2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

TacTip-based Dynamic Contact Force Estimation with Sequential Tactile Images and Its Applications to Robotic Force Tracking

Wantong Xie[1], Zhenyu Lu[2], Jingyang Liu[1], Jialong Yang[2], Lu Chen[1*], Chenguang Yang[3]

***Abstract*— Force estimation is crucial for robotics, human–machine interaction, and industrial automation. However, traditional methods are often hindered by high cost, mechanical wear, and limited accuracy in dynamic scenarios. Vision-based tactile sensing provides a promising alternative, yet existing approaches commonly rely on static calibration and degrade under dynamic interactions such as slip. To overcome these limitations, we present a novel force prediction framework for TacTip sensors, termed as Frame-stack Force Prediction Method (FFPM). The framework integrates a Dynamic Tactile Flow Encoder to capture spatiotemporal features, enabling accurate modeling of dynamic force variations. An Exponentially Weighted Residual Correction strategy is further introduced to refine predictions by leveraging historical residuals, yielding smoother and more reliable force estimation. The predicted forces are incorporated into a force-tracking impedance control scheme, achieving precise tracking during slip interactions. Experiments on our constructed dataset demonstrate state-of-the-art performance, reducing MAPE to 12.54%, and further validate the effectiveness of the proposed framework in real-world dynamic force estimation and control.**

I. Introduction

Force estimation aims to predict and quantify the interaction forces between robot and its environment to enable precise and adaptive manipulation, which is a fundamental technique in human-machine interaction and industrial automation. Traditional approaches generally rely on physical force sensors, which suffer limitations of high cost and mechanical wear [1]. With the rapid development of embodied intelligence and deep learning, vision-based methods [2] could utilize object deformation, motion, or other visual cues to predict forces, and hence have gained increasing attentions due to their advantages of non-contact measurement, low cost, and ease of deployment. However, they still face challenges in dynamic scenarios, especially under lighting variations, occlusions, fast interactions, or micro-deformed and low-textured surfaces [3].

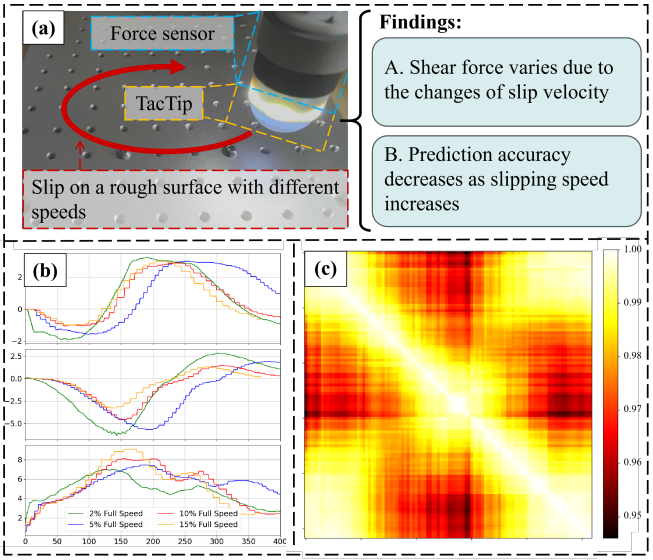

Fig. 1. Illustration of tactile force estimation in dynamic slip movements. (a) A tactile sensor mounted on a robot with a 6D force/torque sensor sliding across rough surface. (b) The effect of different slip speeds (2%, 5%, 10%, and 15% of maximum speed) on force data in Z, X, and Y-axes. The results show minor variations in the Z-axis but significant changes in the X–Y plane. (c) Structural similarity of force prediction at various slip speeds, where prediction accuracy decreases as slip speed increases.

In contrast, optical tactile sensors provide a more direct and robust means of inferring forces. These sensors employ an internal camera to capture the deformation of a soft surface through patterns of light propagation, reflection, or scattering. The resulting tactile images can be processed, typically using Finite Element Method (FEM) models or advanced Neural Networks (NNs), to extract rich contact information, including force, slip, texture, and pressure distribution [4], [5], [6]. A central research focus is the construction of stiffness matrices [7] or force calibration models [8] to map tactile images to force measurements, which is crucial for robotic manipulation. However, these calibration methods are typically predefined under static conditions, limiting their effectiveness in dynamic interactions, particularly when tactile sensors such as TacTip undergo large deformations during slip. As noted in [9], dynamic contact forces are often smaller than static ones due to reduced friction coefficients, further complicating accurate prediction.

To intuitively demonstrate these issues, we design an experimental platform using a TacTip sensor mounted on a robot equipped with a 6D force/torque sensor (Fig.1 a). During experiments, the robot slides the sensor across rough

*Corresponding author: L. Chen, W. Xie and Z. Lu contributed equally to this work.
[1]W. Xie, J. Liu and L. Chen are with Institute of Big Data Science and Industry, Key Laboratory of Evolutionary Science Intelligence of Shanxi Province, and School of Artificial Intelligence, Shanxi University, Taiyuan 030006, China (Email: chenlu@sxu.edu.cn).
[2]Z. Lu and J. Yang are with School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China.
[3]C. Yang is with Department of Computing, The Hong Kong Polytechnic University, HKSAR, China.
This work was jointly supported by the National Natural Science Foundation of China (No. 62373233), the Special Program for Patent Transformation in Shanxi Province (No. 20250012), the Fundamental Research Funds for the Central Universities (No. 2025ZYGXZR057), and the special fund for Science and Technology Innovation Teams of Shanxi Province.

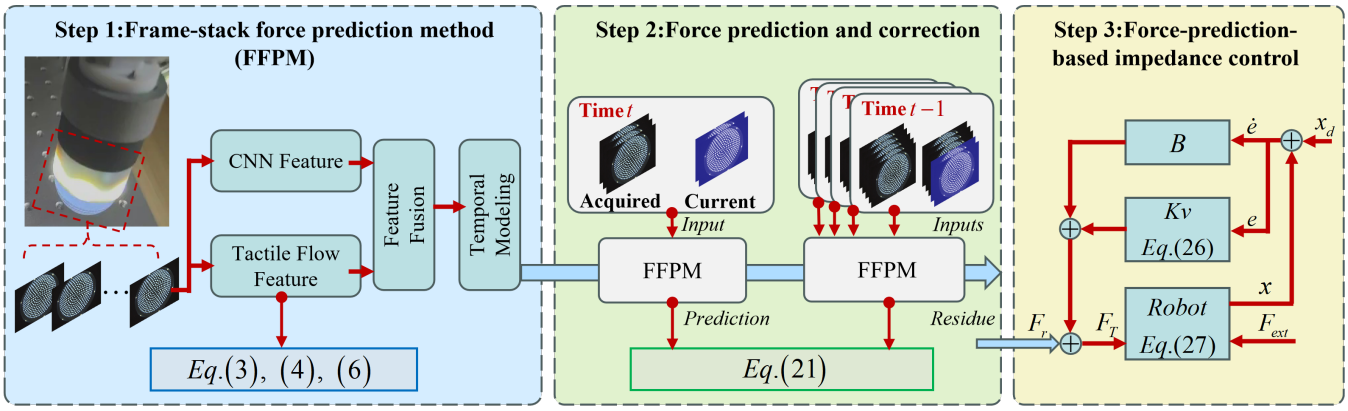

Fig. 2. Overview of the proposed framework for force estimation. Step 1: a frame-stack force prediction method is developed to enhance prediction accuracy under dynamic interactions. Step 2: an Exponentially Weighted Residual Correction strategy is applied to further refine the predictions. Step 3: based on the corrected results, a force-tracking impedance control scheme is constructed for precise interaction.

surfaces while recording tactile images and ground-truth forces. Results show that different slip speeds have only minor effects on Z-axis forces but cause significant variations in the X–Y plane (Fig.1 b). In addition, a CNN is applied to tactile images, achieving structural similarity greater than 95% at low slip speeds (Fig. 1 c). However, the accuracy drops to approximately 60% at higher speeds, highlighting the insufficiency of static, single-frame prediction.

These findings motivate the exploration of video-stream-based tactile force estimation, which leverages temporal continuity to provide smoother and more stable predictions—similar to advances in visual object recognition. Nevertheless, two primary challenges remain for tactile force prediction using video streams:

- **Prediction based on future frames.** During training, sequences centered on current frame can include both past and future frames, enabling better averaging of force estimates. However, in real applications, future frames are unavailable, which limits predictive accuracy.
- **Prediction–control integration.** Predictive control requires both forward-looking force prediction and backward error correction. Historical data alone can not support real-time force prediction or correct prediction errors.

To address these challenges, we propose a novel frame-stack force prediction method for TacTip sensors in dynamic interactions. This framework leverages a tactile flow method to process sequential tactile images and capture spatiotemporal features, and further incorporates an Exponentially Weighted Residual Correction strategy to refine predictions for smoother and more reliable force estimation. The overall structure is illustrated in Fig. 2 and consists of three modules. The main contributions of this work are summarized as follows:

- We introduce a novel Frame-stack Force Prediction Method that effectively models dynamic interaction forces by processing sequential tactile images. This method incorporates a dual-stream architecture featuring a Dynamic Tactile Flow Encoder, enabling the framework to decouple and capture both subtle spatial deformations and complex spatiotemporal flow features.
- We propose an Exponentially Weighted Residual Correction mechanism to refine frame-stack predictions for smoother and more reliable force estimation, which is further integrated into a force-tracking impedance control strategy to improve slip interaction accuracy.
- Extensive experiments demonstrate that the proposed method achieves state-of-the-art performance on self-constructed dataset, reducing the MAPE to 12.54%. Furthermore, its effectiveness and robustness are validated through real-world robotic force-tracking tasks.

II. Related Works

A. Conventional Force Estimation Methods

Conventional methods play a crucial role in force estimation, particularly when combined with data-driven approaches. Peng *et al.* [10] developed a FEM-based modeling approach that generates 3D contact force labels to train regression models for dense force estimation. Lin *et al.* [11] proposed a compact visuotactile sensor capable of reconstructing 3D contact shape and estimating 6-DOF force. Shan *et al.* [12] enhanced tactile image dimensionality by introducing magnetic markers to achieve high-precision non-contact force estimation. Fan *et al.* [13] demonstrated the feasibility and engineering advantages of rapid manufacturing for visuotactile sensors. Andrussow *et al.* [14] developed a fingertip-sized sensor that outputs high-resolution 3D contact force maps at 60 Hz. However, conventional approaches often suffer from limited scalability and adaptability to complex, high-dimensional tactile data, making them less effective for real-time force estimation in dynamic interactions.

B. Tactile Sensing with Deep Networks

Recent years have witnessed rapid progress in tactile sensing and force estimation, with efforts spanning from physics-inspired modeling to learning-based methods. Liu *et al.* [15] proposed a contact state estimation method that explicitly models environment dynamics and constraints.Wang

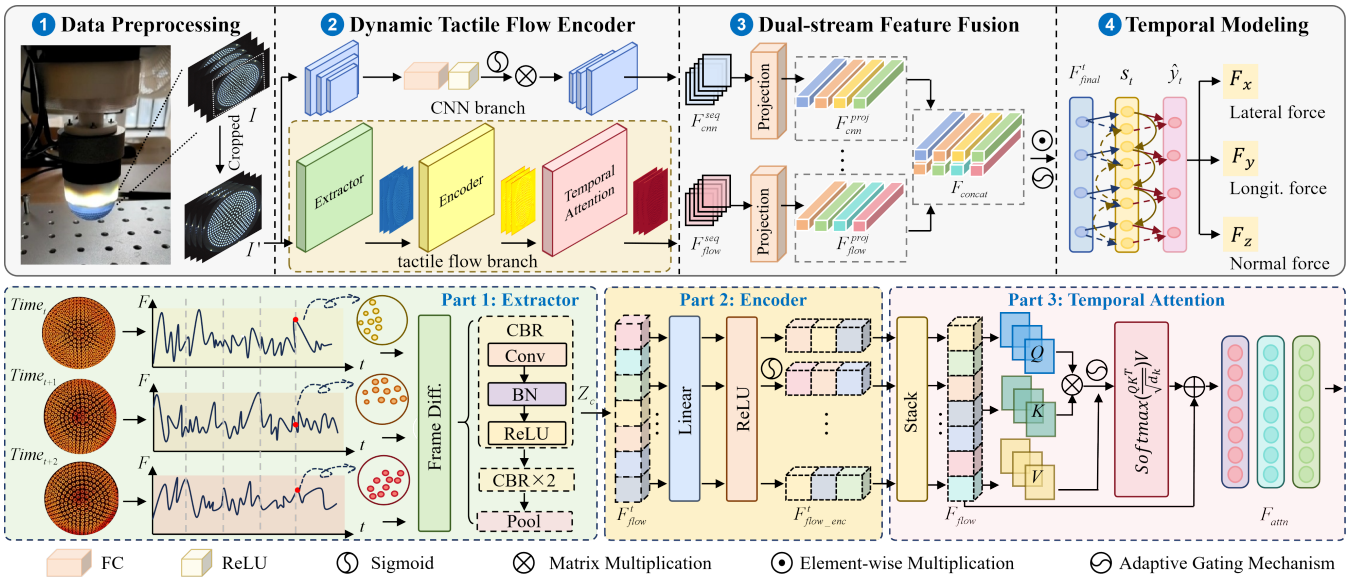

Fig. 3. **Overview:** The architecture consists of four components: data processing, dynamic tactile flow encoder, dual-stream feature fusion, temporal modeling. The input sequence I is preprocessed to obtain I', which is fed into two parallel branches: a CNN branch producing F_{cnn}^{seq} and a tactile-flow branch processing differential images to generate F_{flow}^{seq}. After feature fusion, the final representation F_{final}^{t} is obtained, which is further modeled by LSTM for temporal dynamics and passed to the prediction head for frame-wise 3D force estimation $\{F_x, F_y, F_z\}$.

et al. [16] introduced TACTO, a simulator providing high-fidelity tactile data and benchmarks for learning-based force estimation. Gao *et al.* [17] developed a temporal-convolution-based approach to capture dynamic tactile signals for real-time slip detection. Hu *et al.* [18] estimated contact force fields from tactile signals and utilized their entropy properties for fine-grained slip detection. Komeno *et al.* [19] injected vibrations into a soft sensor and analyzed temporal responses to detect incipient slip. However, most existing learning-based methods still rely on static tactile measurements or offline calibration, lack temporal context modeling, and are therefore limited in their applicability to dynamic interaction scenarios.

III. Method

A. Frame-stack Force Prediction Overview

The overall architecture of the proposed method is illustrated in Fig. 3, which consists of four primary components. First, in data processing stage, it takes a sequential RGB input $I = \{i_1, \ldots, i_T\} \in \mathbb{R}^{A\times T\times 3\times U\times N}$, where A and T denote the batch size and number of time steps, while U and N represent the spatial height and width of the images, respectively, and perform cropping operation to get I' for mitigating the interference of neighboring regions. Second, the dynamic tactile flow encoder employs an encoder-attention architecture to capture spatiotemporal dynamic features. Third, in dual-stream feature fusion phase, dual-stream features are projected, concatenated, and fused via an adaptive gating mechanism to form the final representation F_{final}^{t}. Finally, the temporal modeling stage utilizes a LSTM to capture temporal dependencies, and the force prediction head outputs frame-wise three-dimensional force estimations $\{F_x, F_y, F_z\}$.

B. Dynamic Tactile Flow Encoder

The Dynamic Tactile Flow Encoder consists of two main branches: 1) CNN branch, with MobileNetV3-Small as backbone, processes single-frame images to produce a sequence-level feature F_{cnn}^{seq}, 2) tactile flow branch extracts motion dynamics from tactile sequences, which are encoded and refined through temporal attention to obtain F_{flow}^{seq}. Specifically, tactile flow branch consists of three functional parts: the flow feature extractor, the encoder, and the temporal attention.

In the extractor part, the absolute difference between the pixel values of the current frame i_t and its preceding frame i_{t-1} is computed to quantify the motion and variations across adjacent frames, where i_t represents the t-th frame. The formulation is given as follows:

$$\Delta_t = \begin{cases} |i_2 - i_1|, & \text{if } t = 0 \text{ and } T > 1, \\ |i_t - i_{t-1}|, & \text{if } t > 0, \\ 0, & \text{if } t = 0 \text{ and } T = 1, \end{cases} \tag{1}$$

where Δ_t denotes the frame-difference map at time step t, and $|\cdot|$ indicates the element-wise absolute difference. Then, a sequence of CBR (Convolution-Batch Normalization-ReLU) modules encodes the frame-difference images into high-level features. The final convolutional output $M^{(L)} \in \mathbb{R}^{u\times n}$ is compressed into channel-wise statistics Z_c via global average pooling, which is formulated as:

$$Z_c = \frac{1}{u \times n} \sum_{i=1}^{u} \sum_{j=1}^{n} M_c^{(L)}(i, j), \tag{2}$$

where Z_c denotes the global average pooled feature of the c-th channel.

In encoder part, we perform the following transformation:

$$F_{flow}^{t} = [Z_1, Z_2, \ldots, Z_c]^T \in \mathbb{R}^{D_{flow}}, \tag{3}$$

where D_{flow} is the dimension of the flow features. Then flow features are encoded using a two-layer bottleneck fully connected network, applied independently at each time step to process F_{flow}^{t}. This structure expands the dimension from D_{flow} to $2 \cdot D_{flow}$ and then compresses it back to D_{flow}, thereby enhancing the non-linear representation capability. The process is as follows:

$$F_{flow\_enc}^{t} = \sigma(W_{e2} \cdot \text{ReLU}(W_{e1} \cdot F_{flow}^{t} + b_{e1}) + b_{e2}), \tag{4}$$

where W_{e1} and W_{e2} represent the weights of linear transformations, and b_{e1} and b_{e2} are corresponding bias terms.

In temporal attention part, independent feature vectors $F_{flow\_enc}^{t}$ are first aggregated to form a complete sequence:

$$F_{flow} = [F_{flow\_enc}^{0}, \ldots, F_{flow\_enc}^{T-1}] \in \mathbb{R}^{A \times T \times D_{flow}}. \tag{5}$$

F_{flow} is then linearly transformed using three different sets of learnable weight matrices (W_Q, W_K, W_V). After computing the similarity, a weighted transformation is applied to the last dimension, followed by a weighted sum.

$$F_{attn} = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_{flow}}}\right)V, \tag{6}$$

where $Q = F_{flow}W_Q$, $K = F_{flow}W_K$, and $V = F_{flow}W_V$. This operation ensures that the features of each time step incorporate relevant information from historical frames within the sequence. To facilitate fusion with F_{cnn}^{seq}, F_{attn} is further aggregated along temporal dimension to obtain F_{flow}^{seq}.

C. Dual-stream Feature Fusion and Temporal Modeling

The Dual-stream Feature Fusion first projects F_{cnn}^{seq} and F_{flow}^{seq}, into a latent space to obtain F_{cnn}^{proj} and F_{flow}^{proj}, respectively. Subsequently, a gated mechanism powered by a multi-layer perceptron is employed to adaptively fuse these features, yielding the final representation F_{final}^{t}. The procedure is as follows:

$$\begin{aligned} F_{concat} &= F_{cnn}^{proj} || F_{flow}^{proj}, \\ f(F_{concat}) &= W_{i+1}^{c} \cdot \text{ReLU}(W_{i}^{c} \cdot F_{concat}) + b_{i}^{c}, \\ F_{final}^{t} &= J \odot f(F_{concat}) + (1 - J) \odot F_{concat}, \end{aligned} \tag{7}$$

where $||$ denotes feature concatenation, $\odot$ represents element-wise multiplication, W_i^c and b_i^c denote the weight matrix and bias term used in the feature fusion process, respectively, and $J \in [0, 1]$ is the gating weight generated by the Sigmoid function.

In the Temporal Modeling stage, we utilize a Long Short-Term Memory (LSTM) network to process the temporal sequences and produce the per-frame three-dimensional forces F_x, F_y, F_z:

$$\begin{aligned} s_t &= \text{LSTM}(F_{final}^{t}, s_{t-1}), \\ \hat{y}_t &= W_{i+1}^{l} \cdot s_t + b_i^{l} \in \mathbb{R}^{A \times T \times D_{force}}, \end{aligned} \tag{8}$$

where W_i^l denotes the weight matrix of the i-th layer, in which $i \in \mathbb{Z},\ i = 1, 2$, D_{force} indicates the independent output of a single sample, s_t and b_i^l represent the hidden state at time step t and the bias of the i-th layer, respectively.

D. Loss Function

We implement a multi-task loss function, which consists of three components: the primary force prediction loss L_{force} that measures the discrepancy between the predicted and ground-truth forces, the temporal consistency loss $L_{temporal}$ that incorporates the prior knowledge of force smoothness in the physical world, and the flow feature regularization loss L_{flow}. L_{force} is defined as:

$$L_{force} = \frac{1}{A \cdot T} \sum_{i=1}^{A} \sum_{t=1}^{T} \ell\left(\hat{y}_t^{(i)}, y_t^{(i)}\right), \tag{9}$$

where $\hat{y}_t^{(i)}$ and $y_t^{(i)}$ denote the predicted and ground-truth force of the i-th sample at time step t, respectively. ℓ is defined as the squared L_2 norm.

To capture the smoothness prior of forces in the physical world, we define:

$$L_{temporal} = \frac{1}{A \cdot (T-1)} \sum_{i,t} \left\| \Delta\hat{y}_t^{(i)} - \Delta y_t^{(i)} \right\|^2, \tag{10}$$

where $\Delta\hat{y}_t = \hat{y}_t - \hat{y}_{t-1}$ denotes the temporal change of the model prediction at time t, and $\Delta y_t = y_t - y_{t-1}$ denotes the temporal change of the ground-truth label.

The flow feature regularization loss is formulated as:

$$\begin{aligned} L_{flow} = &\frac{1}{A \cdot (T-1)} \sum_{i=1}^{A} \sum_{t=2}^{T} \left\| F_{flow,t}^{(i)} - F_{flow,t-1}^{(i)} \right\| \\ &+ \lambda \cdot \frac{1}{A \cdot T} \sum_{i=1}^{A} \sum_{t=1}^{T} \left\| F_{flow,t}^{(i)} \right\|, \end{aligned} \tag{11}$$

where $F_{flow,t}^{(i)}$ denotes the flow characteristics of sample i, and λ represents the sparsity regularization coefficient with a fixed value of 0.1.

The total loss is expressed as:

$$L_{total} = L_{force} + \eta \cdot L_{temporal} + \theta \cdot L_{flow}, \tag{12}$$

where η and θ denote the weights of the temporal consistency loss and the flow feature regularization, respectively, with values of 0.001 and 0.005.

E. Exponentially Weighted Residual Correction for Force Prediction

We assume the force estimated by FFPM as F_p, and the corrected result as F_c, and the delayed real force as F_r, then the force after correction is

$$F_c(t) = F_p(t) + \alpha(F_r(t-1) - F_p(t-1)), \tag{13}$$

where α is a coefficient and F_c is achieved based on the frame-stack prediction $F_p(t)$. Using Eq. 13, the current force $F_p(t)$ estimated from the images is corrected by the former force prediction residue error. Defining $e(t-1) = F_r(t-$

$1) - F_p(t-1)$ as a force prediction error, we use historical information for multi-step prediction:

$$F_c(t) = F_p(t) + \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \tag{14}$$

Since $0 < \alpha < 1$, the earlier error $e(t-k), k > 1$ affects the final prediction less than that at time $t-1$. However, it is helpful for the force prediction model from following aspects.

Remark 1: Compared with Eq. 13, Eq. 14 is improved with several differences. First, we use historical data instead of former moment sampling error $e(t-1)$, for $F_r(t-1)$ may not be acquired if $T \neq 1$. We use the newest real force measurement $e(t-k), k \geq K$ for force correction. Hence, from time $t-1$ to $t-T$, there are no correction terms $e(t-k)$. Therefore, if we use more historical data, having a larger value of T, the prediction accuracy may decrease. Instead, we extend the value of K to improve the prediction effect and set $T = 4$ and $K = 2$ in this work.

Based on Eq. 14 and definition of a new error term $e_p(t-1) = F_r(t-1) - F_p(t-1)$, we can correct the force estimation error term as:

$$e_p(t) = e(t) + \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \tag{15}$$

According to the description in Section III.B, we know that the prediction $F_p(t)$ is acquired with a lag of T, and the current real force prediction $e(t)$ cannot be acquired. Instead, we define $\hat{e}_p(t)$ as the force error prediction, and develop the EWRC mechanism that leverages the historical estimation errors $\hat{e}_p(t-P), P \in \mathbb{N}$ to predict future prediction errors. Then, the P-step prediction error can be estimated as:

$$\hat{e}_p(t+P) = \sum_{\varepsilon=0}^{P-1} \beta^{P-\varepsilon} \cdot \hat{e}_p(t+\varepsilon) + \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k), \tag{16}$$

where β is a coefficient, having similar function to α for force prediction.

Remark 2: In Eq. 16, we use $\sum_{\varepsilon=0}^{P-1} \beta^{P-\varepsilon} \cdot \hat{e}_p(t+\varepsilon)$ instead of $e(t)$ in Eq. 15 for prediction. If we set $\beta = 1-\alpha$, Eq. 16 can be further expressed as:

$$\hat{e}_p(t+P) = (1-\alpha)^P \hat{e}_p(t) + \sum_{\varepsilon=1}^{P-1} \beta^{\varepsilon} \cdot \hat{e}_p(t+\varepsilon) + \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \tag{17}$$

Set $P = 1$ and $\hat{e}_p(t) = e_p(t)$ at time t, we have:

$$\hat{e}_p(t+1) = (1-\alpha) e_p(t) + \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \tag{18}$$

If we add the real force at time t at both side of Eq. 18, we have:

$$\begin{aligned} \hat{e}_p(t+1) + F_r(t) &= (1-\alpha) F_p(t) + \alpha F_r(t) + \\ \sum_{k=T}^{K+T} \alpha^k e(t-k) &= (1-\alpha) \left[F_c(t) - \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k) \right] \\ &+ \alpha F_r(t) - \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \end{aligned} \tag{19}$$

It is obvious that $F_r(t)$ is unknown for current and future predictions, we therefore use F_c to replace F_r in Eq. 19 as:

$$\hat{F}_p(t+1) = F_c(t) - (2-\alpha) \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \tag{20}$$

Similarly, we can deduce the following Eq. 21 for force prediction:

$$\hat{F}_p(t+P) = F_c(t) - \sum_{i=0}^{P} (1-\alpha)^i \sum_{k=T}^{K+T} \alpha^k \cdot e(t-k). \tag{21}$$

Remark 3: If $P = 0$, Eq. 21 can be simplified as Eq. 14 to ensure the consistence of the expression. The prediction model is useful for the robot control, since the image processing and precision takes time and robot control has a high-speed signal processing rate, so we can use Eq. 21 for forward force prediction. On the other hand, if the robot is controlled with a higher rate that is much faster that the image sampling and processing, we can improve Eq. 21 using fractional-order residual prediction to achieve a higher rate force control.

F. Force-prediction-based Impedance Control for Dynamic Force Tracking

The robot interaction model can be described as:

$$H(x)\,\ddot{x} + C(x,\dot{x})\,\dot{x} + G(x) = F_\tau + F_{ext}, \tag{22}$$

where x represents the position of the robot end in the Cartesian coordinate and $H(x)$, $C(x,\dot{x})$ and $G(x)$ denote the inertia matrix, Coriolis and centrifugal matrix and gravity vector, respectively. F_τ is the driving force term, and F_{ext} represents the external force term. The interaction with the environment can be utilized to be encounted using an impedance model with forward force control:

$$F_{ext} = F_r + B\dot{e} + Ke, \tag{23}$$

where F_r represents the force predicted by the frame-stack method in Eq. 21, and B and K are the damping and stiffness coefficients, respectively. The damping-stiffness term is used for system stabilization under unknown disturbances and force prediction errors. Here, $e = x - x_d$ denotes the error between actual robot position x and desired position x_d.

We set the desired force as F_d, then the force error is

$$\begin{aligned} \Delta F &= F_{ext} - F_d \\ &= F_r + B\dot{e} + Ke - F_d \\ &= \Delta F + B\dot{e} + Ke. \end{aligned} \tag{24}$$

TABLE I

ACCURACY AND EFFICIENCY COMPARISONS OF DIFFERENT METHODS UNDER OUR DATASET.

Method	Size	Latency	MAE↓	MSE↓	MAPE↓	R^2 ↑
EfficientVit [20]	49.42	5.26	0.15	0.27	17.01%	0.92
CNN-ViT [21]	37.38	4.83	0.14	0.25	16.73%	0.94
CNN-GRU [22]	61.01	6.15	0.13	0.20	15.62%	0.95
Mamba-Sea [23]	**8.59**	**3.98**	0.09	0.03	13.20%	0.96
Ours	17.32	4.17	**0.08**	**0.02**	**12.54%**	**0.99**

TABLE II

IMPACT OF HISTORICAL FRAME LENGTH ON 3D FORCE ESTIMATION PERFORMANCE AND INFERENCE EFFICIENCY.

Frames	MAE↓	Infer. Time (ms)↓	FPS↑	Test Loss↓
1	0.29	3.57 ± 0.03	279.44	0.25
2	0.23	3.95 ± 0.05	252.84	0.12
3	0.28	4.08 ± 0.07	245.07	0.19
4	0.13	4.17 ± 0.02	239.65	0.03
5	0.14	4.30 ± 0.03	232.24	0.04
6	0.14	4.52 ± 0.03	221.00	0.04
7	0.16	4.80 ± 0.05	208.20	0.06
8	0.13	4.77 ± 0.04	209.61	0.04
12	0.12	4.85 ± 0.05	205.77	0.03
16	0.11	5.08 ± 0.02	196.72	0.02

Then set $K_v = \frac{KK_s}{K+K_s}$, where K_v is a variable stiffness term that changes with the movement of the robot. If there is no virtual impedance, K_s plays a compliance role.

The control law of K_v is designed as:

$$K_v = -\frac{k_p \Delta F + k_i \int_0^t \Delta F dt + k_d \dot{\Delta F}}{e}, \tag{25}$$

where k_p, k_i, and k_d are constant gains like PID. The following variable admittance control law can be obtained and discretized as:

$$\dot{x}(t) = \frac{(k_p+1)\Delta F + k_i \Delta t \sum_{i=0}^{t} \Delta F + k_d \frac{d\Delta F}{\Delta t} + F_d(t)}{B}, \tag{26}$$

where Δt is the system sampling period, and $d\Delta F = \Delta F(t) - d\Delta F(t-1)$ represents the force tracking error.

IV. EXPERIMENTS AND RESULTS

A. Dataset and Implementation Details

Dataset: We conduct experiments on our self-constructed dataset, which includes two types of motions: linear motion on an aluminum breadboard and circular motion with a radius of $50\,\mathrm{mm}$. For each object, four different speeds are considered, including 20, 50, 100, and $150\,\mathrm{mm/s}$.

Training Details: All algorithms are trained on a NVIDIA RTX 3090 GPU with $24\,\mathrm{GB}$ memory. We employ the AdamW optimizer with an initial learning rate of 0.001. A cosine annealing schedule is adopted to decay the learning rate from 0.001 to 1×10^{-7} over 200 epochs. The weight decay coefficient is set to 0.01.

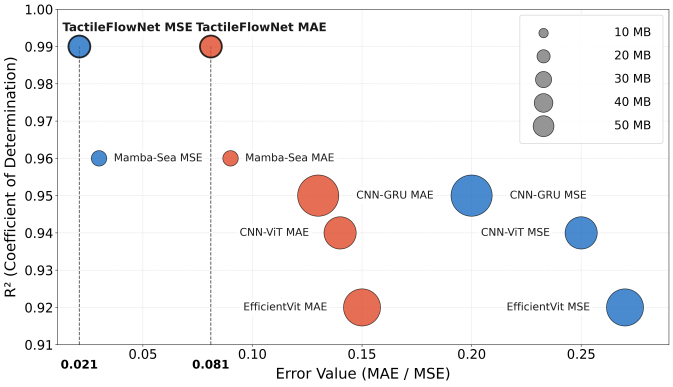

Fig. 4. Comparison of prediction error, coefficient of determination, and computational cost among different methods. The red circles represent MAE values, while blue circles represent MSE values.

B. Force Estimation Results

Quantitative Results. Our method demonstrates accurate three-dimensional force estimation on the constructed dataset. Performance is evaluated using MAE, MSE, MAPE, and the coefficient of determination (R^2) to comprehensively assess estimation accuracy and goodness-of-fit. To validate the performance of FFPM, we select representative models as baselines, including advanced vision-based architectures and state-of-the-art temporal modeling methods, covering spatial and temporal feature extraction capabilities. As summarized in Table I, the proposed FFPM consistently outperforms all competing methods across all metrics, achieving an R^2 of 0.99 with low MAE (0.08), MSE (0.02), and MAPE (12.54%). Compared with EfficientVit, FFPM reduces MSE from 0.27 to 0.02. Despite its significantly smaller model size, FFPM also surpasses CNN-GRU, yielding a 38.5% reduction in MAE. In addition, FFPM achieves a 5% lower MAPE than the lightweight Mamba-Sea. These results indicate that FFPM achieves the best balance between model efficiency and force estimation accuracy.

Table II analyzes the effect of historical frame length on estimation accuracy and computational latency. Increasing the frame length from 1 to 4 markedly improves performance, reducing the test loss from 0.25 to 0.03, which confirms the benefit of temporal context modeling. Further increases yield only marginal gains while causing higher inference latency and lower FPS (model inference frame rate). This indicates redundant temporal information and increased computational cost. Therefore, a four-frame input is adopted in subsequent experiments.

Fig. 4 illustrates the trade-off among estimation accuracy, model complexity, and real-time feasibility, where our method resides in the optimal upper-left region, achieving the highest R^2 values and the lowest estimation errors. These results demonstrate that the proposed approach attains superior accuracy while maintaining a compact and computationally efficient model structure.

Qualitative Analysis. Fig. 5 presents a comparison of different methods across three consecutive frames in an

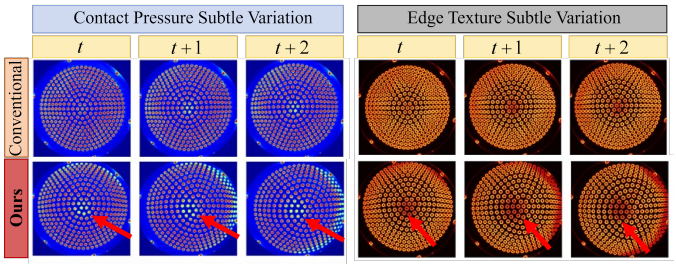

Fig. 5. Comparative visualization of contact pressure and edge texture of different methods across three consecutive frames in an arc motion. "Conventional" denotes standard differencing computed between adjacent frames.

TABLE III
PERFORMANCE COMPARISON OF THE DYNAMIC TACTILE FLOW ENCODER CONFIGURATIONS.

Extractor	Temporal Attention	MAPE↓
✗	✗	15.39%
✓	✗	13.68%
✗	✓	14.25%
✓	✓	12.54%

arc trajectory with a speed of 15%. For contact pressure visualization, standard differencing shows limited intensity variation, whereas our method yields broader and more continuous response regions, with higher responses indicated by warmer colors. In edge texture maps, the proposed method produces clearer and more localized boundary responses, whereas conventional results in weak and diffuse contours. The results indicate that tactile-flow-based differencing is more sensitive to subtle variations in edge texture and contact pressure, leading to higher force estimation accuracy.

Ablation Study. To assess the contribution of Dynamic Tactile Flow Encoder, an ablation study is conducted in Table III. The baseline model without the encoder achieves a MAPE of 15.39%. Adding either Extractor or Temporal Attention module individually reduces the MAPE to 13.68% and 14.25% respectively, indicating the effectiveness of both tactile feature extraction and temporal modeling. The full model achieves the lowest MAPE of 12.54%, suggesting that their combination most effectively exploits differential information in tactile sequences for force prediction.

C. Force Correction Experiment

Based on the FFPM prediction, we apply the correction model in Eq. 16 to compensate for estimation errors from unobservable future images. As shown in Fig. 6, while the current model's estimated forces follow the ground truth, the correction model further refines these values. Fig. 7 presents the force estimation errors of FFPM, which are generally below 0.5 N, except for certain special cases such as at sampling times 83 and 171. The correction model leverages historical estimation errors together with the current prediction to adjust the force estimates. Consequently, the average estimation errors across three directions decrease from

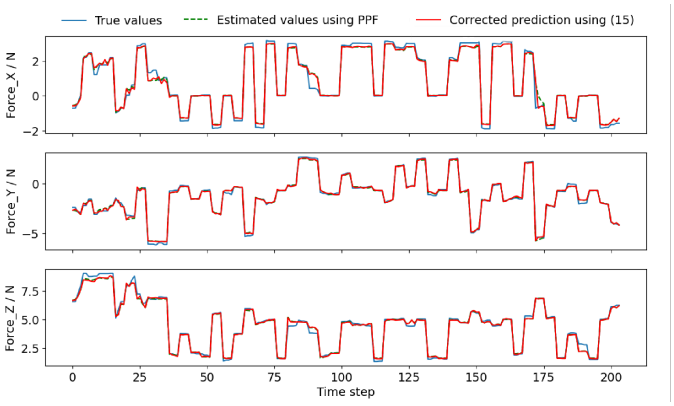

Fig. 6. Comparison of ground-truth force values, initial estimations obtained via FFPM, and corrected results using Eq. 16. The measured forces, FFPM estimates, and corrected results are represented as blue, green, and red lines, respectively.

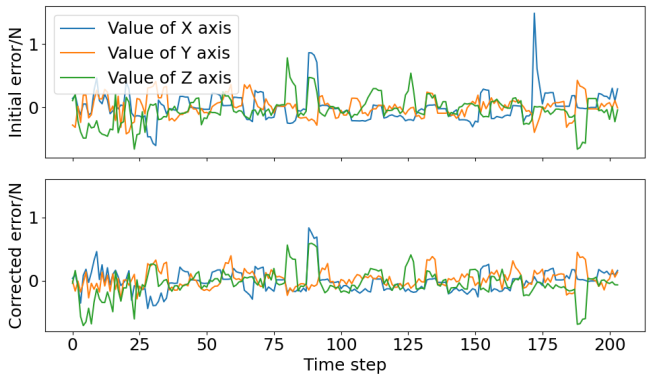

Fig. 7. Comparison of force estimation errors between the original FFPM and corrected model using Eq. 17.

$[0.15, 0.12, 0.16]$ N to $[0.13, 0.10, 0.15]$ N, demonstrating the effectiveness of the proposed framework.

D. Force-prediction-based Impedance Control Experiment

The force-prediction-based admittance control is further implemented to evaluate the performance of dynamic force tracking. The robot, equipped with a TacTip sensor and a 6D force/torque sensor, is commanded to slide over a rough porous plate. Two nuts are placed along the sliding trajectory to serve as obstacles, allowing assessment of force tracking under dynamic interactions. The desired tracking force is set to 2 N, indicated by the red dashed line in Fig. 8 (b).

The robot is evaluated at various sliding velocities along a fixed trajectory on a horizontal plate, while its motion perpendicular to the plate is actively controlled. At low sliding speeds, the contact force remains relatively stable around 2 N, with a force tracking error of approximately 0.2 N. The measured force (yellow line) confirms accurate tracking during the plate-sliding phase. Upon encountering the nuts, the robot slightly lifts after detecting force changes to maintain the desired force, where a noticeable force drop occurs. It can be attributed to the delayed control response in vertical direction.

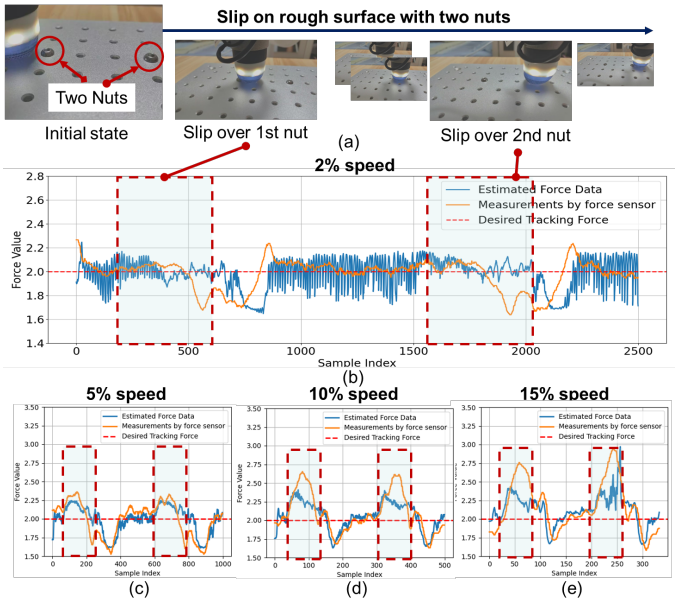

Fig. 8. Force tracking using force-prediction-based impedance control at different slipping speeds.

We subsequently increase the sliding speed to 5%, 10%, and 15% of the maximum speed (1000 mm/s), with the results presented in Fig. 8 (c)–(e). Compared with the lowest speed of 2%, the proposed model maintains high prediction accuracy both before and after collisions with the nuts. However, within the collision intervals (red dashed regions), the contact force increases significantly due to less effective control in vertical direction. A similar force-drop phenomenon is observed across all tested speeds. These results demonstrate the effectiveness of the proposed force-prediction model for dynamic force estimation and tracking.

V. Conclusion

This work introduces the frame-stack force prediction method for TacTip sensors to capture spatiotemporal dynamics. By integrating a Dynamic Tactile Flow Encoder with an Exponentially Weighted Residual Correction mechanism, the framework achieves smooth and reliable force estimation. Experimental results demonstrate that FFPM reduces MAPE to 12.54% and enables high-precision force tracking during complex slip interactions via an impedance control strategy. These findings provide a robust, real-time solution for dynamic tactile perception and robotic force control.

References

[1] H. Yang, H. Zhou, G. S. Fischer, and J. Y. Wu, "A hybrid model and learning-based force estimation framework for surgical robots," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 906–912.

[2] A. Agarwal, A. Wilson, T. Man, E. Adelson, I. Gkioulekas, and W. Yuan, "Vision-based tactile sensor design using physically based rendering," *Communications Engineering*, vol. 4, no. 1, p. 21, 2025.

[3] A.-H. Shahidzadeh, G. M. Caddeo, K. Alapati, L. Natale, C. Fermüller, and Y. Aloimonos, "Feelanyforce: Estimating contact force feedback from tactile sensation for vision-based tactile sensors," in *2025 IEEE International Conference on Robotics and Automation*, 2025, pp. 251–257.

[4] Z. Chen, N. Ou, X. Zhang, and S. Luo, "Transforce: Transferable force prediction for vision-based tactile sensors with sequential image translation," in *2025 IEEE International Conference on Robotics and Automation*, 2025, pp. 237–243.

[5] H. Li, S. Nam, Z. Lu, C. Yang, E. Psomopoulou, and N. F. Lepora, "Biotactip: A soft biomimetic optical tactile sensor for efficient 3d contact localization and 3d force estimation," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5314–5321, 2024.

[6] Z. Lu, Z. Liu, X. Zhang, Y. Liang, Y. Dong, and T. Yang, "3d force identification and prediction using deep learning based on a gelsight-structured sensor," *Sensors and Actuators A: Physical*, vol. 367, p. 115036, 2024.

[7] D. Ma, E. Donlon, S. Dong, and A. Rodriguez, "Dense tactile force estimation using gelslim and inverse fem," in *2019 IEEE International Conference on Robotics and Automation*, 2019, pp. 5418–5424.

[8] B. Fang, J. Zhao, N. Liu, Y. Sun, S. Zhang, F. Sun, J. Shan, and Y. Yang, "Force measurement technology of vision-based tactile sensor," *Advanced Intelligent Systems*, vol. 7, no. 1, p. 2400290, 2025.

[9] Y. Zhang, Z. Kan, Y. Yang, Y. A. Tse, and M. Y. Wang, "Effective estimation of contact force and torque for vision-based tactile sensors with helmholtz–hodge decomposition," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4094–4101, 2019.

[10] J.-C. Peng, S. Yao, and K. Hauser, "3d force and contact estimation for a soft-bubble visuotactile sensor using fem," in *2024 IEEE International Conference on Robotics and Automation*, 2024, pp. 5666–5672.

[11] C. Lin, H. Zhang, J. Xu, L. Wu, and H. Xu, "9dtact: A compact vision-based tactile sensor for accurate 3d shape reconstruction and generalizable 6d force estimation," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 923–930, 2023.

[12] J. Shan, J. Zhao, J. Liu, X. Wang, Z. Xia, G. Chen, Z. Ren, G. Xu, and B. Fang, "Magicgel: A novel visual-based tactile sensor design with magnetic gel," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025, pp. 19 767–19 774.

[13] W. Fan, H. Li, and D. Zhang, "Crystaltac: Vision-based tactile sensor family fabricated via rapid monolithic manufacturing," *Cyborg and Bionic Systems*, vol. 6, p. 0231, 2025.

[14] I. Andrussow, H. Sun, K. J. Kuchenbecker, and G. Martius, "Minsight: A fingertip-sized vision-based tactile sensor for robotic manipulation," *Advanced Intelligent Systems*, vol. 5, no. 8, p. 2300042, 2023.

[15] X. Liu, P. Huang, and Z. Liu, "A novel contact state estimation method for robot manipulation skill learning via environment dynamics and constraints modeling," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3903–3913, 2022.

[16] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra, "Tacto: A fast, flexible, and open-source simulator for high-resolution vision-based tactile sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3930–3937, 2022.

[17] J. Gao, Z. Huang, Z. Tang, H. Song, and W. Liang, "Visuo-tactile-based slip detection using a multi-scale temporal convolution network," *arXiv preprint arXiv:2302.13564*, 2023.

[18] X. Hu, A. Venkatesh, Y. Wan, G. Zheng, N. Jawale, N. Kaur, X. Chen, and P. Birkmeyer, "Learning to detect slip through tactile estimation of the contact force field and its entropy properties," *Mechatronics*, vol. 104, p. 103258, 2024.

[19] N. Komeno and T. Matsubara, "Incipient slip detection by vibration injection into soft sensor," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3251–3258, 2024.

[20] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 420–14 430.

[21] B. H. Ngo, N.-T. Do-Tran, T.-N. Nguyen, H.-G. Jeon, and T. J. Choi, "Learning cnn on vit: A hybrid model to explicitly class-specific boundaries for domain adaptation," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 545–28 554.

[22] T. Y. Yang, J. Y. Lu, Y. Y. Yang, Y. H. Hao, M. Wang, J. Y. Li, and G. C. Wei, "Gnss–vtec prediction based on cnn–gru neural network model during high solar activities," *Scientific Reports*, vol. 15, no. 1, p. 9109, 2025.

[23] Z. Cheng, J. Guo, J. Zhang, L. Qi, L. Zhou, Y. Shi, and Y. Gao, "Mamba-sea: A mamba-based framework with global-to-local sequence augmentation for generalizable medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 44, no. 9, pp. 3741–3755, 2025.