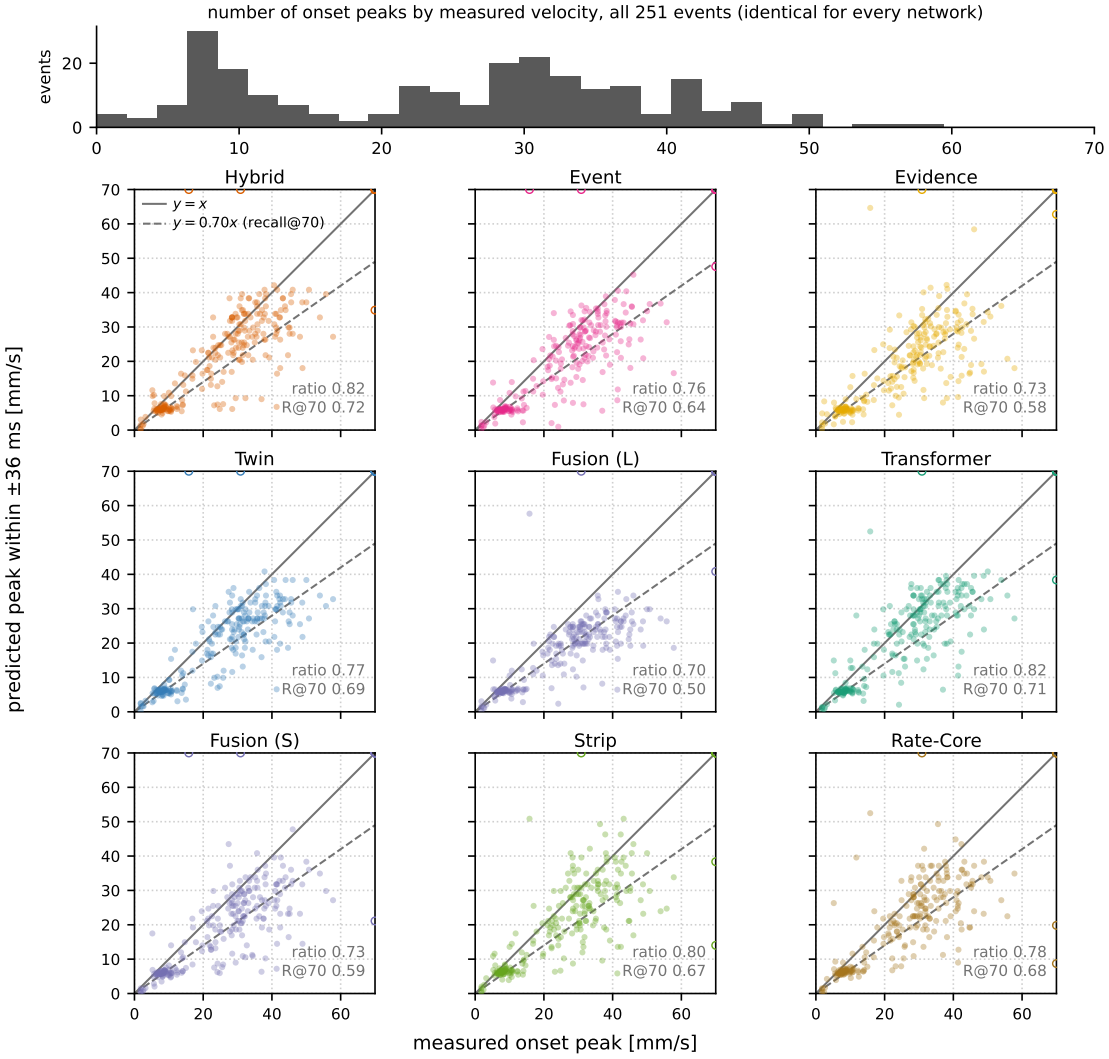


Figure 6.14: Spike prediction performance: measured velocity peak, from the raw pre-despike track, against the highest velocity predicted within ± 36 ms of it, one point per slip event, the same 251 gradeable events in every panel. Axes are linear, cut off at 70 mm/s; 3.8% of points (85/2259) fall outside that window and are drawn open on the frame, skewed toward the fast events. Below the dashed $y = 0.70x$ the spike was missed, so the fraction above that line is recall@70; between the lines it was caught by under-predicting, above $y = x$ by over-predicting. Panels run in slip-frame MAE rank order. Top: the shared distribution of measured peaks, on its own axis.

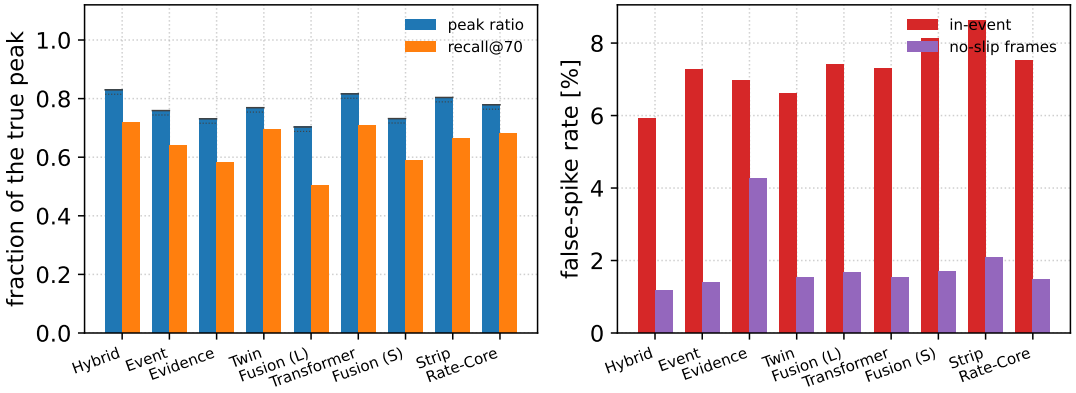


Figure 6.15: Left: peak ratio and recall@70. Right: false-spike rate, in-event and on the no-slip majority. Networks ordered by slip-frame MAE rank, so a disagreement with Figure 6.1 is visible directly.

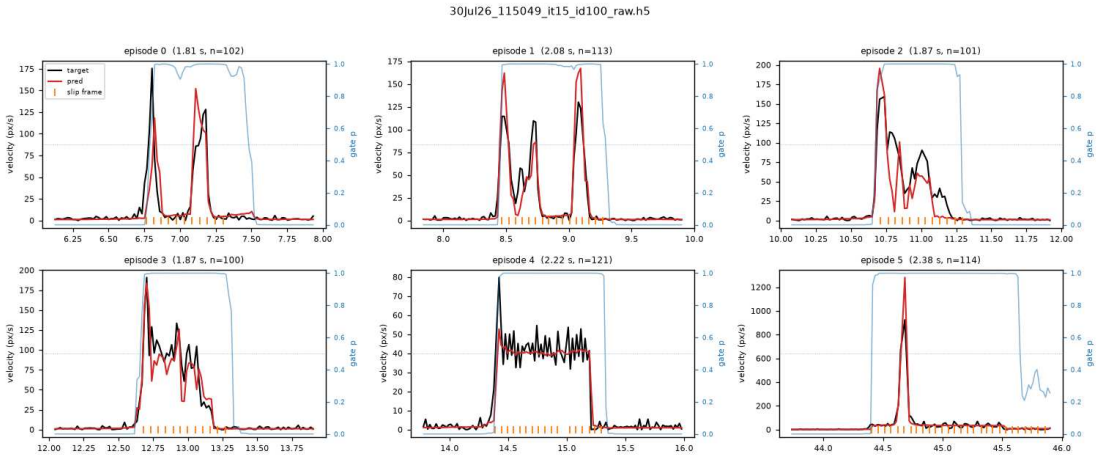


Figure 6.16: Example ground truth slip velocities(black) overlayed with predicted velocity(red) as inferred by the `Transformer` architecture, with the gate probability(blue) overlaid. The red markers along the x-axis show when GelSight frames occur. Note the slip spikes as well as the oscillations during the slip event. What can also be observed is that `Transformer` smoothes over dynamic slip oscillations.

6.10 Future Velocity Prediction

As explained in Section 5.4.3, the `Hybrid` architecture was trained with time-shifted velocity labels to see if predicting future slip is possible. Δ was chosen as $50\,\text{ms}$, $100\,\text{ms}$, and $200\,\text{ms}$. The results can be seen in Table 6.8 and Figure 6.17.

Onset detection survives remarkably far: gate AUROC is 0.998 and gate accuracy is $98.1\,\%$ at $100\,\text{ms}$, essentially matching the $\Delta = 0$ gate. Only after, at $200\,\text{ms}$, does performance reduce drastically.

The slip-velocity error also started off with a surprising accuracy. At $\Delta = 50\,\text{ms}$, the slip MAE is 3.350 mm/s, 57% greater than at $\Delta = 0\,\text{ms}$, but still less than the baseline and the $5\,\text{mm/s}$ reported by Chen et al. [13]. However, as expected, for higher deltas the MAE breaks down towards the baseline.

The onset spike showed the largest performance degradation, as would be expected. While some information about future spikes can be inferred from the shear force extracted from the visuotactile image, this approach quickly breaks down after $\Delta > 200\,\text{ms}$.

The results are very promising. Given the importance of being able to proactively react to slip, these results illustrate a clear path towards using the predictive power of the architectures to enable slip velocity control.

	test MAE					onset peak		false spike	gate	gating
Δ [ms]	all	slip	no-slip [mm/s]	persistence	skill	ratio	recall@70	no-slip [%]	AUROC	ΔMAE [%]
0	0.77	2.14	0.23	—	—	0.824	0.721	1.4	0.999	$+11.7$
50	1.27	3.35	0.46	1.71	0.253	0.733	0.562	3.4	0.999	-2.3
100	1.56	3.84	0.67	2.19	0.288	0.682	0.478	5.3	0.998	-9.8
200	2.09	4.25	1.25	2.77	0.245	0.562	0.274	15.3	0.987	-17.4

Table 6.8: Results from the `Hybrid` model future slip prediction, showing that for short timeframes, slip velocity prediction is feasible.

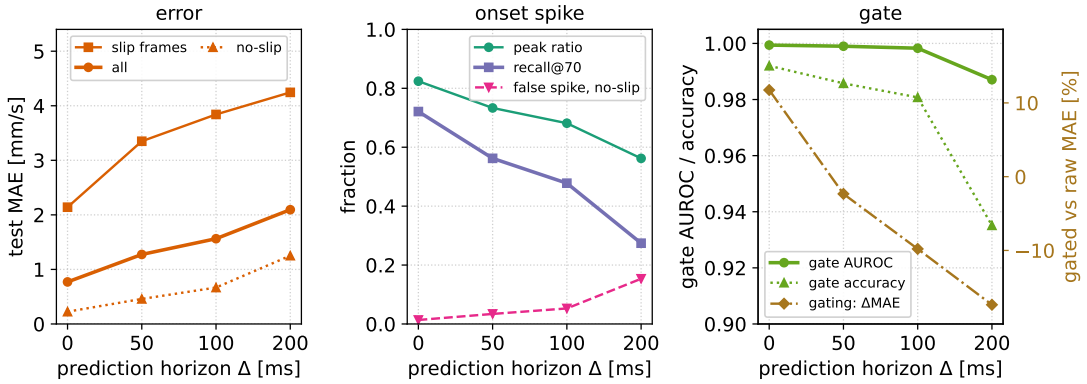


Figure 6.17: Results from the `Hybrid` model future slip prediction. Left: MAE as a function of prediction horizon Δ. Middle: Spike prediction performance. Right: Slip detection performance.

6.11 Limitations

The results presented in this chapter come with a few limitations worth noting. The first is that this thesis does not look at two important variances: material variance and rotation variance. Regarding material and shape, the current networks are only trained on the plastic test object. This has a well textured yet flat surface, which is not representative of the real world. Pulling direction is the second, with the test bench only pulling the object along the GelSights axis, which is also not indicative of the real world. The vibration sensor should be rotation invariant, since for the vibration signal the three axis are added, but for the visuotactile signal this makes a difference. Currently the ablation study shows that the vibration data is more important than the visuotactile data. If those two variances were introduced, that might very well change, together with the amount of data needed for training, and how big the architectures would have to be.

Another real world consideration is also noise. The current testbench is equipped with dampening to isolate it. And even though the second vibration sensor used for noise reduction can also be used, the effect of movement on the usefulness of the vibration data stream would have to be researched.

Another one is that the current test bench does not detect incipient slip, and as such the neural networks are not trained to detect it. Incipient slip detection is very useful

for slip-control, as it allows the controller to prime itself to react to the soon following slip. None of these limitations invalidate the comparison between architectures, but each should be addressed before the networks are deployed outside the test bench.

There are a few methodological shortcomings and limitations that have to be called out. The first is that the ablation study was done on a smaller test corpus, and while it suffices to make the point that vibration sensing is important, this prevents it from being directly comparable to the other results. In addition, this prevented the later designed `Evidence` architecture, which focus was sensor stream disappearing, from being included in the ablation, as the old test corpus does not exist anymore in a form that would have allowed the test to be comparable.

Regarding repeatability, all runs seen in the results were run with the same seed. The `Hybrid`, `Strip`, and `Transformer` were run a second time with different seeds, and the run-to-run difference was below 1% in every case. Nevertheless, all networks would have to be trained multiple times with different seeds to be able to make a stronger claim about their relative performance.

Latency also showed some inconsistencies, with the figures presented already presenting average over three runs. Nevertheless, the latency numbers varied by up to 11 %, when rerun later. A choice was made to include the latency figures that were generated while running the associated tests, though this means that comparing cross test latency has to be done carefully.

Which leads into the last point: with a 17 % total spread, a 0.61 mm/s noise floor in the velocity data, the difference between the different implementations is too small to make a definite claim about the superiority of any one architecture. Variations in training data, testing, or the above mentioned changes to the testing methodology could produce results that change the relative ordering of the architectures.

Despite these limitations, the range of tests and different architectures still show that slip velocity estimation is possible, and that vibration sensing is important for it, which are the main points this thesis is trying to make.

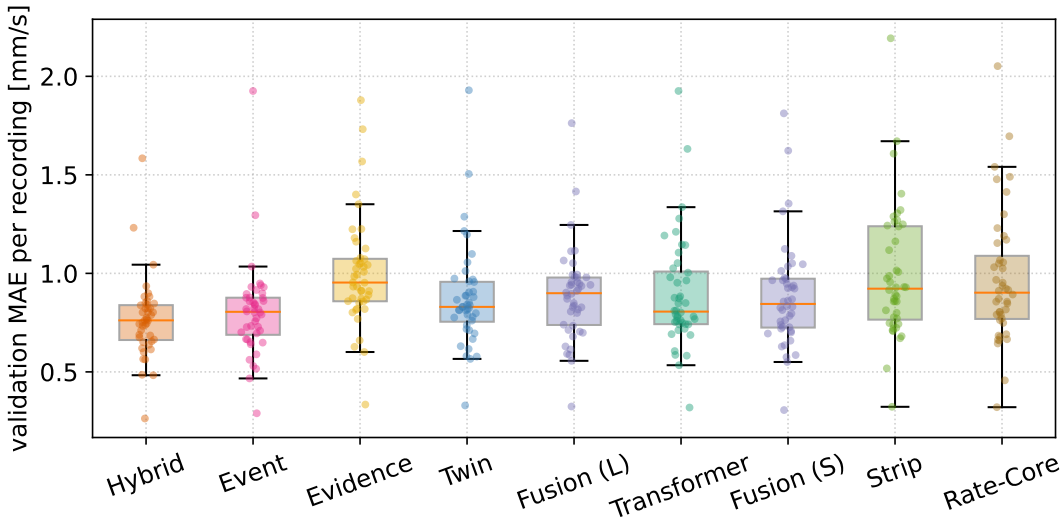


Figure 6.18: Validation MAE per recording, at each network's selected checkpoint. Between-recording spread is the evidence behind the noise band of Figure 6.2.

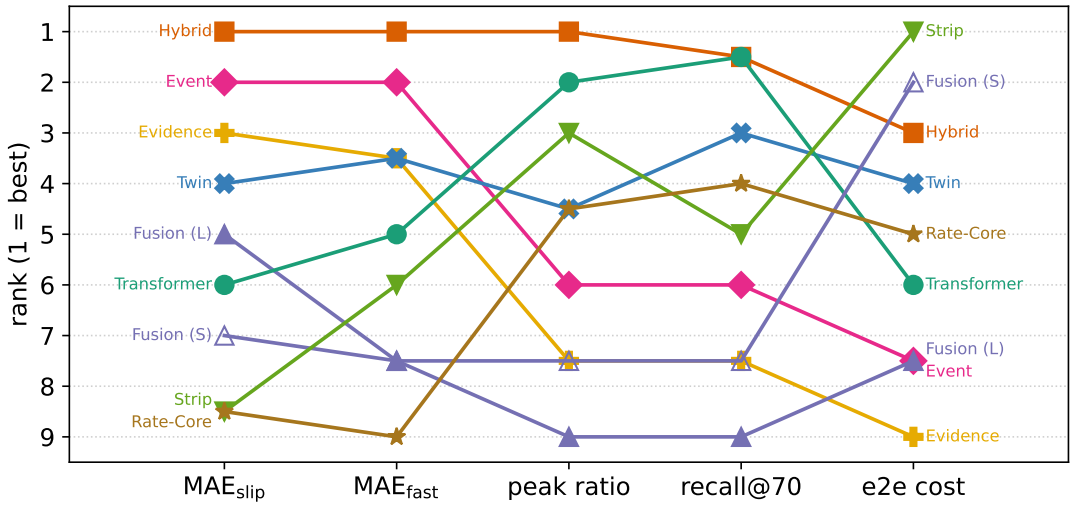


Figure 6.19: Each network's rank (1 = best) on five metrics — slip-frame MAE, fast-slip MAE, peak ratio, recall@70 and end-to-end cost.

7 Conclusion

With well-labeled data from a purpose-built test bench, slip velocity is inferable from visuotactile and vibration signals to within an absolute error of 2.13 mm/s, achieved by a Hybrid architecture that combines classical MLP-based late sensor fusion with a transformer trunk, at a latency within 6 ms of the fastest of the eight architectures evaluated.

This thesis presented the road to designing an architecture that is capable of slip-velocity estimation, starting with understanding what slip is on a physical level, and analyzing the importance of vibrations in tribology. Further, this thesis analyzes and evaluates different sensors, settling on a combination of a GelSight Mini visuotactile sensor and ADXL355 vibration sensor for their ability to capture both contact forces and slip-vibrations. It goes on further to look at different approaches for slip-detection and velocity estimation, settling on a transformer-based approach. Starting with data collection, a test bench to produce high-quality labeled slip-velocity data across different surfaces is used. In the methodology, eight different neural network architectures, three base architectures and five themed architectures, are designed in an attempt to solve specific subproblems. Finally, the architectures are evaluated, with the result that a Hybrid architecture that combines classic MLP-based late sensor fusion with a transformer trunk performed the best in terms of accuracy, and a specialized small neural network best in terms of inference time.

7.1 Core Learnings

Slip velocity is continuously inferable from sensor fusion. The central claim of this thesis, that slip velocity can be estimated as a continuous quantity, is confirmed. The best architecture, `Hybrid` achieves a mean absolute error of 2.13 mm/s during slip at sub 10ms latencies.

Vibration sensing carries most of the slip-velocity information. Visuotactile sensing carries most of the slip-detection information. Removing the accelerometer stream degrades

slip-frame MAE by 20–21 % across the main architectures, while removing the GelSight stream only costs 9–16 %.

Architecture is the most important parameter. There is no single parameter that correlates with performance. Architectures for slip velocity control have to be designed with tradeoffs and the data in mind.

Latency and performance is a tradeoff. The lowest latency models where not the best performing. The difference in MAE was less than 15%, for an more than 80% gain in latency, but that is a tradeoff that has to be made in accordance with the use cases needs.

Grey-box priors are outperformed by black-box architectures. Two architectures that embedded domain knowledge, Event, which explicitly models the velocity spike as a Gaussian transient, and Twin, which uses cross-modal self-supervised pretraining, were both outperformed by architectures that learn the mapping end to end. Slip physics are complex enough that the simplifications required to build an explicit model introduce more error than they remove.

The slip-onset velocity spike is the hardest single event to predict. At the moment of slip onset a sharp velocity spike occurs, on average within 36 ms of the event start. All architectures systematically underpredict its magnitude, because committing to a spike prediction in the wrong location incurs a heavy training loss.

Future Slip Velocity is predicable to an extend. Predicting slip 50ms into the future performed 57% worse than non-predictive force estimation, considerable, but usuable when taking into consideration that the 3.35mm/s MAE during slip events this yielded is still usuable as shown by Chen et al. [13].

7.2 Discussion

Slip is difficult. As such it might seem like it is a good idea to use a grey box approach, providing a neural network with a model, to better understand the problem and simplify learning. Both Twin (Section 5.5.3) and Event (Section 5.5.4) did this: the former implicitly through unsupervised pre-training, and the latter explicitly, by providing a model of the velocity spike. Both were outperformed by black box architectures.

Next, it might seem sensible to use an architecture with many parameters for a complex problem like slip. But as shown in Section 6.6 this also proved to not be true. And, as slip velocity does not seem extractable from any one representation, it might seem prudent

to help inference by inputting as many representations as possible. But as the ablation study and Strip (Section 5.5.2) showed: while some representations like marker velocity provide a marginal improvement, others like depth degrade performance, and raw sensor data is enough.

What is needed is a high frequency data source to capture the slip vibrations, and what provides the most performance is good architecture design: in the case of this thesis a mixture of classical late MLP-based sensor fusion and a transformer architecture.

And lastly while it seems like a very difficult problem, predicting future velocity is possible, with a performance penalty. But seeing how effective predictive control is, this could be the way forward towards slip velocity control.

7.3 Outlook

One of the main motivations for this research was providing a basis for slip-velocity control. Parallel work on this topic showed that slip-velocity control, where a gripper adjusts its grip strength to let an object slip with a set velocity, requires very tight, precise, and low latency feedback about the object's slip velocity.

Future work would adopt the insights made in this thesis to construct a system to provide exactly that, combining inference about the current speed with predictions about future velocity changes to enable slip-velocity control, which in turn could be used for more complex manipulation tasks.

Future work would also address the limitations that were presented: adding rotation variance and different materials to better represent real-world use cases, researching how networks trained on test bench data transfer to robot arms, and how different architectures could be merged to provide an efficient, low latency, and precise architecture.

That being said, this thesis showed what it set out to test: that with vibration and visuotactile sensing, slip velocity can be inferred.

Bibliography

[1] Ali Ajoudani et al. “Reflex control of the pisa/iit softhand during object slippage”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1972–1979. URL: https://doi.org/10.1109/ICRA.2016.7487326.

[2] Adnan Akay. “Acoustics of friction”. In: *The Journal of the Acoustical Society of America* 111.4 (2002), pp. 1525–1548. DOI: 10.1121/1.1456514.

[3] Analog Devices, Inc. *ADXL354/ADXL355 Data Sheet: Low Noise, Low Drift, Low Power, 3-Axis MEMS Accelerometers*. Rev. A. 2018. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_adxl355.pdf.

[4] Anthropic. *Claude*. Was used for code and writing assistance. 2026. URL: https://claude.ai/.

[5] Anne-Sophie Augurelle et al. “Importance of cutaneous feedback in maintaining a secure grip during manipulation of hand-held objects”. In: *Journal of Neurophysiology* 89.2 (2003), pp. 665–671. DOI: 10.1152/jn.00249.2002.

[6] Murana Awad et al. “Novel model-based force and object slip estimation approach for neuromorphic vision tactile sensors”. In: *Robotica* 44 (2026), pp. 210–230. DOI: 10.1017/S0263574726103130.

[7] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In: *arXiv preprint arXiv:1803.01271* (2018).

[8] Tristan Baumberger and Christiane Caroli. “Solid friction from stick–slip down to pinning and aging”. In: *Advances in Physics* 55.3-4 (2006), pp. 279–348. DOI: 10.1080/00018730600732186.

[9] Oded Ben-David, Gil Cohen, and Jay Fineberg. “The Dynamics of the Onset of Frictional Slip”. In: *Science* 330.6001 (2010), pp. 211–214. DOI: 10.1126/science.1194777. eprint: https://www.science.org/doi/pdf/10.1126/science.1194777. URL: https://www.science.org/doi/abs/10.1126/science.1194777.

[10] F. P. Bowden and D. Tabor. *The Friction and Lubrication of Solids*. Oxford University Press, 2001.

[11] Yu Cao et al. “Medical Image Retrieval: A Multimodal Approach”. In: *Cancer Informatics* 13.Suppl 3 (2014), p. 125.

[12] Wei Chen et al. “Tactile Sensors for Friction Estimation and Incipient Slip Detection—Toward Dexterous Robotic Manipulation: A Review”. In: *IEEE Sensors Journal* 18.22 (2018), pp. 9049–9064. DOI: 10.1109/JSEN.2018.2868340.

[13] Yuan Chen et al. “Tactile Velocity Estimation for Controlled In-Grasp Sliding”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1614–1621. DOI: 10.1109/LRA.2021.3058931.

[14] Cheng Chi et al. *Universal Manipulation Interface: In-The-Wild Robot Teaching Without In-The-Wild Robots*. 2024. arXiv: 2402.10329 [cs.RO]. URL: https://arxiv.org/abs/2402.10329.

[15] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[16] Christine A. Curcio et al. “Human photoreceptor topography”. In: *The Journal of Comparative Neurology* 292.4 (1990), pp. 497–523. DOI: 10.1002/cne.902920402.

[17] Antonino D’Alessandro and Giuseppe D’Anna. “Suitability of Low-Cost Three-Axis MEMS Accelerometers in Strong-Motion Seismology: Tests on the LIS331DLH (iPhone) Accelerometer”. In: *Bulletin of the Seismological Society of America* 103.5 (2013), pp. 2906–2913. DOI: 10.1785/0120120287.

[18] Antonino D’Alessandro, Salvatore Scudero, and Giovanni Vitale. “A Review of the Capacitive MEMS for Seismology”. In: *Sensors* 19.14 (2019), p. 3093. DOI: 10.3390/s19143093.

[19] James H. Dieterich. “Modeling of rock friction: 1. Experimental results and constitutive equations”. In: *Journal of Geophysical Research: Solid Earth* 84.B5 (1979), pp. 2161–2168. DOI: 10.1029/JB084iB05p02161.

[20] Siyuan Dong, Wenzhen Yuan, and Edward H. Adelson. “Improved GelSight tactile sensor for measuring geometry and slip”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 137–144. DOI: 10.1109/IROS.2017.8202149.

[21] Siyuan Dong et al. “Maintaining Grasps within Slipping Bounds by Monitoring Incipient Slip”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 3818–3824. DOI: 10.1109/ICRA.2019.8793538.

[22] David Dornfeld and Christopher Handy. “Slip detection using acoustic emission signal analysis”. In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. Raleigh, NC, USA, 1987, pp. 1868–1875.

[23] Christoph Feichtenhofer et al. “SlowFast Networks for Video Recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6202–6211.

[24] Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Vol. I, Ch. 12: Characteristics of Force*. URL: https://www.feynmanlectures.caltech.edu/I_12.html (visited on 05/19/2026).

[25] Jeremy A. Fishel, Veronica J. Santos, and Gerald E. Loeb. “A robust micro-vibration sensor for biomimetic fingertips”. In: *Proc. IEEE RAS & EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*. 2008.

[26] Maria Teresa Francomano, Dino Accoto, and Eugenio Guglielmelli. “Artificial sense of slip—a review”. In: *IEEE Sensors Journal* 13.7 (2013), pp. 2489–2498.

[27] Isao Fujimoto, Yoji Yamada, and Takashi Maeno. “Development of artificial finger skin to detect incipient slip for realization of static friction sensation”. In: *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. 2003.

[28] Niklas Funk et al. *Evetac: An Event-based Optical Tactile Sensor for Robotic Manipulation*. 2023. arXiv: 2312.01236 [cs.RO].

[29] Niklas Funk et al. “Evetac: An Event-based Optical Tactile Sensor for Robotic Manipulation”. In: *arXiv preprint arXiv:2312.01236* (2023).

[30] Junli Gao et al. *Visuo-Tactile-Based Slip Detection Using a Multi-Scale Temporal Convolution Network*. 2023. DOI: 10.48550/arXiv.2302.13564. arXiv: 2302.13564 [cs.RO].

[31] Yuri Gloumakov, Tae Myung Huh, and Hannah S. Stuart. “Fast In-Hand Slip Control on Unfeatured Objects With Programmable Tactile Sensing”. In: *IEEE Robotics and Automation Letters* 9.7 (2024), pp. 6059–6066. DOI: 10.1109/LRA.2024.3400148.

[32] Dorian A. H. Hanaor, Yixiang Gan, and Itai Einav. “Static friction at fractal interfaces”. In: *Tribology International* 93 (2016), pp. 229–238. DOI: 10.1016/j.triboint.2015.09.016.

[33] Simon Haykin. *Adaptive Filter Theory*. 5th. Pearson, 2014.

[34] Erik Helmut et al. “Actuated Version of the Universal Manipulation Interface Gripper”. In: 2024. URL: https://actuated-umi.github.io/.

[35] François Heslot et al. “Creep, stick-slip, and dry-friction dynamics: Experiments and a heuristic model”. In: *Physical Review E* 49.6 (1994), pp. 4973–4988. DOI: 10.1103/PhysRevE.49.4973.

[36] Barrett Heyneman and Mark R. Cutkosky. “Slip classification for dynamic tactile array sensors”. In: *The International Journal of Robotics Research* 35.4 (2016). DOI: 10.1177/0278364914564703.

[37] Carolina Higuera et al. *Tactile Beyond Pixels: Multisensory Touch Representations for Robot Manipulation*. 2025. arXiv: 2506.14754. URL: https://arxiv.org/abs/2506.14754.

[38] Geoffrey E. Hinton. “Training Products of Experts by Minimizing Contrastive Divergence”. In: *Neural Computation* 14.8 (2002), pp. 1771–1800. DOI: 10.1162/089976602760128018.

[39] E.G.M. Holweg et al. “Slip detection by tactile sensors: algorithms and experimental results”. In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 4. 1996, 3234–3239 vol.4. DOI: 10.1109/ROBOT.1996.509205.

[40] Max Horn et al. “Set Functions for Time Series”. In: *arXiv preprint arXiv:1909.12064* (2020).

[41] R. D. Howe and M. R. Cutkosky. “Sensing skin acceleration for slip and texture perception”. In: *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*. 1989, pp. 145–150. URL: https://doi.org/10.1109/ROBOT.1989.100011.

[42] Robert D. Howe and Mark R. Cutkosky. “Dynamic tactile sensing: perception of fine surface features with stress rate sensing”. In: *IEEE Transactions on Robotics and Automation* 9.2 (1993), pp. 140–151.

[43] Xiaohai Hu et al. *Learning to Detect Slip through Tactile Estimation of the Contact Force Field and its Entropy*. 2024. arXiv: 2303.00935.

[44] Ian M. Hutchings. "Leonardo da Vinci's studies of friction". In: *Wear* 360.Supplement C (2016), pp. 51–66. DOI: 10.1016/j.wear.2016.04.019. URL: https://doi.org/10.1016/j.wear.2016.04.019.

[45] Raouf A. Ibrahim. "Friction-induced vibration, chatter, squeal, and chaos—Part II: Dynamics and modeling". In: *Applied Mechanics Reviews* 47.7 (1994), pp. 227–253. DOI: 10.1115/1.3111080.

[46] Homer Jacobson. "The informational capacity of the human eye". In: *Science* 113.2933 (1951), pp. 292–293. DOI: 10.1126/science.113.2933.292.

[47] Andrew Jaegle et al. "Perceiver IO: A General Architecture for Structured Inputs & Outputs". In: *arXiv preprint arXiv:2107.14795* (2021).

[48] Jasper W. James, Nicholas Pestell, and Nathan F. Lepora. "Slip detection with a biomimetic tactile sensor". In: *IEEE Robotics and Automation Letters* 3.4 (2018).

[49] Neel Jawale et al. "Learned Slip-Detection-Severity Framework Using Tactile Deformation Field Feedback for Robotic Manipulation". In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. 2024. DOI: 10.1109/IROS58592.2024.10802687.

[50] Roland S Johansson and Göran Westling. "Signals in tactile afferents from the fingers eliciting adaptive motor responses during precision grip". In: *Experimental Brain Research* 66.1 (1987), pp. 141–154. DOI: 10.1007/BF00236210.

[51] Roland S. Johansson and Åke B. Vallbo. "Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin". In: *The Journal of Physiology* 286.1 (1979), pp. 283–300. DOI: 10.1113/jphysiol.1979.sp012619.

[52] Roland S. Johansson and Göran Westling. "Roles of glabrous skin receptors and sensorimotor memory in automatic control of precision grip when lifting rougher or more slippery objects". In: *Experimental Brain Research* 56.3 (1984), pp. 550–564. DOI: 10.1007/BF00237997.

[53] Micah K. Johnson and Edward H. Adelson. "Retrographic sensing for the measurement of surface texture and shape". In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. Miami, FL, USA, June 2009, pp. 1070–1077.

[54] H Peter Jost. *Lubrication (Tribology) Education and Research: A Report on the Present Position and Industry's Needs*. Tech. rep. Department of Education and Science, HMSO, London, 1966.

[55] Samira Ebrahimi Kahou et al. "Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video". In: *Proceedings of the 15th ACM International Conference on Multimodal Interaction*. 2013, pp. 543–550.

[56] Seyed Mehran Kazemi et al. "Time2Vec: Learning a Vector Representation of Time". In: *arXiv preprint arXiv:1907.05321* (2019).

[57] Gyeongwon Kim and Donghyun Hwang. "BaroTac: barometric three-axis tactile sensor with slip detection capability". In: *Sensors* 23.1 (2022), p. 428.

[58] Sung Joon Kim et al. "A Non-Array Type Cut to Shape Soft Slip Detection Sensor Applicable to Arbitrary Surface". In: *Sensors* 20.21 (2020), p. 6185. DOI: 10.3390/s20216185.

[59] Nathan M. Kinkaid, Oliver M. O'Reilly, and Panayiotis Papadopoulos. "Automotive disc brake squeal". In: *Journal of Sound and Vibration* 267.1 (2003), pp. 105–166. DOI: 10.1016/S0022-460X(02)01573-0.

[60] Kristin Koch et al. "How much the eye tells the brain". In: *Current Biology* 16.14 (2006), pp. 1428–1434. DOI: 10.1016/j.cub.2006.05.056.

[61] Jan J. Koenderink. "Optic flow". In: *Vision Research* 26.1 (1986), pp. 161–179. DOI: 10.1016/0042-6989(86)90078-7.

[62] Jan J. Koenderink and Andrea J. van Doorn. "Affine structure from motion". In: *Journal of the Optical Society of America A* 8.2 (1991), pp. 377–385. DOI: 10.1364/JOSAA.8.000377.

[63] Jan J. Koenderink and Andrea J. van Doorn. "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer". In: *Optica Acta: International Journal of Optics* 22.9 (1975), pp. 773–791. DOI: 10.1080/713819112.

[64] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. "Using Inertial Sensors for Position and Orientation Estimation". In: *Foundations and Trends in Signal Processing* 11.1-2 (2017), pp. 1–153. DOI: 10.1561/2000000094.

[65] Kenneth J. Kokjer. "The information capacity of the human fingertip". In: *IEEE Transactions on Systems, Man, and Cybernetics* 17.1 (1987), pp. 100–102. DOI: 10.1109/TSMC.1987.289337.

[66] Sergey Krylov and Joost Frenken. “The physics of atomic-scale friction: Basic considerations and open questions (Phys. Status Solidi B 4/2014)”. In: *physica status solidi (b)* 251 (Apr. 2014). DOI: 10.1002/pssb.201350154.

[67] Ian Lenz, Honglak Lee, and Ashutosh Saxena. “Deep Learning for Detecting Robotic Grasps”. In: *International Journal of Robotics Research* 34.4-5 (2015), pp. 705–724.

[68] Jianhua Li, Siyuan Dong, and Edward Adelson. “Slip Detection with Combined Tactile and Visual Information”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 7772–7777. DOI: 10.1109/ICRA.2018.8460495.

[69] Rui Li. “Touching is Believing : Sensing and Analyzing Touch Information with GelSight by”. In: 2015.

[70] Paul List et al. *Development of a Testbed for Benchmarking Tactile Slip Detection*. Advanced Design Project. Darmstadt, Germany: Technische Universität Darmstadt, Institute for Mechatronic Systems, Institute for Intelligent, and Autonomous Systems, July 2025.

[71] Siqi Liu et al. “Multimodal Neuroimaging Feature Learning for Multiclass Diagnosis of Alzheimer’s Disease”. In: *IEEE Transactions on Biomedical Engineering* 62.4 (2015), pp. 1132–1140. DOI: 10.1109/TBME.2014.2372011.

[72] Bruce D. Lucas and Takeo Kanade. “An iterative image registration technique with an application to stereo vision”. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI’81. Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[73] A. Majumdar and B. Bhushan. “Fractal model of elastic-plastic contact between rough surfaces”. In: *Journal of Tribology* 113.1 (1991), pp. 1–11.

[74] B. Mamat et al. “EveTac: An event-based optical tactile sensor for robotic manipulation”. In: *IEEE Robotics and Automation Letters* (2023).

[75] Héctor P. Martínez and Georgios N. Yannakakis. “Deep Multimodal Fusion”. In: *Proceedings of the 16th International Conference on Multimodal Interaction*. 2014, pp. 34–41.

[76] Michael E. McIntyre and Jim Woodhouse. “On the fundamentals of bowed-string dynamics”. In: *Acustica* 43.2 (1979), pp. 93–108.

[77] Claudio Melchiorri. “Slip detection and control using tactile and force sensors”. In: *IEEE/ASME Transactions on Mechatronics* 5.3 (2000), pp. 235–243. URL: https://doi.org/10.1109/3516.870052.

[78] Mariacarla Memeo et al. “Blind people can actively manipulate virtual objects with a novel tactile device”. In: *Scientific Reports* 13.1 (2023), p. 22845. DOI: 10.1038/s41598-023-49507-1.

[79] Raymond D. Mindlin. “Compliance of Elastic Bodies in Contact”. In: *Journal of Applied Mechanics* 16.3 (1949), pp. 259–268. DOI: 10.1115/1.4009973.

[80] Nobutomo Morita et al. “Grasping Force Control for a Robotic Hand by Slip Detection Using Developed Micro Laser Doppler Velocimeter”. In: *Sensors* 18.2 (2018), p. 326. DOI: 10.3390/s18020326.

[81] Arsha Nagrani et al. “Attention Bottlenecks for Multimodal Fusion”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14200–14213.

[82] Satoru Nasuno et al. “Time-resolved studies of stick-slip friction in sheared granular layers”. In: *Physical Review E* 58.2 (1998), pp. 2161–2171. DOI: 10.1103/PhysRevE.58.2161.

[83] Natalia Neverova et al. “ModDrop: Adaptive Multi-Modal Gesture Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.8 (2016), pp. 1692–1706.

[84] Jiquan Ngiam et al. “Multimodal Deep Learning”. In: *Proceedings of the 28th International Conference on Machine Learning (ICML)*. 2011, pp. 689–696.

[85] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation Learning with Contrastive Predictive Coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

[86] Aäron van den Oord et al. “WaveNet: A Generative Model for Raw Audio”. In: *arXiv preprint arXiv:1609.03499* (2016).

[87] A. Papangelo, M. Ciavarella, and J. R. Barber. “Shear-induced contact area anisotropy explains the directionality of transverse peeling of adhesive films”. In: *Physical Review E* 100.5 (2019), p. 053001.

[88] Ethan Perez et al. “FiLM: Visual Reasoning with a General Conditioning Layer”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018, pp. 3942–3951.

[89] Bo N. J. Persson. *Sliding friction*. 2011.

[90] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. “Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Utterance-Level Multimodal Sentiment Analysis”. In: *Proceedings of the Conference on Empirical Methods on Natural Language Processing*. 2015, pp. 2539–2544.

[91] Alec Radford et al. *Language Models are Unsupervised Multitask Learners*. Tech. rep. OpenAI, 2019.

[92] Dhanesh Ramachandram and Graham W. Taylor. "Deep Multimodal Learning: A Survey on Recent Advances and Trends". In: *IEEE Signal Processing Magazine* 34.6 (Nov. 2017), pp. 96–108. DOI: 10.1109/MSP.2017.2738401.

[93] Amin Rigi et al. "A Novel Event-Based Incipient Slip Detection Using Dynamic Active-Pixel Vision Sensor (DAVIS)". In: *Sensors* 18.2 (2018), p. 333. DOI: 10.3390/s18020333.

[94] ROBOTIS Co., Ltd. *XL430-W250 e-Manual*. https://emanual.robotis.com/docs/en/dxl/x/xl430-w250/. Accessed July 2026.

[95] Rocco Antonio Romeo and Loredana Zollo. "Methods and sensors for slip detection in robotics: a survey". In: *IEEE Access* 8 (2020).

[96] Shmuel M. Rubinstein, Gil Cohen, and Jay Fineberg. "Detachment fronts and the onset of dynamic friction". In: *Nature* 430.7003 (2004), pp. 1005–1009. DOI: 10.1038/nature02830.

[97] Shmuel M. Rubinstein, Gil Cohen, and Jay Fineberg. "Dynamics of Precursors to Frictional Sliding". In: *Physical Review Letters* 98.22 (2007), p. 226103. DOI: 10.1103/PhysRevLett.98.226103.

[98] Andy Ruina. "Slip instability and state variable friction laws". In: *Journal of Geophysical Research: Solid Earth* 88.B12 (1983), pp. 10359–10370. DOI: 10.1029/JB088iB12p10359.

[99] Nicu Sebe et al. *Machine Learning in Computer Vision*. Vol. 29. Dordrecht, The Netherlands: Springer, 2005.

[100] Jianbo Shi and Carlo Tomasi. "Good features to track". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.

[101] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *Advances in Neural Information Processing Systems*. 2014, pp. 568–576.

[102] STMicroelectronics. *LIS331DLH: MEMS digital output motion sensor, ultra low-power high performance 3-axis "nano" accelerometer, Data Sheet*. Rev. 3. 2009. URL: https://www.st.com/resource/en/datasheet/lis331dlh.pdf.

[103] Zhanfan Su et al. “Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor”. In: *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. 2015, pp. 297–303.

[104] Yaroslav Tenzer, Leif P. Jentoft, and Robert D. Howe. “The feel of MEMS barometers: inexpensive and easily customized tactile array sensors”. In: *IEEE Robotics & Automation Magazine* 21.3 (2014), pp. 89–95. DOI: 10.1109/MRA.2014.2310152.

[105] Seiichi Teshigawara et al. “High Sensitivity Slip Sensor Using Pressure Conductive Rubber”. In: *Proceedings of IEEE Sensors* (Oct. 2009), pp. 988–991. DOI: 10.1109/ICSENS.2009.5398213.

[106] David Titterton and John L. Weston. *Strapdown Inertial Navigation Technology*. 2nd ed. The Institution of Engineering and Technology, 2004. DOI: 10.1049/PBRA017E.

[107] Marc R. Tremblay and Mark R. Cutkosky. “Estimating friction using incipient slip sensing during a manipulation task”. In: *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. 1993, pp. 429–434.

[108] Abhinav Valada et al. “Deep Multispectral Semantic Scene Understanding of Forested Environments Using Multimodal Fusion”. In: *Proceedings of the International Symposium on Experimental Robotics (ISER)*. 2016, pp. 465–477.

[109] Giuseppe Vitrani and Michaël Wiertlewski. “Predicting Object Slippage in Robotic Grippers Using Human-Inspired Tactile Processing”. In: *ViTac 2024*. Delft University of Technology, Cognitive Robotics Department. 2024.

[110] Qiang Wang et al. “Robust Learning-Based Incipient Slip Detection Using the PapillArray Optical Tactile Sensor for Improved Robotic Gripping”. In: *IEEE Robotics and Automation Letters* 9.2 (2024), pp. 1827–1834. DOI: 10.1109/LRA.2023.3347141.

[111] Peter D. Welch. “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”. In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.

[112] Bernard Widrow et al. “Adaptive noise cancelling: Principles and applications”. In: *Proceedings of the IEEE* 63.12 (1975), pp. 1692–1716. DOI: 10.1109/PROC.1975.10036.

[113] Robert J Woodham. “Photometric method for determining surface orientation from multiple images”. In: *Optical engineering* 19.1 (1980), pp. 139–144.

[114] Jim Woodhouse and Paul M. Galluzzo. “The bowed string as we know it today”. In: *Acta Acustica united with Acustica* 90.4 (2004), pp. 579–589.

[115] Di Wu et al. “Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.8 (2016), pp. 1583–1597.

[116] Yang Yang, Corinne Dejous, and Hamida Hallil. “Trends and Applications of Surface and Bulk Acoustic Wave Devices: A Review”. In: *Micromachines* 14 (2022), p. 43. DOI: 10.3390/mi14010043.

[117] Jianzhu Yin et al. “Measuring Dynamic Shear Force and Vibration With a Bioinspired Tactile Sensor Skin”. In: *IEEE Sensors Journal* 18.9 (2018), pp. 3544–3553. DOI: 10.1109/JSEN.2018.2811407.

[118] Wenzhen Yuan, Siyuan Dong, and Edward H. Adelson. “GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force”. In: *Sensors* 17.12 (2017). ISSN: 1424-8220. DOI: 10.3390/s17122762. URL: https://www.mdpi.com/1424-8220/17/12/2762.

[119] Wenzhen Yuan et al. “Measurement of shear and slip with a GelSight tactile sensor”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 304–311. DOI: 10.1109/ICRA.2015.7139016.

[120] Andrea Zangrandi et al. “Neurophysiology of slip sensation and grip reaction: insights for hand prosthesis control of slippage”. In: *Journal of Neurophysiology* 126.2 (2021), pp. 477–492. DOI: 10.1152/jn.00087.2021.

[121] Yaonan Zhu et al. “Visual Tactile Sensor Based Force Estimation for Position-Force Teleoperation”. In: *2022 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. 2023, pp. 49–52. DOI: 10.1109/CBS55922.2023.10115342.