IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Ultra-Fast Lightweight Incipient Slip Detection Using Hyperdimensional Computing With the PapillArray Tactile Sensor

Jingtao Zhang, *Graduate Student Member, IEEE*, Yi Liu, *Graduate Student Member, IEEE*, Yanxun Lu, Stephen J. Redmond, *Senior Member, IEEE*, and Changhong Wang

***Abstract*—Timely detection of incipient slip is critical for delicate robotic grasping and dexterous manipulation. However, existing learning-based methods suffer from detection latency and high computational demands. In this paper, we present an ultra-fast lightweight incipient slip detection framework based on hyperdimensional (HD) computing, using the PapillArray optical tactile sensor. Our approach introduces a novel graphical-spatial-temporal HD encoding scheme coupled with a context-driven training and inference strategy, achieving a slip detection accuracy of 91.78% in offline evaluation. The resulting model is exceptionally compact and highly edge-compatible, with a size of only 0.375 kB. Furthermore, hardware acceleration on an FPGA enables inference within 0.42 microseconds, representing an over $10^4\times$ speedup compared to optimized CPU implementations. Online robotic experiments involving grip-force control based on the proposed slip detection method further validate its practical effectiveness. This work offers a practical and scalable solution for real-time slip detection in robotic manipulation tasks.**

***Index Terms*—Force and tactile sensing, hardware-software integration in robotics, machine learning for robot control.**

I. Introduction

THE ability to detect incipient slip – the initial stage of relative motion between a robotic gripper and a grasped object – is essential for achieving secure and dexterous manipulation [1]. Timely detection of such early slip events enables robotic systems to perform corrective actions, preventing object drops and enhancing overall manipulation stability. To achieve this, advanced tactile sensors play a crucial role by providing rich spatiotemporal information, enabling robots to perceive subtle changes at the object-surface interface and address complex tactile perception tasks, including accurate slip detection [2].

Recent advances in learning-based slip detection have demonstrated high performance by leveraging the abundant data captured by sophisticated tactile sensors [3], [4]. However, many of these machine learning (ML) approaches – such as support vector machines (SVMs) [5], convolutional neural networks (CNNs) [6], and recurrent neural networks (RNNs) [7] – introduce substantial computational overhead. This complexity leads to increased detection latency, limiting their responsiveness during manipulation tasks [8]. Additionally, their memory and compute demands hinder deployment on resource-constrained robotic platforms, where lightweight and ultra-fast inference is critical [9].

To address the pressing need for efficient and responsive incipient slip detection, we propose a lightweight, ultra-fast framework based on hyperdimensional (HD) computing.[1] HD computing is a brain-inspired computational paradigm that enables efficient representation and reasoning for cognitive tasks by emulating properties of large-scale neural circuits [10]. By leveraging this paradigm, our framework enables early and efficient characterization of slip events on edge devices operating under strict latency and resource constraints. The key contributions of this work are:

- We develop a compact, edge-compatible incipient slip detection framework based on HD computing, designed for low-latency, high-efficiency deployment.
- We propose a novel graphical-spatial-temporal HD encoding scheme tailored to high-resolution tactile array data, capturing both local spatial deformation patterns and their temporal dynamics in a high-dimensional binary space.
- We introduce a context-driven training and inference strategy that exploits sequential tactile context to enhance decision-making under variable grasping conditions, improving generalization and robustness.
- We demonstrate a hardware-accelerated design that benefits from the compactness and binary nature of HD computing, enabling real-time slip detection suitable for integration into reactive control loops on edge devices.

Received 25 June 2025; accepted 6 December 2025. Date of publication 17 December 2025; date of current version 6 January 2026. This article was recommended for publication by Associate Editor P. Maiolino and Editor A. Banerjee upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 62303496, Grant 62573441, and Grant 62503339, in part by Shenzhen Medical Research Fund under Grant D250403003, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515011729, and in part by the Natural Science Foundation of Liaoning Province under Grant 2025-BS-0317. *(Corresponding author: Changhong Wang.)*

Jingtao Zhang, Yi Liu, Yanxun Lu, and Changhong Wang are with the School of Biomedical Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518000, China, and also with the Key Laboratory of Sensing Technology and Biomedical Instruments of Guangdong Province, Sun Yat-sen University, Shenzhen 518000, China (e-mail: zhangjt77@mail2.sysu.edu.cn; liuy2226@mail2.sysu.edu.cn; luyx58@mail2.sysu.edu.cn; wangchh55@mail.sysu.edu.cn).

Stephen J. Redmond is with the Insight Research Ireland Center for Data Analytics, University College Dublin, D04 C1P1 Dublin, Ireland, and also with University College Dublin, D04 C1P1 Dublin, Ireland (e-mail: stephen.redmond@ucd.ie).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3645655, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3645655

[1] https://github.com/JT2106/Hyperdimensional-computing-tactile-sensing

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

II. Related Work

A. Learning-Based Slip Detection

Leveraging advancements in machine learning and deep learning, learning-based approaches have demonstrate satisfactory accuracy and robustness on slip detection in robotic systems [4]. These methods typically involve training models on datasets of tactile sensor readings to identify patterns associated with slip events. One of the earliest works was accomplished by Hosoda et al. [11] in 2002 by training a neural network with vision and tactile data. Though it could not distinguish slip and pressure change, it gave a primitive representation of the slip phenomenon.

More recently, the GelSight camera-based tactile sensor has demonstrate effectiveness in slip detection. Yuan et al. [12] propose a method to analyze the contact condition of the GelSight sensor, which is able to determine the degree of slip. Dong et al. [13] additionally combined this strategy with CNN-based methods to successfully detect slip. Moreover, Li et al. [14] further combined a GelSight sensor with an external camera and applied a deep neural network method, combining CNN and RNN to improve the accuracy of slip detection. Similarly, Calandra et al. [15] also predicted slip when picking up daily objects using a deep learning approach with two GelSight sensors along with an RGB camera.

The TacTip sensor is another camera-based tactile sensor that has shown utility for slip detection. James et al. [16] first proposed a slip detection method by measuring the positions of internal pins embedded in the TacTip's compliant skin and employed SVM to analyze pin velocity data. In a later study, Taunyazov et al. [17] modified the TacTip sensor to create a traction differential and facilitate the occurrence of incipient slip. An external camera was used for data labeling, and incipient slip was detected using CNN-based methods.

Compared to camera-based tactile sensors, optical sensors with much less instrumental complexity, such as the PapillArray sensor used in our work, have also shown significant performance in slip detection. Khamis et al. [18] first developed a heuristic algorithm employing the PapillArray sensor for translational incipient slip detection. Martinez Ulloa et al. [19] further refined this heuristic approach to account for rotational slip conditions. Moreover, to improve the accuracy and robustness of incipient slip detection using the PapillArray sensor, Wang et al. [7] developed an integrated learning-based approach combining data annotation, augmentation, an RNN-based algorithm for classification.

While achieving excellent performance, these learning-based methods face limitations including computational complexity and latency, limiting their real-time applicability and deployment on resource-constrained robotic platforms. In contrast, our work introduces a lightweight and ultra-fast slip detection framework using HD computing to address these limitations.

B. Hyperdimensional Computing

HD computing is a brain-inspired computational framework that represents and manipulates information using high-dimensional binary vectors, known as hypervectors. It exploits key properties of high-dimensional spaces – such as sparsity, near-orthogonality, and robustness to noise – to perform efficient and distributed computation [10].

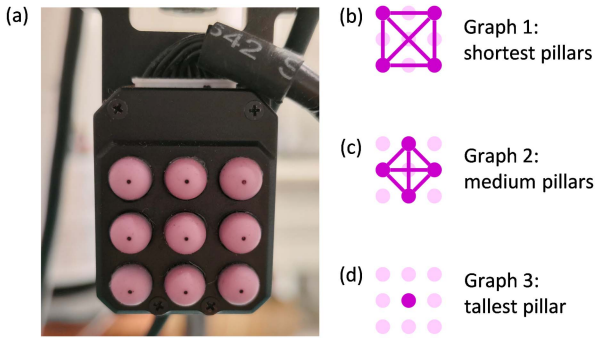

Fig. 1. (a) PapillArray tactile sensor. (b) Graph-1: formed by connecting the shortest pillars. (c) Graph-2: formed by connecting pillars with medium height. (d) Graph-3: single-node graph formed by the tallest central pillar.

HD computing was initially introduced for classification tasks, such as language recognition, computer vision, and human activity recognition [20], [21]. These tasks typically involve biosignal processing, such as electromyography (EMG) [22], electroencephalography (EEG) [23], electrocorticography (ECoG) [24], and photoplethysmography (PPG) [25], as summarized in [26].

Similar to many biosignals, the PapillArray sensor provides multi-channel analog signals, making it inherently suitable for HD computing. In our HD-based framework, we introduce a graphical-spatial-temporal HD encoding scheme specifically designed for multi-channel tactile array data. Additionally, we further propose a context-driven training and inference strategy that enhances generalization across diverse manipulation conditions.

III. Methods

A. Dataset and Preprocessing

We utilized an open-source dataset collected using a commercial PapillArray tactile sensor (Fig. 1(a)). The dataset collection process is described in detail in [7]. Briefly, the dataset was collected using a six-degree-of-freedom hexapod robot (H-820, Physik Instrumente) with the sensor mounted on its top. A transparent acrylic plate was fixed above the sensor, and a video camera (Logitech Streamcam) was positioned overhead to record the contact interface. The hexapod pushed the sensor vertically against the plate and then moved it laterally to induce slip. Lateral motion included translation, rotation, or a combination of both, across speeds ranging from 0.5 to 16 mm/s and varying compression depths. A total of 200 tactile sequences were recorded under various conditions, including different compression levels, velocities, and slip directions, covering diverse slip initiation conditions.

The PapillArray sensor records the 3D force and 3D displacement on each pillar at a sampling rate of 1000 Hz. Slip labels were originally assigned at the pillar level by tracking the tangential velocity of each pillar tip using video. A pillar was labeled as slipping if its velocity exceeded a predefined threshold. To obtain sensor-level labels, we aggregated pillar-level states as follows. *Non-slip:* no pillar has slipped. *Incipient slip:* at least one pillar has slipped, but not all. *Gross slip:* all pillars in contact have slipped. For rotational sequences, only the outer eight pillars were considered, as the central pillar remains stationary by design.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

To mitigate small-value glitch noise, sensor signals were downsampled to 100 Hz. Raw tactile readings were rounded to the nearest 2^{-3}, enabling fixed-point representation for hardware deployment. For feature extraction, the proportion of the magnitude of tangential displacement to the magnitude of total 3D displacement for each pillar was calculated. This produced a one-dimensional feature, ranging from [0, 1], which was used for subsequent HD computing operations.

For arrayed tactile sensors, a graph-based approach can be employed to capture the relationships between different channels. In this method, each sensor channel is represented as a node in a graph, and edges are established between nodes based on dependencies. This graph structure, when combined with the spatial information from the sensor array, provides a powerful representation for modeling the interactions and dependencies between individual sensor channels. In our implementation, we constructed three distinct graphs for the PapillArray sensor (Fig. 1(b)–(d)), where each graph connects pillars of the same height, thereby capturing the potential properties related to the inducement of incipient slip.

B. Incipient Slip Detection Based on HD Computing

This section introduces the proposed HD computing framework for incipient slip detection. The method maps spatiotemporal and graphical tactile data into high-dimensional representations, enabling rapid and efficient slip classification with minimal computational overhead. Tactile features are first encoded into unified hypervectors via HD mapping. A graphical–spatial–temporal HD (gstHD) encoder then captures local deformation patterns and their temporal evolution. Finally, a context-driven training and inference scheme leverages contextual cues for similarity-based classification. The following subsections detail each component.

1) HD Mapping: The HD mapping process involves projecting real-valued features into predefined hypervectors. Initially, the one-dimensional features were uniformly quantized into 21 discrete levels (i.e., {0, 0.05, 0.1, ..., 0.95, 1}). Each quantization level is associated with a unique hypervector stored in a continuous item memory (CIM). Specifically, two orthogonal hypervectors are generated for the minimum and maximum levels, while the hypervectors for intermediate levels are derived through linear interpolation between these two orthogonal hypervectors.

In addition to mapping real-valued features, domain information also needs to be encoded into hyperspace. Given that domains are finite and distinct, hypervectors representing these domains are randomly generated and ensured to be orthogonal. In our application, domain information includes nine pillar identities (denoted as {"p1", "p2", ..., "p9"}) of the PapillArray sensors, and the three graph identities (denoted as {"g1", "g2", "g3"}) of the constructed graphs. Consequently, we generate nine hypervectors for the pillar IDs and three hypervectors for the graph IDs, all of which are stored in an item memory (IM).

Furthermore, a context memory is utilized to map the tactile context into hyperspace, with each context represented by a unique hypervector. For our implementation, we empirically categorized the tactile context into four distinct levels based on the normal compression of the central pillar (measured in mm): [0, 0.75], (0.75, 1], (1, 1.125], (1.125, $+\infty$). These compression levels are defined by partitioning the real-valued intervals into four intervals, which is similar to quantized real-valued information. Thus, our context memory (CM) can be viewed as an alternative continuous item memory (CIM) with four hypervectors, generated through a process identical to that used for CIM. In this work, we set the dimensionality $D = 1000$. All predefined hypervectors above are bipolar $\{-1, +1\}^D$ vectors.

2) The Graphical-Spatial-Temporal HD Encoder: The gstHD encoder is a critical component of our incipient slip detection framework, integrating graphical, spatial, and temporal information to enhance the representation of tactile data (Fig. 2). For graphical encoding, each vertex v in the graph is mapped to a level vector $CIM(f_v)$ based on its feature value f_v. Edges are encoded by combining the hypervectors of their corresponding vertex pairs via binding (i.e., bitwise multiplication, denoted as $*$), and the entire graph is represented by bundling (i.e., bitwise addition, denoted as $+$) all vertex and edge hypervectors into a single hypervector Γ, as described in (1):

$$\Gamma = \sum_{v \in \text{vertices}} \text{CIM}(f_v) + \sum_{\substack{(p,q) \in \text{edges} \\ p \neq q}} \text{CIM}(f_p) * \text{CIM}(f_q). \quad (1)$$

To encode multiple graphs into a unified representation, each graph hypervector Γ_g is associated with a unique graph ID vector $IM(\Gamma_g)$. The final hypervector G is formed by binding each graph hypervector with its corresponding graph ID, as shown in (2):

$$G = \sum_{g \in graphs} \Gamma_g * IM(\Gamma_g). \quad (2)$$

Subsequently, spatial encoding integrates the positional information of the sensor pillars with the graphical data. For each pillar c, the level vector $CIM(f_c)$ is bound with its corresponding positional ID vector $IM(c)$. These pillar-bound spatial hypervectors are then bundled together along with the graphical-encoded hypervector G, as represented in (3):

$$GS = \left[G + \sum_{c \in \text{pillars}} CIM(f_c) * IM(c) \right], \quad (3)$$

where the square brackets [...] denote bipolarization.

To incorporate temporal dynamics, a sequence of graphical-spatial encoded hypervectors within a time interval of length N, denoted in superscripts as GS^t, GS^{t-1}, ..., is further encoded. In this work, temporal window length is empirically set to 0.15 s (i.e., $N = 15$, samples at 100 Hz). The temporal sequence is encoded using k-bit bitwise cyclical permutation (ρ^k), where k indicates the temporal displacement of the hypervector GS^{t-k} relative to the current time t. These permuted hypervectors are then bound together to form a unified temporal hypervector GST^t, as expressed in (4):

$$GST^t = \prod_{k=0}^{N-1} \rho^k GS^{t-k} \quad (4)$$

3) Context-Driven HD Training and Inference: The gstHD encoder serves for both training and inference stage (Fig. 3). During training, a context encoding process is performed by binding the encoded hypervector GST^t with its corresponding context hypervector CV^t, resulting in a context-encoded hypervector $EC^t = GST^t * CV^t$. Subsequently, a bipolar prototype hypervector for each class, denoted as P_k, is generated by computing the class centroid of all of the corresponding class:

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

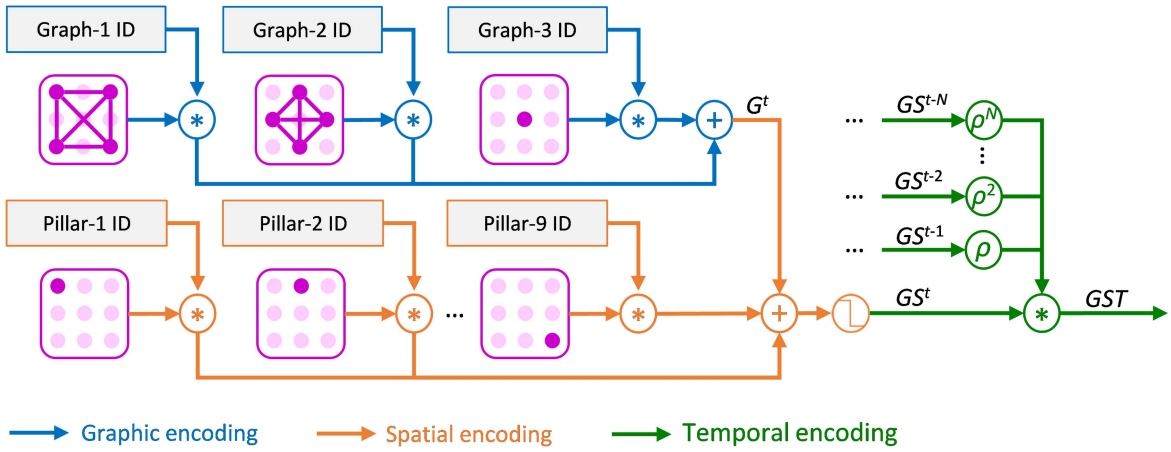

Fig. 2. The graphical-spatial-temporal HD encoder.

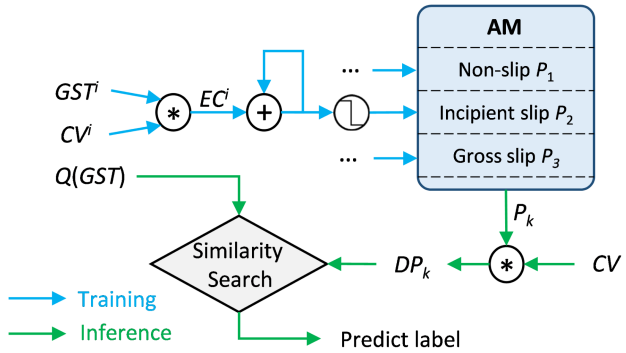

Fig. 3. Context-driven HD training and inference.

$P_k = [\sum_i EC^i]$. This training process is executed in a single pass over the training data. All prototype hypervectors are then stored in associative memory (AM). At inference time, each prototype hypervector in AM is bound with the query context vector CV^t, for "retrieving" context-specific prototypes, denoted as $DP_k = P_k * CV^t$. A Hamming distance check d_{Ham} is then performed between the encoded query vector $Q(GST^t)$ and all DP_k. The label of the prototype with the closest similarity to the query vector is returned as the predicted class (i.e., slip condition) $\hat{y}$, as expressed in (5):

$$\hat{y} = \arg\min_k d_{\text{Ham}}(Q, DP_k). \tag{5}$$

C. FPGA Implementation

FPGA are well-suited to HD computing acceleration due to their ability to efficiently handle large volumes of bitwise operations [27]. Fig. 4 illustrates the FPGA acceleration of the gstHD encoder and inference process. In this implementation, bipolar hypervectors are represented in binary from $\{0,1\}^D$ for efficient logit operation, and bitwise multiplication is replaced with *XNOR* logic. The process commences with tactile feature extraction and quantization fully in parallel, where quantized features can be subsequently mapped to hyperspace via a look-up table (A).

For HD operations, rather than employing full parallelism across all pillars, we balance computational resource utilization and latency through pipelining. Specifically, IM and CIM are stored in block RAM (BRAM), and the level and ID vectors for each graphical element or pillar are retrieved via an index counter (B). This well-designed index counter enables graphical-spatial HD encoding to be executed in a pipelined fashion, incorporating bitwise *XNOR* logic, accumulation and binarization (C). The resulting graphical-spatial encoded hypervectors GS are then forwarded to a predefined temporal register, where temporal HD encoding is achieved through cascaded premutation and *XNOR* operations (D). Overall, this gstHD encoder is structured in a pipeline architecture, which efficiently outputs the well-encoded hypervector GST with substantial acceleration.

For inference, an *XNOR* operation decodes the prototype P_k from the AM. The context-specific prototype DP_k then performs a similarity measurement with the query vector to for classification. This classification process is logically achieved by combining *XOR* logic with a tree adder (E). Given that our incipient slip detection task involves only three classes, this inference process is executed in parallel across all classes to ensure rapid classification.

D. Estimation of Ideal Grip Force Via Incipient Slip

As described in Section III-A, incipient slip refers to the condition where a portion of the contact interface between the sensor and the object begins to slip, while the remaining regions remain stationary, occurring prior to any observable relative motion (gross slip). This phenomenon, commonly observed in human grasping, enables fine-tuned grip-force regulation, whereby the applied force is just sufficient to prevent slippage and object drop [28].

Building upon this concept, the static coefficient of friction (μ_s) can be estimated from incipient slip events and subsequently used to derive the minimal grip force required to prevent slip under ideal conditions (referred to as the ideal grip force). This estimation facilitates quantitative assessment of grasp dexterity during "grasp-and-lift" operations by computing the ratio of the maximum applied grip force to the ideal grip force, which can circumstantially evaluate the effectiveness of the proposed HD-based incipient slip detection method.

In this study, a valid slip interval is empirically defined as a 0.6-second time window in which at least 50% of the timestamps are predicted as slip (either incipient or gross). Based on the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

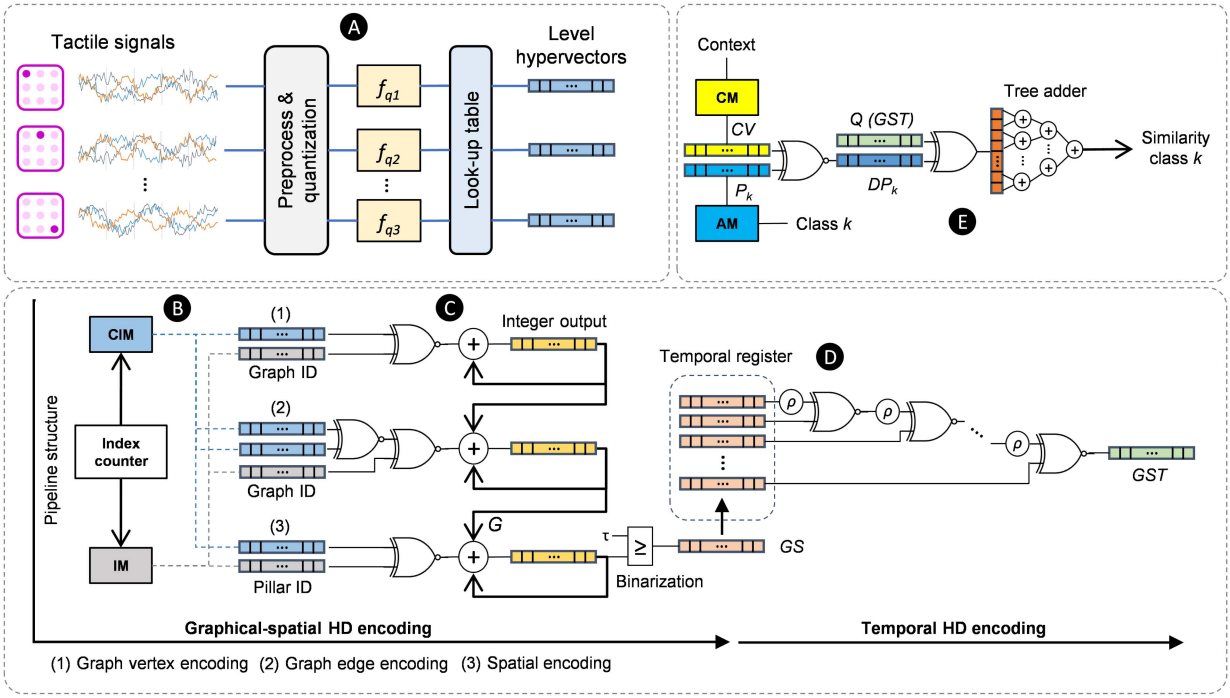

Fig. 4. FPGA implementation of the HD-based incipient slip detection.

Coulomb friction model, neglecting torque effects and microscopic variations. Hence, the local coefficient of friction μ_{sp} is estimated for each tactile pillar at incipient slip incident within valid slip intervals. The overall static coefficient of friction μ_s is then computed as the mean of all μ_{sp} values. Consequently, the ideal grip force for each instance is obtained by dividing the total tangential force across all pillars by μ_s. As multiple incipient slip events may occur within a single "grasp-and-lift" operation, the ultimate ideal grip force is determined as the minimum value among all estimated instances.

IV. Experiments and Results

A. Offline Incipient Slip Detection

All offline processing was conducted using MATLAB 2022b. The dataset was randomly partitioned into training and testing sets. To ensure balanced representation across different conditions, we employed a conditional bootstrapping method based on compression levels and velocities, iterating this process six times. One iteration was reserved for hyperparameter tuning using a training-validation subset, while the remaining five were used for performance evaluation.

Fig. 5 shows the offline performance of our HD-based incipient slip detection model. The model achieved an accuracy of 91.78% ± 0.53%, with its effectiveness of detection of slip event at early stages confirmed by the confusion matrix.

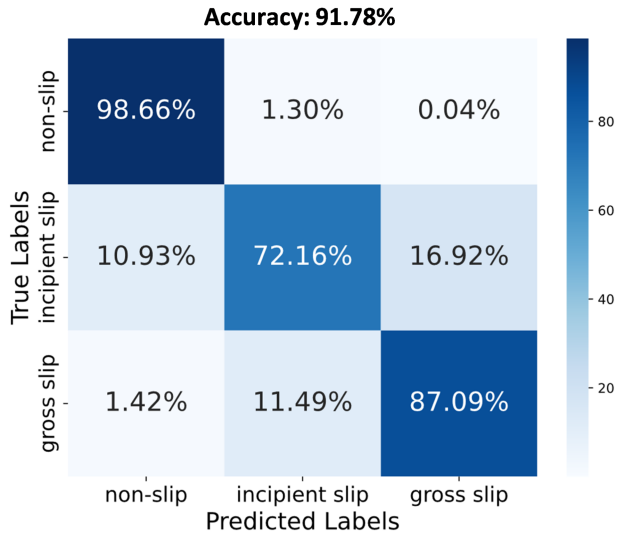

Fig. 5. Results of offline evaluation.

B. Ablation Study

In this ablation study, we denote our full model as CL-gstHDC, where "CL-" signifies the context-driven learning and "gstHDC" refers to our proposed gstHD encoder. To assess the contribution of the context-driven training and inference strategy, we constructed a baseline model using standard HD training and inference [20], while retaining the same encoder architecture. This ablation model is referred to as gstHDC, without the "CL-" prefix.

To further evaluate the impact of the gstHD encoder itself, we implemented two additional baseline models using more conventional HD encoders that have shown success in biosignal processing [26]: a purely spatial encoder (sHDC) and a spatial-temporal encoder (stHDC).

The results in Table I highlight the effectiveness of both the gstHD encoder and the context-driven approach. Specifically, the gstHDC model outperformed sHDC and stHDC by 4.86% and 3.44%, respectively. Furthermore, the proposed CL-gstHDC

TABLE I
ABLATION RESULTS

Model	sHDC	stHDC	gstHDC	CL-gstHDC
Accuracy	85.00% ±1.15%	86.42% ±1.35%	89.86% ±0.94%	91.78% ±0.53%

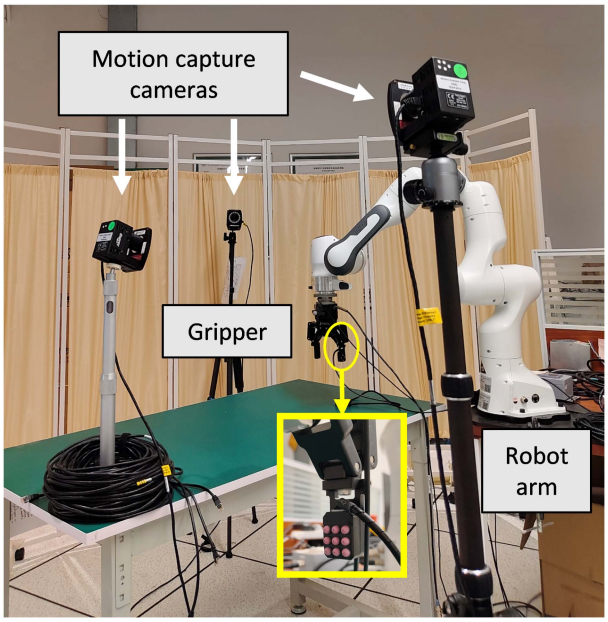

Fig. 6. Robotic set up for online evaluation.

model improved upon gstHDC by an additional 1.92%, confirming the advantage of context-driven learning.

C. Hardware Acceleration

We implemented our HD-based model on a Xilinx Artix-7 xc7a200tfbg484-2 FPGA using SystemVerilog HDL. Resource utilization and latency were analyzed using Vivado 2023.1 tools. The model achieved an inference latency of just 0.42 µs, while consuming 28.1% of the look-up tables (LUTs), 14.9% of the flip-flops (FFs), and 19.2% of the BRAM. The total model size is only 375 bytes, making it well-suited for deployment on resource-constrained edge platforms.

To benchmark the benefit of hardware acceleration, we also ran a software implementation of our model in C++ on an Intel Core i7-10700 CPU. The measured latency was 5985 µs, which is over 10^4 times slower than the FPGA hardware implementation.

D. Online Evaluation

For real-time validation, we deployed the trained model on an FPGA. Since tracking individual pillar slip states is infeasible during real-world grasping, we evaluated system effectiveness via real-time grip-force control based on incipient slip detection.

The robotic platform is illustrated in Fig. 6. A PapillArray sensor was mounted on one side of a modified Robotiq 2F-85 gripper, which was mounted on a Franka Emika Panda 7-DOF robotic arm. A Cortex v5 motion capture system with three infrared cameras was used to track object displacement. We

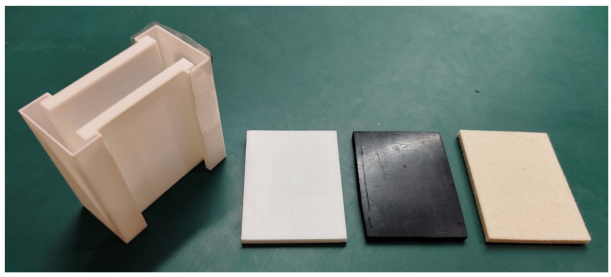

Fig. 7. From left to right: surface-replaceable 3D-printed box, PTFE surface, rubber surface and felt surface.

TABLE II
ONLINE EVALUATION RESULTS

Surface	Slip displacement		Max / ideal grip force
	Translational (mm)	Rotational (°)	
PTFE	3.56 ± 2.65	1.36 ± 2.18	1.48 ± 0.29
Rubber	1.34 ± 3.35	0.51 ± 0.22	1.36 ± 0.36
Felt	0.44 ± 0.41	0.64 ± 0.68	1.25 ± 0.27
Average	1.78 ± 2.74	0.84 ± 1.33	1.35 ± 0.31

used a custom 3D-printed thermoplastic cuboid box with interchangeable surface materials – PTFE (low-friction) rubber (high-friction), and felt (uneven, direction-dependent friction), as shown in Fig. 7.

During each trial, the robot grasped the object with a preset force of 1 N and lifted it vertically by 100 mm at a maximum speed of 5 mm/s and acceleration of 50 mm/s^2. A simple grip-force control mechanism increased the gripping force upon slip detection by moving the gripper fingers inward by 0.4 mm. A total of 20 test trials were completed, with 10 trials for each surface.

The effectiveness of the real-time grip-force control was evaluated using two complementary metrics: the relative displacement and the ratio of maximum grip force to the ideal minimal grip force. The former quantifies the safety and stability of the grasp, whereas the latter reflects its efficiency and dexterity. Table II summarizes the online evaluation results. The mean translational and rotational displacements were 1.78 ± 2.74 mm and $0.84 \pm 1.33°$, respectively. On average, the maximum applied grip force exceeded the ideal minimal grip force by 35%. As expected, achieving dexterous grasp control was more challenging on the low-friction PTFE surface, leading to increased displacement and a higher maximum-to-ideal grip force ratio. Nevertheless, the results remained within satisfactory limits for all surface types, demonstrating the practicality of our incipient slip detection model in real-time applications.

V. DISCUSSION

The results presented in this study demonstrate the effectiveness, efficiency, and deployability of our proposed HD-based framework for incipient slip detection. The offline classification accuracy (91.78%) and the sub-microsecond-latency hardware implementation (0.42 µs) collectively highlight the strength of our approach in addressing a critical challenge in robotic tactile perception – namely, the need for early and reliable slip detection under tight computational and memory constraints.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE III
OVERALL COMPARISON WITH STATE-OF-THE-ART APPROACH ON INCIPIENT SLIP DETECTION BASED ON PAPILLARRAY SENSOR

		Wang *et al.* (2023) [7]	Our approach
Offline evaluation: incipient slip detection	slip detection methods	GRU-based	HDC-based
	classification task	incpient slip, others	non-slip, incipient slip, gross slip
	data preprocessing	data augmentation, sample rate 25Hz	no augmentation, sample rate 100Hz
	accuracy (%)	95.60%	91.78%
	model size (kB)	*1468.456	0.375
	latency (us)	GPU: ∼25,000	FPGA: 0.42, CPU: 5985
	edge deployable	✗	✓
**Online evaluation: grip-force control	gripper	OnRobot RG2 (modified with stepper motors)	Robotiq 2F-85
	tactile sensor	one PapillArray	one PapillArray
	grip-force control mechanism	simple feedback control	simple feedback control
	slip displacement	2.5 ± 0.7 mm,1.9 ± 0.5°	1.78 ± 2.74 mm, 0.84 ± 1.33°
	max / ideal grip force	not reported	1.35 ± 0.31

* Model size obtained from the publicly available source code provided by the author, other results are reported by the author in the paper.
** Online experiments used different grippers and objects. Experimental results are intended as indicative performance and demonstrate utility under respective configurations, rather than direct numerical comparison.

The proposed graphical–spatial–temporal HD encoder effectively captures local tactile deformation patterns that are critical for early-stage slip detection. Unlike baseline HD encoders (sHDC and stHDC), which only model spatial or spatiotemporal features in isolation, gstHDC jointly incorporates graphical topology, intra-frame spatial distribution, and inter-frame temporal evolution. The graphical encoding preserves the structural relationships among pillars, enabling the model to exploit local correlations across the tactile array. The spatial encoding further characterizes fine-grained pressure variations within each frame, while the temporal encoding captures the dynamic evolution of these deformation patterns over time. By integrating these complementary components, gstHDC forms richer and more discriminative high-dimensional representations, leading to consistent improvement on classification accuracy by 4.86% and 3.44% over sHDC and stHDC, respectively. Moreover, the incorporation of a context-driven training and inference strategy (CL-gstHDC) further enhances robustness by leveraging pillar compression states to make more stable predictions under wide-spanning tactile conditions. As a result, CL-gstHDC achieves a classification accuracy of 91.78%, representing a 1.92% improvement over gstHDC.

The FPGA deployment further affirms the framework's suitability for embedded, edge-level applications. With a memory footprint of only 0.375 kB and modest resource utilization, the model can be integrated directly into compact, low-power systems – such as gripper-mounted controllers – without requiring external computation or cloud connectivity. This is critical for enabling fast reflexive control, where even small latencies can undermine grasp stability in dynamic environments.

The online further evaluation validates the practical applicability. The system maintained acceptable translational and rotational displacement, while the maximum applied grip force exceeded the ideal minimal grip force by only 35% across different surface materials, even when initialized with a minimal grip force. These results underscore the generalizability of our slip detection framework in realistic robotic manipulation scenarios.

Table III presents a comprehensive comparison between the proposed HD-based approach and a state-of-the-art method based on a gated recurrent unit (GRU) neural network for incipient slip detection using the PapillArray tactile sensor. The GRU model achieves 3.82% higher accuracy in offline evaluation. However, this improvement comes at the cost of substantially increased complexity. Specifically, the GRU network has a model size $\sim 4000\times$ larger, requires extensive data augmentation, and is trained on a relatively simpler binary classification task. Consequently, despite its higher accuracy, the GRU model is impractical for deployment on resource-constrained edge devices, which fundamentally limits its applicability. In contrast, our HD-based model offers a superior trade-off between accuracy, efficiency, and deployability. Even when executed on a CPU, the proposed approach outperforms the GRU model running on a GPU, achieving $> 4\times$ lower inference latency. Furthermore, its hardware-friendly architecture enables seamless FPGA implementation, yielding an additional $10^4\times$ acceleration.

For online evaluation, both studies share several methodological similarities: each demonstrates the utility of incipient slip detection through real-time grip-force control, employs a two-fingered gripper equipped with a single PapillArray sensor, and implements a simple reactive control scheme that increases grip force upon slip detection. These online experimental results provide indicative performance and underscore the practical utility of our approach. The grasping tasks employed in our study are sufficiently representative, encompassing not only PTFE surface which is typically more slippery than most everyday objects, but also a non-uniform felt surface with directional-dependent roughness. In addition, beyond demonstrating grasp stability through slip displacement measurements, we further validated that this stability was not achieved through overshooting or excessive grip force, by estimating the ideal grip force and calculating the ratio of maximum to ideal grip force. Collectively, these results provide a more comprehensive validation of HD-based incipient slip detection compared to the prior state-of-the-art approach.

Despite these promising outcomes, we realized that the current experimental platform, while capable of demonstrating the utility of our HD-based incipient slip detection method, does not fully showcase its superiority. For instance, in the present configuration, CPU implementation suffices, and the advantages of FPGA deployment are not fully exploited. However, we anticipate that the proposed ultra-fast (sub-microsecond), lightweight ($<$ 1 kB) slip detection model, designed through algorithm–hardware co-optimization, will be particularly advantageous in scenarios requiring high-speed tactile feedback or stringent resource constraints, such as haptic communication [29], micro-robotics [30], and robotic skin [31]. Future research will further explore these application domains.

VI. Conclusion

This letter presents an efficient HD-based framework for incipient slip detection, combining a novel gstHD encoder with context-driven inference. The model achieves 91.78% accuracy with sub-microsecond latency and minimal resource usage, enabling real-time deployment on edge hardware. Both offline and online evaluations confirm the its robustness and practicality in diverse tactile scenarios, highlighting HD computing as a practical and powerful approach for tactile perception in robotics.

References

[1] W. Chen, H. Khamis, I. Birznieks, N. F. Lepora, and S. J. Redmond, "Tactile sensors for friction estimation and incipient slip detection—Toward dexterous robotic manipulation: A review," *IEEE Sensors J.*, vol. 18, no. 22, pp. 9049–9064, Nov. 2018.

[2] Q. Li, O. Kroemer, Z. Su, F. F. Veiga, M. Kaboli, and H. J. Ritter, "A review of tactile information: Perception and action through touch," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1619–1634, Dec. 2020.

[3] M. A. Lee et al., "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8943–8950.

[4] Z. Hu et al., "Machine learning for tactile perception: Advancements, challenges, and opportunities," *Adv. Intell. Syst.*, vol. 5, no. 7, 2023, Art. no. 2200371.

[5] Y. Lu et al., "Machine learning-enabled tactile sensor design for dynamic touch decoding," *Adv. Sci.*, vol. 10, no. 32, 2023, Art. no. 2303949.

[6] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de Gabriel, "CNN-based methods for object recognition with high-resolution tactile sensors," *IEEE Sensors J.*, vol. 19, no. 16, pp. 6872–6882, Aug. 2019.

[7] Q. Wang, P. M. Ulloa, R. Burke, D. C. Bulens, and S. J. Redmond, "Robust learning-based incipient slip detection using the papillarray optical tactile sensor for improved robotic gripping," *IEEE Robot. Automat. Lett.*, vol. 9, no. 2, pp. 1827–1834, Feb. 2024.

[8] R. D. Howe, N. Popp, P. Akella, I. Kao, and M. R. Cutkosky, "Grasping, manipulation, and control with tactile sensing," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 1258–1263.

[9] M. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–37, 2021.

[10] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cogn. Computation*, vol. 1, pp. 139–159, 2009.

[11] K. Hosoda, Y. Tada, and M. Asada, "Internal representation of slip for a soft finger with vision and tactile sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, vol. 1, pp. 111–115.

[12] W. Yuan, R. Li, M.A. Srinivasan, and E. H. Adelson, "Measurement of shear and slip with a gelsight tactile sensor," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 304–311.

[13] S. Dong, W. Yuan, and E. H. Adelson, "Improved gelsight tactile sensor for measuring geometry and slip," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 137–144.

[14] J. Li, S. Dong, and E. Adelson, "Slip detection with combined tactile and visual information," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 7772–7777.

[15] R. Calandra et al., "The feeling of success: Does touch sensing help predict grasp outcomes?," in *Proc. 1st Annu. Conf. Robot Learn.*, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., 2017, vol. 78, pp. 314–323.

[16] J. W. James, S. J. Redmond, and N. F. Lepora, "A biomimetic tactile fingerprint induces incipient slip," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9833–9839.

[17] T. Taunyazov et al., "Extended tactile perception: Vibration sensing through tools and grasped objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1755–1762.

[18] H. Khamis, B. Xia, and S. J. Redmond, "Real-time friction estimation for grip force control," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1608–1614.

[19] P. Martinez Ulloa, D. C. Bulens, and S. J. Redmond, "Incipient slip detection for rectilinear movements using the papillarray tactile sensor," in *Proc. IEEE Sensors*, 2022, pp. 1–4.

[20] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits Syst. Mag.*, vol. 20, no. 2, pp. 30–47, Secondquarter 2020.

[21] M. Heddes et al., "Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures," *J. Mach. Learn. Res.*, vol. 24, no. 255, pp. 1–10, 2023.

[22] A. Moin et al., "A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition," *Nature Electron.*, vol. 4, no. 1, pp. 54–63, 2021.

[23] A. Rahimi, P. Kanerva, J. D. R. Millán, and J. M. Rabaey, "Hyperdimensional computing for noninvasive brain-computer interfaces: Blind and one-shot classification of EEG error-related potentials," in *Proc. 10th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol.*, 2017.

[24] A. Burrello, K. Schindler, L. Benini, and A. Rahimi, "One-shot learning for IEEG seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2018, pp. 1–4.

[25] T. Chen et al., "Energy-efficient sleep apnea detection using a hyperdimensional computing framework based on wearable bracelet photoplethysmography," *IEEE Trans. Biomed. Eng.*, vol. 71, no. 8, pp. 2483–2494, Aug. 2024.

[26] A. Rahimi, P. Kanerva, L. Benini, and J. M. Rabaey, "Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of ExG signals," *Proc. IEEE Proc. IRE*, vol. 107, no. 1, pp. 123–143, Jan. 2019.

[27] M. Imani et al., "Revisiting hyperdimensional learning for FPGA and low-power architectures," in *Proc. IEEE Int. Symp. High- Perform. Comput. Architecture*, 2021, pp. 221–234.

[28] G. Westling and R. S. Johansson, "Factors influencing the force control during precision grip," *Exp. Brain Res.*, vol. 53, no. 2, pp. 277–284, 1984.

[29] K. H. Allen, C. Rogers, and E. S. Short, "Haptic communication in human-human and human-robot co-manipulation," 2025, *arXiv:2509.18327*.

[30] A. F. Tabak and I. S. Khalil, "Haptics in micro-and nano-manipulation," 2024, *arXiv:2412.06917*.

[31] P. Roberts, M. Zadan, and C. Majidi, "Soft tactile sensing skins for robotics," *Curr. Robot. Rep.*, vol. 2, no. 3, pp. 343–354, 2021.