Fourier-enhanced Transformer Encoder Network for Efficient Haptic Texture Modeling/Rendering

Mudassir Ibrahim Awan[1], Sungjoo Kang[2], Dongbeom Ko[2], Waseem Hassan[1,3], Seong Tae Kim[1], and Seokhee Jeon[1,4]

***Abstract*— Haptic texture modeling is essential for immersive environments, yet accurate texture vibration synthesis remains challenging due to high-frequency nature and strong dependence on interaction parameters such as speed and force. Traditional piecewise approaches handle this variability through algorithmic segmentation of signals into stationary components, followed by labeling across discrete contact conditions. However, the resulting segmentation overhead and post-modeling validation create a procedural bottleneck, limiting their ability to represent continuous user interactions. Recent deep learning methods eliminate explicit segmentation by learning continuous mappings, but their high computational cost hinders real-time deployment. To address these limitations, a lightweight Fourier-enhanced Transformer Encoder Network is proposed that eliminates segmentation and enables real-time texture rendering while maintaining high reconstruction fidelity. The model integrates a transformer encoder with the Fourier Transform to process uniform sliding window inputs, separating spectral magnitude and phase in a Fourier block to capture texture-dependent frequency structure, while temporal features are extracted in a compact encoder for one-step prediction at approximately 1 kHz. The proposed approach was evaluated on a diverse set of textures, achieving strong reconstruction accuracy (time-domain: MAE 0.149; spectral domain: GFC 92.15%). A psychophysical experiment further confirms its perceptual effectiveness against existing methods when comparing real and synthesized textures.**

***Index Terms*— Haptic feedback, haptic texture modeling, texture rendering, Transformer, vibration synthesis**

I. Introduction

MANY virtual reality (VR) applications, from fashion and e-commerce to medical training and video games, have shown significant interest in integrating haptic texture feedback in their contents [1]. It is well agreed that haptic textures play a vital role in identifying the target surface, enhancing the user's haptic experience. Thus, producing perceptually identical copies of real textures has become a key focus in the haptics community [2].

In the last few decades, a number of different modeling and rendering approaches have been proposed to generate realistic haptic textures, including geometry-based deterministic models [3], stochastic models [4], and data-driven contact dynamics mapping models [5]. Among them, the data-driven approach is showing promising performance in regenerating an exact copy of a real texture. This approach generally involves the recording of elicited vibrations in a rigid tool while stroking the object and mapping them based on the applied actions (i.e., scanning speed and applied force). These vibrations are inherently non-stationary time-series signals with a diverse range of high frequencies. Modeling such signals can be challenging due to their high sensitivity to noise and rapid fluctuations [6]. Initially, several approaches were introduced to model these contact dynamics by dividing the signals into short stationary segments using sophisticated algorithms (e.g., AutoPARM [7], AutoSLEX [8], and Recursive Constraint Projection (RCP) [9]). Each segment was then modeled individually using techniques such as Linear Predictive Coding (LPC), adapted from speech processing [10], as well as autoregressive (AR) and autoregressive moving average (ARMA) models [11], [12]. Subsequently, these segment models were projected onto a multidimensional input space and interpolated during rendering to reconstruct the desired vibrations. Although these techniques achieve reasonable accuracy in signal reconstruction, they pose several challenges, as illustrated in Fig. 1. First, they require intensive preprocessing including signal segmentation, construction of multiple local models, which often involves significant manual effort and parameter tuning, coverage of models in the interaction space, and conversion of model parameters for interpolation. Second, the segmentation-based formulation models each region independently, leading to fragmented representations that fail to capture continuous dynamics across interaction conditions. Third, these local models must be interpolated in the speed–force interaction space during rendering, which may introduce approximation errors and limit generalization under varying interaction condi-

Manuscript received Sept. 8, 2025; revised Dec. 16, 2025 and Mar. 9, 2026; accepted Apr. 3, 2026. This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grants (RS-2022-II220078 "Explainable Logical Reasoning for Medical Knowledge Generation" and RS-2024-00406245 "Development of Software-Defined Infrastructure Technologies for Future Mobility") and in part by the National Research Council of Science & Technology (NST) grant (CRC23021-000), funded by the Korea government (MSIT).

[1]M. I. Awan, W. Hassan, S. T. Kim, and S.Jeon are with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, Korea (e-mail: {miawan, st.kim}@khu.ac.kr).

[2]S. Kang and D. Ko are with the AI Research Laboratory, ETRI, Daejeon 34129, Korea (e-mail: {dbko112,sjkang}@etri.re.kr).

[3]W. Hassan is now with the Public University of Navarre, 31006 Pamplona, Spain (e-mail: waseem.hassan@unavarra.es).

[4]S. Jeon is with the Department of Immersive AX Convergence, Kyung Hee University, Yongin 17104, Korea (e-mail: jeon@khu.ac.kr). Corresponding author: S. Jeon.

tions. Finally, most approaches do not explicitly model phase information and exhibit limited scalability when extending to additional input and output dimensions [10]–[12].

Recent advances in machine learning and deep learning provide promising alternatives for modeling vibration signals directly from data. Neural network based approaches [13], [14] reduced segmentation needs but operated on frequency-domain features such as the Discrete Fourier Transform (DFT), requiring separate post-processing steps to convert predictions back to time-domain signals for rendering, introducing a trade-off between accuracy and computational efficiency. To model raw acceleration signals directly, encoder-decoder architectures combining CNN and BiLSTM layers were introduced [15], avoiding frequency-domain post-processing. However, the sequential nature of recurrent layers and the encoder-decoder structure introduced significant computational overhead, limiting real-time applicability. More recently, Transformer-based architectures have shown strong capability for modeling sequential data through attention mechanisms that capture local and global dependencies. Sequence-to-sequence designs further improved reconstruction quality; however, their encoder–decoder structure remains computationally demanding and limits their applicability in real-time haptic rendering scenarios [16].

Motivated by the limitations of both classical segmentation-based models and recent learning-based approaches, this paper introduces the Fourier-enhanced Transformer Encoder Network (FoTEN) for data-driven haptic texture modeling and rendering. Traditional piecewise modeling methods rely on signal segmentation and independent local models that must be interpolated during rendering, which increases preprocessing complexity and may introduce artifacts when interaction conditions vary. Although neural network based approaches such as [13], [14] demonstrated the value of frequency-domain representations, they required separate post-processing to convert frequency-domain predictions back to time-domain signals for rendering. Encoder-decoder architectures [15] avoided this by modeling raw signals directly, but reintroduced computational overhead through recurrent sequential processing.

Building on these observations, FoTEN introduces two key design choices. First, an encoder-only Transformer replaces both the manual segmentation used in classical approaches and the recurrent encoder–decoder structures used in prior learning-based methods. This design eliminates sequential computation and decoder overhead while operating directly on sliding-window inputs. Second, rather than relying on external frequency-domain post-processing as in prior neural network approaches, FoTEN incorporates a dedicated parallel Fourier encoder path that applies the Fast Fourier Transform (FFT) and its inverse (IFFT) operations inside the network, jointly modeling spectral magnitude and phase alongside temporal dynamics. Together, these two complementary paths capture both temporal interaction dynamics and spectral characteristics of texture-induced vibrations, improving reconstruction performance beyond what either domain achieves alone.

The FoTEN architecture builds on the Transformer framework, which has demonstrated strong performance in tasks such as language processing [17] and time-series analysis

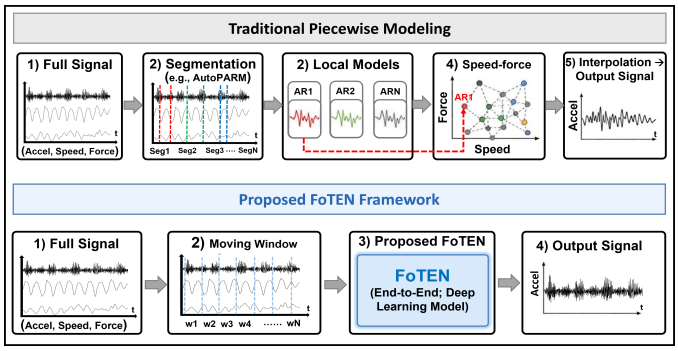


Fig. 1: Conceptual comparison of haptic texture modeling frameworks. (Top) Traditional piecewise modeling uses algorithmic segmentation (e.g., AutoPARM) and segment-specific AR modeling; this requires complex speed-force interpolation during rendering, which struggles with non-linear transitions. (Bottom) FoTEN learns a unified continuous representation, capturing interaction dynamics directly without discrete partitioning or external interpolation.

[18], and is commonly implemented using an encoder–decoder structure. In such architectures, the encoder processes the input sequence while the decoder generates variable-length output sequences. However, this sequence-generation capability is not required for haptic rendering. In our case, the objective is to predict the next vibration (acceleration) sample corresponding to the current user interaction, defined by the applied scanning speed and force. Accordingly, FoTEN adopts an encoder-only architecture and employs its encoder layers as lightweight feature extractors, similar to how CNN or LSTM layers are used to capture spatial and temporal features. Using these encoder layers, FoTEN constructs two parallel processing modules operating in the temporal and spectral domains. The temporal block processes raw acceleration signals in the time-domain, while the spectral block applies a Fourier-based decomposition into magnitude and phase, which are then processed in separate encoder layers. This dual-domain formulation captures complementary characteristics of texture vibrations. The temporal pathway models the evolution of interaction dynamics, while the spectral pathway captures the frequency structure and phase relationships that strongly influence perceived texture. Explicit phase-aware modeling improves spectral fidelity and enables the model to capture long-range dependencies that are difficult for purely time-domain approaches. The outputs of both modules, together with interaction parameters (speed and force), are fused to predict the next acceleration sample for vibrotactile feedback.

Through this design, FoTEN learns directly from continuous vibration signals without explicit segmentation and avoids the sequential overhead associated with recurrent or encoder–decoder architectures [15], [16]. This unified formulation improves generalizability and supports real-time haptic texture synthesis with high reconstruction fidelity and stability, as demonstrated through numerical, spectral, and perceptual analyses. The main contributions of this work are as follows:

- An encoder-only Transformer architecture that operates on fixed-size sliding windows, enabling efficient feature extraction while removing the need for handcrafted seg-

mentation and avoiding the complexity of a decoder.
- A phase-aware Fourier representation processed through separate encoder layers, allowing the model to capture amplitude patterns and phase relations explicitly, preserve spectral fidelity, and learn long-term dependencies.
- A computationally efficient and low-latency framework that achieves perceptually accurate real-time synthesis.

The paper is organized as follows. Section II provides a review of related work. Section III describes the data collection setup and dataset. The FoTEN framework is introduced in Section IV, and the rendering process is detailed in Section V. Section VI presents numerical evaluation, while Section VII reports perceptual validation. Section VIII discusses the findings, and Section IX concludes the paper.

II. Related Works

This section reviews related work in two parts. First, data-driven deep learning approaches for haptic texture modeling are discussed, highlighting their limitations in terms of computational efficiency and spectral modeling. Second, Transformer-based architectures for time-series data are reviewed, with a focus on encoder-only designs that motivate the proposed FoTEN framework.

A. Deep Learning for Haptic Texture Modeling

In recent years, the focus of texture modeling has shifted from traditional piecewise AR-based methods [12] to deep learning networks. Early work by Shin et al. [13] demonstrated the potential of data-driven approaches by decomposing acceleration signals into frequency bands as input features for neural network modeling. Similarly, Heravi et al. [14] leveraged high-resolution images and user interaction data, predicting the DFT magnitude of acceleration as model output and recovering the time-domain signal through a separate phase retrieval post-processing step. These approaches highlight that simple neural networks struggle to model raw acceleration signals directly, instead relying on frequency-domain representations as inputs or outputs, with phase information either discarded or recovered externally. In response, Joolee et al. [15] introduced a network combining CNNs and Bi-LSTM units to model vibrotactile feedback directly from interaction parameters in the time-domain, avoiding post-processing. However, the sequential nature of the LSTM component introduced computational overhead and the model lacked explicit spectral modeling. In contrast, FoTEN processes raw acceleration signals directly, incorporating FFT and IFFT operations inside the network through a dedicated Fourier encoder block, enabling joint temporal and spectral modeling without any post-processing step.

In a parallel but novel direction, Naeem et al. [19] introduced a text-driven generative framework that synthesizes both visual and haptic textures from natural language descriptions. Their system integrates Stable Diffusion for image generation and a regression-based model to estimate perceptual haptic attributes, which are then used in an interpolation-based AR texture authoring pipeline. This multimodal approach enables scalable texture creation without physical interaction data and marks a shift toward more accessible virtual haptics authoring.

Recently, HapInf, a Transformer-based sequence-to-sequence model inspired by the Informer architecture [18], was introduced for generating acceleration signals from interaction sequences. Although effective in reconstruction, the model was computationally intensive. To improve efficiency, the authors later introduced a low-delay haptic texture display framework (LDHTD) [20], using a student–teacher strategy, combining an LSTM-based encoder for action inputs with a lightweight Transformer-based decoder. In this strategy, a compact student model is trained to mimic a larger teacher network, with only the student used at inference. While this reduces delay, it increases training complexity and introduces a trade-off between inference speed and accuracy. Furthermore, the LSTM encoder reintroduces recurrent computation limiting parallelization, and the decoder provides no benefit in a one-step prediction setting. One-step prediction is well suited for haptic rendering as it allows the model to adapt immediately to rapidly changing interaction conditions such as scanning speed and force, without the latency of sequence generation. Additionally, LDHTD operates entirely in the time-domain, without explicit spectral modeling of vibration signals. FoTEN addresses these limitations through an encoder-only design that avoids recurrent computation, decoder overhead, and distillation complexity, while introducing a Fourier block that explicitly separates magnitude and phase for spectral modeling alongside temporal feature extraction. These diverse advances reflect the growing interest in leveraging deep learning for efficient, scalable, and perceptually rich haptic texture modeling and generation, and motivate the design choices behind FoTEN.

B. Transformers for time-series data modeling

Transformer-based networks have set new benchmarks in natural language processing tasks and computer vision applications [21], outperforming traditional methods such as RNNs, LSTMs, GRUs [22], and MLP based [23]. While these traditional recurrent models are effective for processing sequential data, they face challenges, including the vanishing gradient problem, which limits their ability to learn long-term dependencies, and their sequential processing nature, which leads to slow training and prediction speeds [24]. Originally proposed for seq2seq tasks (e.g., translation), Transformers address many of these challenges through their attention mechanism [17]. They consist of an equal number of layers in both the encoder and decoder blocks, usually six each. However, recent studies have modified transformer-based methods specifically for time-series applications, including time series forecasting [24], music generation [25], and vibration signal classification [26]. In these applications, researchers have often opted to remove the decoder and use only the encoder in transformer architectures. Additionally, they have made strategic modifications to the encoder's structure, such as adjusting the number of layers and tailoring input features to specific tasks. These changes streamline the model and improve both training and inference efficiency [24]–[26]. These findings motivate the adoption of an encoder-only Transformer architecture in this work, extended to process haptic vibration signals in both

temporal and spectral domains through a dual-path design.

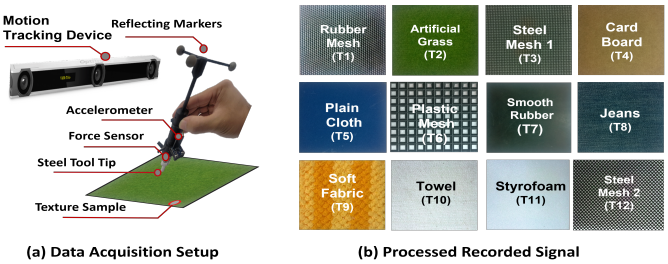
III. Data Acquisition

a) Apparatus: The data acquisition system used in this study, as shown in Fig.2(a), utilizes a custom rigid tool equipped with an interchangeable tip for interacting with textured surfaces. The body of the rigid tool is 3D printed using ABS-Plastic material, while the attached tool-tip is made of stainless steel and has a 2.0 mm diameter. A 3-axis accelerometer (ADXL335; Analog Devices) is attached to record interaction vibrations while reflective markers attached to the tool are used to track motion using Optitrack: V120. Additionally, it incorporates a force sensor (Nano17; ATI Industrial Automation) to measure 3-axis interaction forces. Both the force sensor and accelerometer are connected to the PC through a data acquisition card (USB-6351; National Instrument).

b) Texture Dataset: Twelve texture samples were selected to cover a broad range of perceptual characteristics relevant to haptic texture rendering, including roughness, hardness, and bumpiness, which are widely recognized as key dimensions of haptic texture perception [27], [28]. The selected samples span diverse surface conditions, from extremely rough and bumpy surfaces such as Steel Mesh to very smooth and compliant surfaces such as Smooth Rubber, ensuring a meaningful testbed for evaluating haptic texture modeling performance. These real texture samples are depicted in Fig.2(b). Each texture was mounted on a hard acrylic surface measuring 100x100x5 mm using liquid surface glue to avoid influence of underlying objects during data recording.

c) Data Collection and Pre-processing: For each texture, data was collected for 60 seconds through unconstrained manual stroking of the rigid tool over textured surfaces, following the unconstrained interaction protocol adopted in prior haptic texture modeling studies [10]–[12]. Each surface was stroked continuously with varying speed and applied force without lifting the tool, maintaining constant contact throughout the recording. Any recording in which surface contact was lost was discarded and repeated. Additionally, the first and last 2 seconds of each recording were discarded to exclude transient artifacts arising from initial tool contact and final release.

The 3D position data is captured at 120Hz used to estimate the scanning speed. The applied 3D-Force data is sampled at 10KHz and projected onto the normal direction of the contacting surface to estimate the scalar normal force. Acceleration data from the 3-axis accelerometer is captured at 1000Hz. All these signals are up/down-sampled at 1000Hz for synchronization. Furthermore, the recorded interaction signals (i.e., scanning speed and normal force) are low-pass-filtered at 25Hz to remove high-frequency noise while preserving the natural variation in scanning speed and force, as human hand motion during stroking does not exceed this frequency range [12], whereas the recorded acceleration signals are band-pass filtered from 20Hz to 1000Hz to reduce noise and to remove the gravitational component. Next, these 3-axis acceleration signals are mapped onto a single axis using the DFT321 algorithm, which can capture both the temporal information and spectral energy of all three axes [28].



(a) Data Acquisition Setup (b) Processed Recorded Signal

Fig. 2: (a) Data acquisition setup. (b) Real textures samples used in this study.

IV. Modeling Approach

This paper presents a transformer encoder-only architecture to predict the acceleration signals from historical data (i.e., predefined length of previous acceleration signal, interaction speed, and force). While prior Transformer-based approaches such as HapInf were computationally expensive due to their encoder-decoder architecture, LDHTD attempted to address this using a student-teacher distillation strategy, increasing training complexity, requiring an additional student model, and still introducing a trade-off between inference speed and signal accuracy. FoTEN instead achieves low latency through architectural design alone, without requiring distillation.

The encoder-only structure is chosen over the encoder-decoder architecture for its efficiency and suitability for the fixed input and output lengths, obviating the need for a decoder (see Sect. II-B) The proposed FoTEN architecture, illustrated in Fig. 3, consists of two parallel blocks: the Encoder block (TEN) and Fourier-encoder block. A comprehensive explanation of the model's inputs and each block's function is provided below.

a) Model Input: The proposed network is designed to predict the acceleration $a[n]$ at time n by taking a fixed-length m sequence of previous acceleration a, scanning speed v, and applied force f as input. The input for each time point n is derived using a single-step sliding window mechanism, which can be represented as:

$$\begin{aligned}\text{Input}_{\text{Sequence}}[n] = (a_{n-m},\ldots,a_{n-1}, \quad v_{n-m},\ldots,v_{n-1},\\ f_{n-m},\ldots,f_{n-1}),\end{aligned} \tag{1}$$

where m represents the length of the sliding window. This approach allows the model to consistently use recent data for predictions and effectively adapt to data shifts over time.

b) Position Encoding: Position encoding is crucial for non-recurrent models like transformers to preserve the temporal sequence of input elements. Unlike recurrent models such as LSTMs, which process data sequentially and inherently capture the order, transformers process data in parallel, leaving the network without a natural way to track positional order. To address this, position encoding is added to the embedding layers to preserve the order of input elements [17].

$$\begin{aligned}PE_{(pos,2i)} &= \sin\left(pos \cdot 10000^{-2i/d_{\text{model}}}\right),\\ PE_{(pos,2i+1)} &= \cos\left(pos \cdot 10000^{-2i/d_{\text{model}}}\right).\end{aligned} \tag{2}$$

where pos represents the position within the sequence, i is the dimension index, and d_{model} denotes the size of the

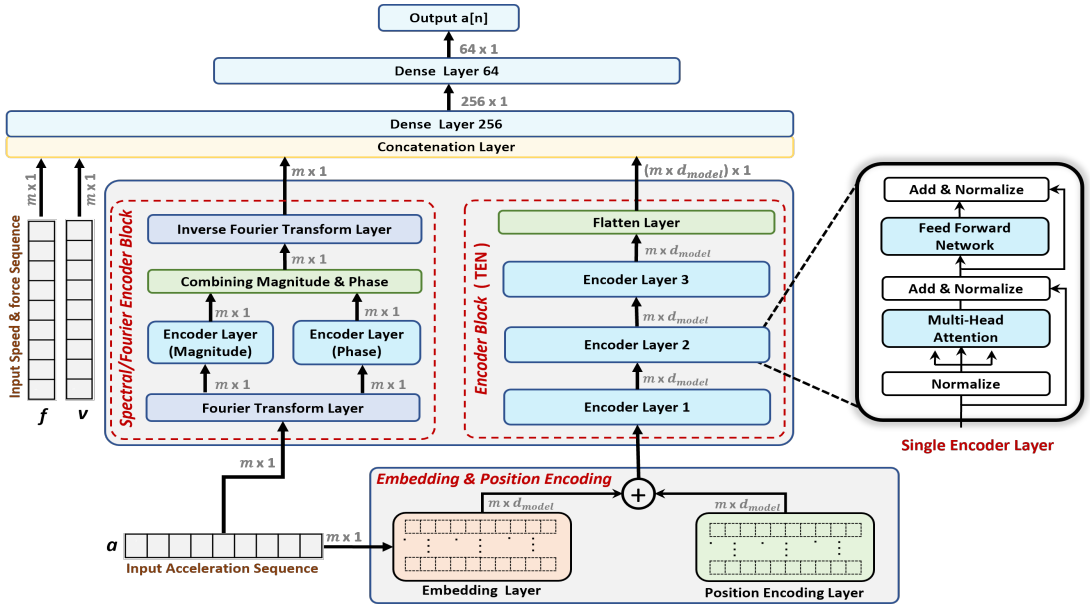


Fig. 3: Structure of the proposed Fourier Enhanced Transformer Encoder Network (FoTEN).

embedding space. These position vectors are added to the embedding vectors before forwarding to the encoder block.

c) Encoder Block: The encoder block comprises three standard Transformer encoder layers. Each encoder layer is identical in structure to the one proposed by [17] and can be seen in Fig. 3. The input to this encoder block is a sequence of the acceleration signal, which is first transformed and then augmented with embeddings and positional encoding, resulting in a tensor of shape $m \times d_{\text{model}}$ that is fed to the first encoder layer. The encoder block preserves the sequence length and produces a temporal embedding $E_t \in \mathbb{R}^{m \times d_{\text{model}}}$.

Initially, the embedded input sequence is passed to the first encoder layer where they undergo normalization followed by a multi-head attention layer. This operates with several scaled dot-product attention units in parallel, referred to as "heads." Each head focuses on different sub-spaces of the sequence independently by generating unique attention weights. The transformation is defined as:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V, \tag{3}$$

where X is the input, and W^Q, W^K, W^V are learned weight matrices.

The multi-head attention combines these heads as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{4}$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \tag{5}$$

Here, W^O is a learned weight matrix that integrates the attention results from all heads. This architecture allows the model to capture rich representations of the input sequence, followed by residual connections, normalization, and a feed-forward network (FFN) with a hidden layer set to 128 units. The hyperparameters were fixed at $d_{\text{model}} = 32$, heads $= 4$, and three encoder layers after exhaustive evaluation. For further details on Transformer encoders, refer to [17].

d) Fourier Encoder Block: In the Fourier encoder block, the input acceleration signal $a[n]$ with dimensions $m \times 1$, is transformed into the frequency domain using the Fast Fourier Transform (FFT), an efficient implementation of the Discrete Fourier Transform (DFT):

$$Z(k) = \sum_{n=0}^{m-1} a[n]\, e^{-i2\pi kn/m}, \tag{6}$$

where $a[n]$ is the acceleration signal at time index n, m is the number of samples, k is the frequency index, and $Z(k)$ is the complex coefficient of the frequency component k. The FFT yields m complex coefficients, each linked to a frequency. The complex coefficients are then decomposed into magnitude and phase components to enable separate representation learning in the spectral domain:

$$M(k) = |Z(k)|, \quad \phi(k) = \arg(Z(k)), \tag{7}$$

both retaining dimensions of $m \times 1$. Magnitude and phase are processed in separate encoder layers to enhance the representation of frequency-domain features.

The encoded components are recombined as:

$$\hat{z}(k) = \hat{M}(k) \cdot e^{i\hat{\phi}(k)}, \tag{8}$$

where $\hat{M}(k)$ and $\hat{\phi}(k)$ are the encoded magnitude and phase, respectively.

Finally, the enhanced frequency-domain representation is transformed back into the time-domain using the Inverse Fast Fourier Transform (IFFT), corresponding to the Inverse Discrete Fourier Transform (IDFT):

$$a_{\text{spec}}[n] = \frac{1}{m} \sum_{k=0}^{m-1} \hat{z}(k)\, e^{i2\pi kn/m}. \tag{9}$$

The time-domain sequence $a_{\text{spec}} \in \mathbb{R}^{m \times 1}$ is then mapped to a spectral embedding via a position-wise linear projection:

$$E_s = a_{\text{spec}} W_s + b_s \tag{10}$$

where $W_s \in \mathbb{R}^{1 \times d_{\text{model}}}$ denotes the learnable projection matrix and $E_s \in \mathbb{R}^{m \times d_{\text{model}}}$ represents the resulting spectral embedding.

This formulation leverages magnitude–phase separation and recombination to model phase and improve spectral fidelity. Positional encoding is omitted in this spectral branch because FFT bin indices provide ordering. The resulting spectral embedding E_s is forwarded to the fusion module.

e) Network Training: The FoTEN model was trained on an NVIDIA RTX 3070 using Keras/TensorFlow 2.8. The input acceleration sequence ($m \times 1$) is routed to the spectral branch (without positional encoding) and to the temporal branch (with positional encoding). Both branches produce embeddings $E_s, E_t \in \mathbb{R}^{m \times d_{\text{model}}}$. The synchronized interaction signals (speed v and force f) are provided as sequences $v, f \in \mathbb{R}^{m \times 1}$. Features are fused along the last dimension:

$$X_{\text{fuse}} = [\, E_t \mid E_s \mid v \mid f \,] \in \mathbb{R}^{m \times (2d_{\text{model}}+2)}. \tag{11}$$

For one-step-ahead prediction, the last token of the fused sequence, corresponding to time step $n-1$, i.e., $x_{n-1} = X_{\text{fuse}}[m, :]$, is selected and passed through two dense layers (256 and 64 units), followed by a linear regression head that outputs the next-sample acceleration $a[n] \in \mathbb{R}^1$. The model is trained with the Adam optimizer using a learning rate of 0.001 for up to 100 epochs, with early stopping (patience 10) and a batch size of 16 across all experiments. Loss functions include MAE, MSE, and Huber loss (details in Sect. VI). ReLU is used as the activation function, and dropout with a rate of 0.2 is applied for regularization.

V. Texture Rendering

a) Signal Synthesis: The goal of the rendering algorithm is to synthesize acceleration sequences based on the user's interaction, such as stroking a pen on a tablet screen. The algorithm generates vibration output $a[n]$ at time n by using three inputs: stroking speed v, applied force f, and previous acceleration a, each of size m as defined in Eq. 1. At the initial contact, there is no acceleration history available for the model's input of length m. For v and f, data can be captured as soon as interaction begins, reflecting real user behavior. The challenge lies in how to initialize a. Prior studies proposed two main strategies. [13] initialized the first acceleration window using a randomized sequence, which is computationally simple but perceptually inconsistent. By comparison, [15], [20] interpolated stored acceleration segments recorded at constant speed and force. This method yields a deterministic initial sequence but requires storing large sets of signals and introduces computational overhead. Moreover, it depends on data collected under fixed interaction parameters, which is misaligned with the goal of modeling unconstrained natural data and was contextually suited for their applications.

Building on these techniques, the proposed approach employs a warm-start strategy with two main goals: to achieve stable and rapid convergence, and to provide perceptually smooth feedback at the beginning of stroking. It is hypothesized that by starting the model near realistic dynamics rather than from random or interpolated sequences, the transient mismatch can be reduced and convergence accelerated. To realize this, the method combines the efficiency of fixed initialization with the realism of parameter-based signals. Specifically, the first acceleration window is deterministically generated using values representative of initial stroking. To select these parameters, raw interaction data were analyzed and it was found that early contacts exhibited limited variation, with mean values of about 45 mm/s for speed and 0.8 N for force. From these values, an initial acceleration sequence of length m was synthesized and stored as model input to provide a lightweight, repeatable, and perceptually natural start. After this short warm-start window, FoTEN updates acceleration from its own outputs, while v and f are refreshed in real time within a moving window to reflect ongoing interaction.

This initialization avoids the instability of randomization and the overhead of interpolation while ensuring rapid stabilization. Its effectiveness is demonstrated in both numerical and perceptual results: the model converges within 100 iterations at 1 kHz (0.1 s) even when evaluated with recorded signals at unconstrained speed and force (Fig. 6), highlighting both the fidelity of the initialization approach and the generalizability of the model. Please refer to Sec. VI for quantitative analysis and Sec. VII for perceptual validation.

b) Rendering Hardware and Software: The complete illustration of the texture rendering setup is depicted in Fig. 8. A touch tablet PC (Surface Pro 4; Microsoft) equipped with an active digital stylus (Surface Pen; Microsoft, report rate:$\sim$200Hz) is selected as the interface for rendering. The vibration feedback for the virtual texture was generated using a high-bandwidth voice-coil actuator (Haptuator MM1C; Tactile Labs) attached to the stylus pen near the tip with plastic zipties. The haptuator is driven by an NI-DAQ device (USB-6351; National Instruments) at 1 kHz, with an analog amplifier regulating the signal gain and bridging the DAQ and actuator.

A user interface (UI) has been developed to render texture. The UI detects three inputs upon interaction: the texture type to load the initial sequence, the sliding speed of the stylus, and the applied pressure. The speed v calculations are based on the x and y positions captured from the screen during interactions within the 2D space, specifically when the user interacts with the area displaying the texture. Meanwhile, the force is calculated using pressure values from Eq.12, derived by recording the generated force through a force sensor placed under the tablet. The pressure is recorded with a step of 0.1 and is later modeled using an exponential equation. This step is crucial because the tablet does not provide Force (N) instead normalized pressure values ranging from 0 to 1.

$$\mathrm{F} = 0.13 \cdot \exp(3.32 \cdot p) \tag{12}$$

where F is the force in Newtons and p is the pressure value obtained from the software. Furthermore, it is noted that the speed and force values are first up-sampled to 1kHz, as the model requires, and then filtered at 20Hz. This step ensures that the sampling rate matches that of the data used for model training and is filtered to remove DC position offset. Finally, the acceleration output from the model is denormalized using pre-stored statistical parameters and sent to the NI DAQ after a dynamic compensation filter [29] to mitigate the haptuator's natural frequency response. This prepares it for vibrotactile feedback using haptuator.

VI. Model Prediction Accuracy Measures

Tactile signals from twelve different textures were collected to evaluate the proposed model's performance (see Sec.III). Each texture dataset comprised 60-second recordings at 1kHz (60,000 samples), with the first and last 2 seconds removed to reduce tapping artifacts at the start and end of each recording, following standard practice in haptic texture data processing [12]. The data was divided into 2000 samples and regrouped based on speed and force regions, following the method in [14]. This regrouping strategy, adopted from prior work, is designed so that the model learns interpolation rules across the interaction space, where the test set contains speed and force conditions that lie within the range observed during training rather than outside it, providing a meaningful evaluation of the model's interpolation capability across natural interaction variability. The overall split ratio was set to training (60%), validation (20%), and test sets (20%) for each texture (see Fig. 4). The training and validation datasets were used for training, while all reported numerical results, including error metrics and ablation study outcomes, are computed exclusively on the test subset, which was not used during training or validation.

a) Error Metrics : Comparisons between synthesized acceleration signals and actual recorded acceleration waveforms were performed in both the time and spectral domains to gauge the proposed model's performance. In the , prediction accuracy was quantified using mean absolute error (MAE). For spectral domain comparison, the Hernandez-Andres Goodness-of-Fit Criterion (GFC) was utilized to measure the degree to which the reconstructed signal matches with the actual signal. The GFC score values range from 0 to 1, where a value of one is considered a perfect reconstruction [30]. Mathematically, it can be computed as:

$$GFC = \frac{\left\| \sum_i A_d\left(f_i\right) A_m\left(f_i\right) \right\|}{\sqrt{\left\| \sum_j \left[A_d\left(f_j\right)\right]^2 \right\|} \sqrt{\left\| \sum_k \left[A_m\left(f_k\right)\right]^2 \right\|}}, \tag{13}$$

where $A_d(f_i)$ and $A_m(f_i)$ are the DFT amplitudes at a frequency f_i of the measured and reconstructed signals, respectively. These metrics were selected following prior studies that employ them for haptic texture modeling evaluation [30], [31].

b) Finding optimum sequence size and loss function: As part of the ablation study, the influence of input sequence size and loss functions on model performance is examined. Identifying the temporal length of the input sequence is crucial for accurate predictions [32], and selecting an effective loss function is essential for improving training outcomes [33]. Importantly, loss functions guide the weight updates during training, while error metrics evaluate model performance on test datasets.

For this study, various input sequence lengths were tested m (see Eq. 1) and three loss functions: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Huber Loss (HL). MAE is less sensitive to outliers, treating all errors equally, whereas MSE emphasizes small residuals but penalizes large errors heavily. The Huber loss combines these properties and is defined as:

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2 & \text{if } |r| \le \delta, \\ \delta|r| - \frac{1}{2}\delta^2 & \text{otherwise,} \end{cases} \tag{14}$$

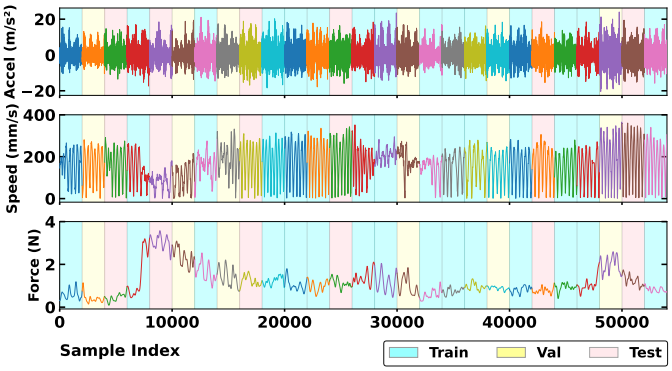


Fig. 4: Recorded acceleration with corresponding speed and force signals of Steel Mesh (T3) sample, partitioned into fixed-length segments (2,000 samples, displayed with different wave colors) and allocated to training (cyan), validation (yellow), and test (red) sets, as indicated by the background color.

where $L_\delta(r)$ is the Huber loss, r denotes the residual error, and δ is a threshold parameter. For $|r|$ below δ, HL behaves like MSE, while for larger errors it behaves like MAE, providing a balanced trade-off between sensitivity and robustness [33]. This robustness property is particularly beneficial when training on unconstrained interaction data, where natural variability in scanning speed and applied force can introduce occasional large residuals in the recorded acceleration signals.

The results of 216 experiments are summarized in Fig. 5 as averaged MAE for each of the 12 textures across six sequence sizes ($m = 5, 10, 15, 20, 25, 30$) and the three loss functions. Multiple values of δ were explored for HL, with $\delta = 1.6$ and $m = 20$ empirically identified as optimal, achieving the lowest mean MAE of 0.149. This configuration provided stable learning by preserving high-frequency detail while limiting the influence of large residuals. Huber loss performed better than MAE or MSE because it retains sensitivity to small residuals, which is essential for reproducing fine texture features, while reducing the destabilizing effect of sharp transients. Such transients are common in bumpier textures, for instance, plastic mesh, where local surface irregularities and variable force can create sudden spikes in the acceleration signal. Under these optimal conditions, Fig. 6 shows the time-domain comparison of synthesized and recorded signals for the first 400 samples, while Fig. 7 presents the PSD of the complete test signals, demonstrating accurate reconstruction across all 12 textures.

c) Comparison with other approaches: The signal reconstruction accuracy of the proposed framework (FoTEN) was also evaluated against various established methods. These methods include traditional AR based technique [12], a neural network strategy (FNN) [13], an advanced deep learning based spatio temporal network (DSTN) [15], the low delay haptic rendering framework (LDHTD), which employs a transformer-based architecture [20], and GAN based technique [34].

The comparison results with other approaches for each texture, along with the experimental settings, are summarized in Table I. FoTEN achieves the lowest mean MAE of 0.149 and the highest mean GFC of 92.15%, showcasing its effectiveness in both time and spectral domains. In contrast, the piece-wise AR and FNN approaches showed less favorable outcomes

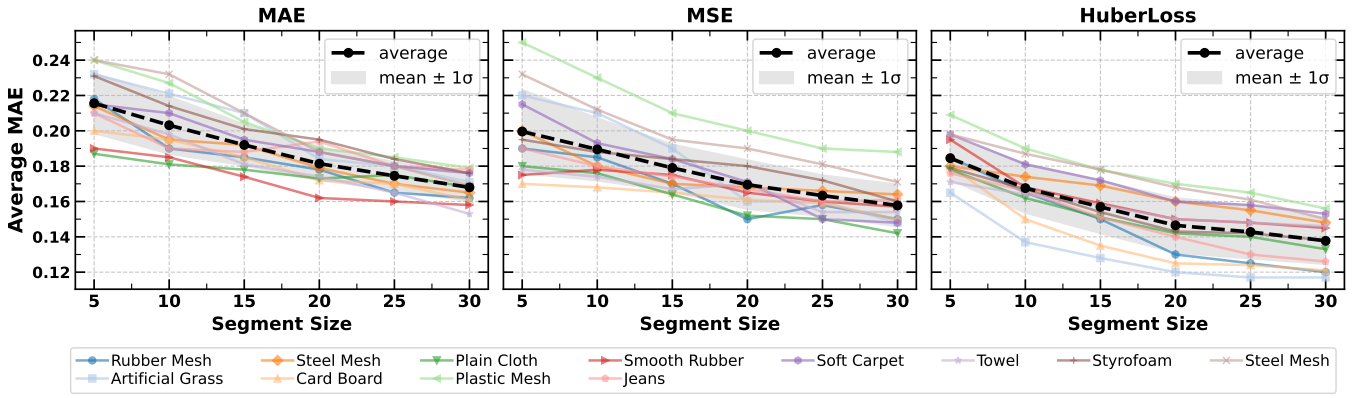


Fig. 5: The comparison of model performance for various sequence sizes (i.e., 5, 10, 15, 20, 25, and 30) and loss functions (MAE, MSE, Huber) as part of the ablation study on the test dataset.

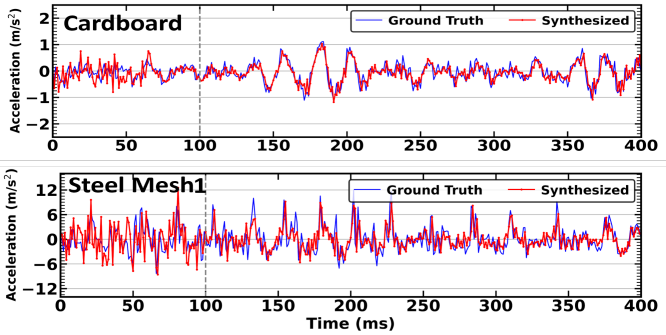


Fig. 6: Time-domain comparison of synthesized and recorded acceleration signals on unseen test data, zoomed into the first 400 samples (0.4 s). The vertical line marks the warm-up period of 100 iterations (0.1 s), showing that the model converges even before this point.

in the time and frequency domains, respectively. LDHTD (91.02%) and DSTN (90.33%) achieved competitive reconstruction accuracy, while the GAN-based approach lagged behind other DL methods. Notably, FoTEN surpassed most methods across different textures, except for Soft Carpet (T9), where LDHTD achieved slightly better accuracy.

d) Ablation study on architectural variants: Table II summarizes the ablation study on several variants of the architecture. The baseline encoder-only model (TEN) showed the highest errors, indicating that temporal features alone are insufficient. Adding the spectral branch improved performance across MFCC, Wavelet, and FFT, with FFT giving the largest gain through a smaller optimal window and the highest reconstruction accuracy. Wavelet performed best with MSE, whereas MFCC and FFT were optimal with Huber loss ($\delta = 1.6$); TEN reached its best result with Huber loss ($\delta = 2.0$). All variants used the core Transformer settings described in Sect. IV. A FoTEN+Decoder (FoTEN-D) variant was also tested ($d_{\text{model}} = 32$, 4 heads, query token = 1). Although it used the same FFT-based representation as FoTEN, it showed higher MAE and lower GFC. In a one-step prediction setup, the decoder receives only a single query token and cannot perform its intended sequence modeling, so the added parameters do not translate into accuracy gains.

TABLE I: Comparison of error metrics (MAE and GFC) across existing methods and textures. The parenthetical information after each study indicates: (Year, Modeling Type, Segmentation Technique, Modeling Features), where S = Stochastic, NN = Neural Network, DL = Deep Learning, AP = AutoPARM, CSF = Constant Speed & Force, SW = Sliding Window, RA = Raw Acceleration, RA+FD = Raw Acceleration + Frequency Decomposition, RA+S = Raw Acceleration + Spectrogram, RA+FT = Raw Acceleration + Fourier Transform.

Study	Metric	Textures												Mean
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	
Piece-wise AR [12]	MAE	0.48	0.36	0.39	0.41	0.27	0.53	0.42	0.35	0.38	0.44	0.31	0.46	0.400
(S, AP, RA)	GFC	87.56	80.58	86.58	85.16	91.18	88.79	85.24	82.35	87.42	84.96	89.15	86.38	86.28
FNN [13]	MAE	0.41	0.28	0.46	0.29	0.39	0.47	0.36	0.33	0.44	0.31	0.37	0.42	0.378
(NN, CSF, RA+FD)	GFC	89.31	90.88	86.37	86.14	85.08	89.92	88.41	89.55	86.92	87.26	85.94	88.12	87.83
DSTN [15]	MAE	0.15	0.18	0.27	0.17	0.15	0.27	0.19	0.21	0.24	0.20	**0.16**	0.22	0.201
(DL, CSF, RA)	GFC	91.83	89.96	90.54	88.18	90.22	90.71	91.02	89.34	90.15	89.96	**91.14**	90.88	90.33
LDHTD [20]	MAE	0.17	0.19	0.17	0.14	0.17	0.29	0.19	0.13	**0.14**	0.16	0.20	0.18	0.178
(DL, SW, RA)	GFC	89.24	92.10	91.47	92.04	91.26	88.98	90.34	91.05	**92.20**	90.42	92.15	90.75	91.02
GAN-based [34]	MAE	0.23	0.28	0.27	0.29	0.32	0.37	0.26	0.31	0.28	0.34	0.30	0.33	0.298
(DL, CSF, RA+S)	GFC	90.85	89.39	88.50	90.54	83.21	90.71	89.34	88.72	87.95	89.86	84.56	90.12	88.65
FoTEN (ours)	MAE	**0.13**	**0.11**	**0.16**	**0.13**	**0.14**	**0.19**	**0.15**	**0.12**	0.17	**0.14**	0.16	**0.17**	**0.149**
(DL, SW, RA+FT)	GFC	**90.83**	**93.82**	91.17	**93.83**	**92.17**	**91.84**	**92.35**	**93.24**	91.62	**92.88**	90.08	**91.95**	**92.15**

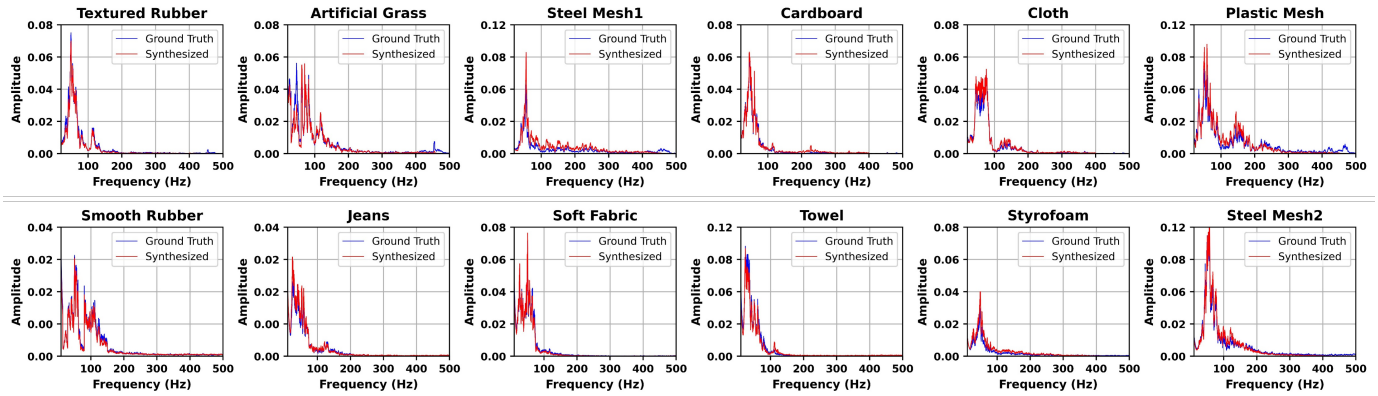


Fig. 7: Spectral-domain analysis comparing recorded and synthesized acceleration signals for all 12 textures on test data.

TABLE II: Comparison of the proposed model variants.

Model Variant	Input Size	Mean MAE	Mean GFC (%)
TEN (Encoder Block)	25	0.318	86.77
TEN + MFCC	30	0.281	90.28
TEN + Wavelet	30	0.284	86.18
FoTEN (TEN + FFT)	**20**	**0.149**	**92.15**
FoTEN-D	20	0.193	90.64

e) Computational Efficiency: An important aspect of deep learning-based real-time rendering is the model's inference time. The efficiency evaluation includes established baselines such as DSTN [15], LDHTD [20], and GAN-based [34], followed by FoTEN and its variants (TEN and FoTEN-D). Results were averaged over 1000 inference iterations on both CPU and GPU, with performance metrics summarized in Table III. TEN shows the lowest latency due to its minimal architecture, FoTEN provides significantly higher accuracy (see Table II) while remaining fast, and FoTEN-D incurs extra decoder overhead without offering benefits.

FoTEN consistently outperforms the baselines. Compared to GAN, it offers 44.2% faster GPU and 53.2% faster CPU inference, reflecting the cost of training and running both generator and discriminator networks. Against DSTN, FoTEN achieves 30.4% faster GPU inference, 41.3% faster CPU inference, and 76.9% faster training per epoch, largely because DSTN's Bi-LSTM layers process inputs sequentially and limit parallelization. Relative to LDHTD, which also uses a Transformer-based architecture, FoTEN still shows 15.0% faster GPU and 11.9% faster CPU inference. Overall, the encoder-only, parallelizable design provides both high accuracy and strong efficiency for real-time haptic texture rendering.

TABLE III: Efficiency comparison of FoTEN and Existing Deep Learning based approaches.

Models	Input Seq. Size	Inference GPU(ms)	Inference CPU(ms)	Training One Epoch
DSTN [15]	40	1.38	1.89	44.35
LDHTD [20]	20	1.13	1.26	27.45
GAN [34]	56	1.72	2.37	85.71
TEN (ours)	25	0.87	1.07	8.92
FoTEN (ours)	20	0.96	1.11	10.26
FoTEN-D (ours)	20	1.16	1.39	14.81

VII. Perceptual Performance

A total of twenty-one participants (17 male and 4 female), aged 22-48 years (mean 34.6), took part in this study. They compared virtual textures with their corresponding real counterparts and rated their similarity on a scale of 0 to 100 (100 being completely the same). Each participant rated all twelve textures using the AR model [12] as a traditional baseline method, alongside three advanced deep learning-based techniques: DSTN [15], LDHTD [20], and FoTEN. Consequently, users were presented with twelve virtual-real comparisons, repeated across the four methods, for their assessment.

a) Procedure: The complete setup can be seen in Fig. 8, where users were seated in front of a table playing white noise to minimize environmental noise. The table was divided into two parts with a divider to avoid visual cues. Users faced a screen running a GUI to record responses on the left side while experiencing real/virtual textures on the right side through a customized tablet PC. The tablet screen was modified using styrofoam, which included two cutouts of identical size to facilitate the comparison between virtual and real textures; real on the right and virtual on the left. Users experienced virtual textures with a stylus equipped with haptuator for vibration cues while for the real textures, a custom tool with the same tool-tip used for data recording was provided to maintain consistency in the modeled tactile feedback. They were asked to freely explore and switch between the real and virtual textures until they were satisfied. The experimenter assisted by guiding their hand and the tools while switching between textures. The order of the presented stimuli was

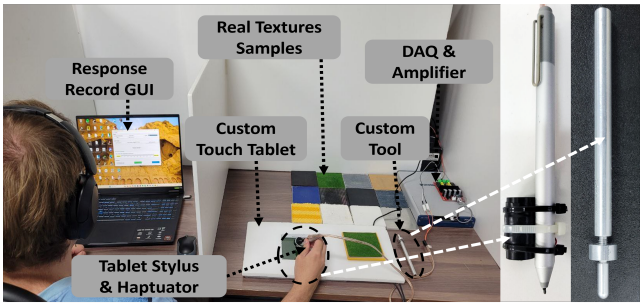


Fig. 8: Experimental setup for the user study and the hand-held tools used in the experiments.

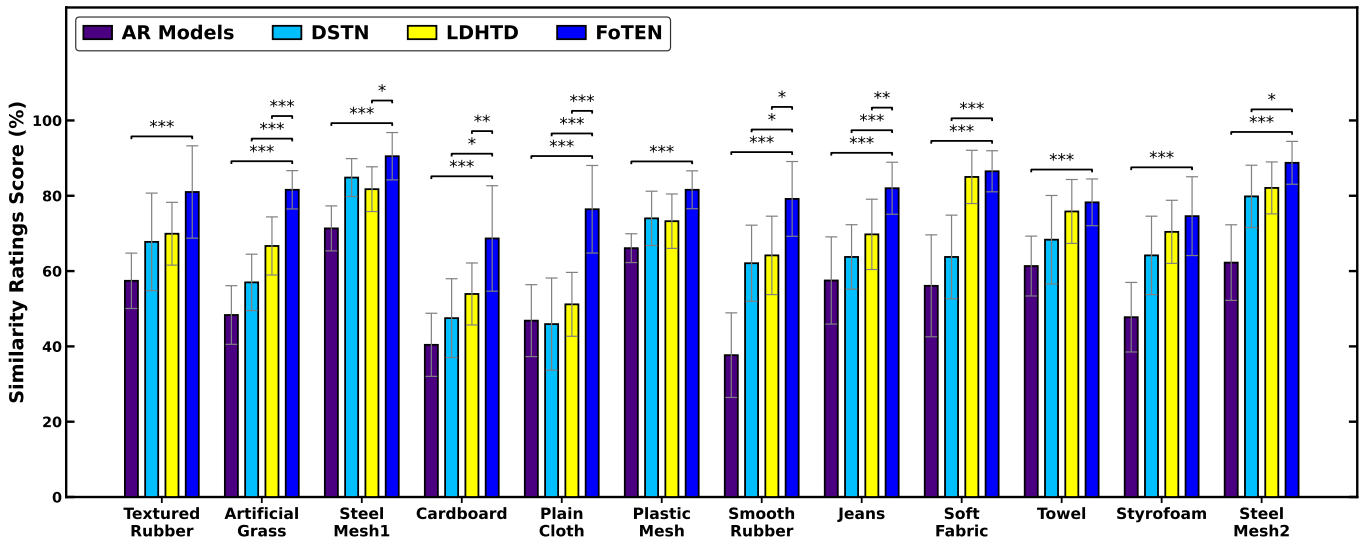


Fig. 9: Perceptual comparison of FoTEN with classical (AR Model) and deep learning (DSTN and LDHTD) baselines across 12 textures. Bars show mean $\pm$ SD across $n = 21$ participants on a 0 to 100 similarity scale. Brackets indicate within texture paired comparisons between FoTEN and each baseline method. Statistical significance is marked ($***p < .001$, $**p < .01$, $*p < .05$).

randomized across textures, algorithms, and participants and was determined before the experiment. The experiment took an average of 90 minutes per participant, and each received approximately $12 as a token of appreciation.

b) Results: The perceptual evaluation yielded 1,008 similarity ratings ($21 \times 12 \times 4$) from $n = 21$ participants, assessing four rendering methods across 12 virtual textures. Fig. 9 presents mean similarity ratings by method and texture. FoTEN consistently produced the highest similarity ratings, while AR was reliably lowest. DSTN and LDHTD occupied an intermediate tier with more variability across textures.

To examine the effect of rendering method, a one-way repeated measures ANOVA was conducted on participant-level ratings averaged across textures. The main effect of Method was statistically significant, $F(3, 60)$, $p < .001$, indicating robust differences in perceived similarity across methods.

The analysis was extended using a two-factor repeated measures ANOVA with Method and Texture as within-subject factors, applying Greenhouse Geisser corrections for sphericity. All main effects and the interaction were significant: Method, $F(3, 60)$, $p_{\mathrm{GG}} < .001$; Texture, $F(11, 220)$, $p_{\mathrm{GG}} < .001$; and Method $\times$ Texture, $F(33, 660)$, $p_{\mathrm{GG}} < .01$. These findings confirm that rendering method significantly impacts perceptual similarity, texture-specific difficulty is non-uniform, and benefit of specific methods depends on the surface being simulated.

Pairwise comparisons used within subject t tests with familywise error control per texture. With Bonferroni correction, FoTEN exceeded AR on all 12 textures ($p_{\mathrm{corr}} \in [2.0 \times 10^{-7}, .001]$). Against DSTN, FoTEN was significant on 8 of 12 textures: Artificial Grass, Steel Mesh1, Cardboard, Plain Cloth, Smooth Rubber, Jeans, Soft Fabric, Steel Mesh2 ($p_{\mathrm{corr}} \in [1.4 \times 10^{-5}, .012]$; $\Delta\mathrm{mean} \in [9.75, 31.33]$). Against LDHTD, FoTEN was significant on 8 of 12 textures: Artificial Grass, Steel Mesh1, Cardboard, Plain Cloth, Plastic Mesh, Smooth Rubber, Jeans, Soft Fabric ($p_{\mathrm{corr}} \in [4.0 \times 10^{-5}, .022]$; $\Delta\mathrm{mean} \in [8.75, 25.25]$). The remaining FoTEN comparisons did not survive correction on Textured Rubber, Towel, or Styrofoam, which is consistent with the significant Method $\times$ Texture interaction. AR was frequently lower than both DSTN and LDHTD, while DSTN and LDHTD rarely differed after correction. For completeness, results under Holm correction were qualitatively identical, with a small number of marginal cases becoming significant at $\alpha = .05$.

Overall, FoTEN achieved the highest perceptual fidelity across textures and participants, AR was consistently lowest, and DSTN and LDHTD occupied an intermediate band with few reliable differences between them. The uniform advantage of FoTEN over AR on every texture, along with frequent advantages over DSTN and LDHTD and a significant Method $\times$ Texture interaction, indicates that FoTEN is generally superior, with effect magnitude that depends on the surface.

VIII. Discussion

The results from both numerical and perceptual evaluations demonstrate that FoTEN provides clear advantages in haptic texture modeling while also exposing challenges that remain open in the field. The dual-path design, with temporal and spectral processing, improved the stability and coherence of synthesized vibrations. These improvements were reflected in the lowest mean error (MAE = 0.149) and highest spectral fidelity (GFC = 92.15%) across twelve textures (Table I). In perceptual evaluations, these numerical gains translated into experiential benefits, as FoTEN achieved the highest average similarity rating ($80.14\% \pm 10.67\%$) and avoided artifacts reported for DSTN and LDHTD. Together, these findings suggest that separating frequency components is an effective inductive bias for capturing texture-specific vibrations.

Texture-dependent differences further highlighted both the strengths and limitations of the approach. Coarse surfaces such as Steel Mesh, Artificial Grass, and Jeans achieved the highest similarity ratings, with participants consistently describing them as realistic and sharp. These gains can be attributed

to the complementary roles of FoTEN's dual-block design: the temporal block stabilized vibration continuity, while the spectral block, with magnitude–phase separation, preserved high-frequency detail. Training on unconstrained stroking data, where scanning speed and applied force naturally fluctuated (Sec. III), further improved robustness by enabling the model to generalize across natural interaction variability rather than artificially controlled signals. The largest improvements over DSTN and LDHTD were observed in textured surfaces with broad spectral components, where this combination of temporal stability and spectral fidelity was particularly effective.

In contrast, softer or slippery textures such as Smooth Rubber and Styrofoam received relatively lower ratings across all rendering algorithms. Even though FoTEN reconstructed these signals with high GFC values (greater than 90%), the actuator was unable to transmit subtle variations in friction or compliance. This divergence points to a perceptual bottleneck: while the model accurately reconstructs the spectral envelope and maintains consistent vibration flow, the stylus-based actuator cannot replicate fine-grain tactile cues such as stick–slip or softness. Similar issues have been reported in previous studies [10], suggesting that these mismatches originate from hardware constraints rather than limitations in signal modeling. From a signal-processing perspective, smooth textures typically produce very low-amplitude vibrations with weak spectral structure. Such signals contain fewer distinctive features and are therefore more sensitive to noise or small prediction deviations. While rough textures contain strong high-amplitude spectral components that dominate perception, smooth textures rely on subtle vibration cues that are more difficult to reproduce accurately. As a result, small spectral differences may be perceptually amplified, which partly explains the reduced similarity ratings observed for these surfaces.

The role of initialization was also evident in perceptual outcomes. By synthesizing a short initial sequence under low speed and force conditions, FoTEN's warm start mechanism avoided glitched or delayed sensations that often accompany random or stored-segment initialization [13]. Participants reported that rendering began smoothly, with no abrupt artifacts, and predictions stabilized within 100 ms (see Fig. 6). This approach reduced storage demands while ensuring perceptual continuity. However, in an ideal setup this start would also be fully online, and the proposed design can support that later. Future studies may explore lightweight models for generating texture-specific initial sequences or alternative designs that eliminate the need for explicit initialization.

Computational efficiency was another central design goal. FoTEN achieved sub-millisecond inference on GPU and low latency on CPU, outperforming GAN, DSTN, and LDHTD across all metrics (Table III). The encoder-only streaming design avoided decoder-related exposure bias, while omitting positional encoding in the spectral path reduced overhead. Together, these modifications lowered training time per epoch by more than 75% compared to DSTN, confirming FoTEN's practicality for interactive applications where large texture libraries must be rendered in real time. GAN-based models, while useful for generating novel textures [34], are computationally demanding due to their adversarial training paradigm, which limits their suitability for real-time use. Recurrent designs such as DSTN effectively capture temporal dependencies but suffer from sequential latency. LDHTD alleviated this with a multimodal Transformer-based design, yet FoTEN's encoder-only, dual-path structure achieved a more favorable balance of accuracy and efficiency. This comparison suggests that while GANs excel at generative diversity and recurrent models capture sequential patterns, FoTEN's streamlined architecture is particularly well suited for real-time deployment.

Although FoTEN delivers significant improvements, certain limitations remain. As seen in the perceptual evaluation (Fig. 9), smooth and compliant textures remain difficult to reproduce accurately. This limitation is consistent with prior research [14], [15], suggesting that signal modeling alone cannot fully resolve the perceptual gap. Another practical consideration concerns the operational ranges of interaction parameters. During rendering, the system operates within the scanning velocity and applied force ranges observed during data collection, with upper limits of approximately 350 mm/s for speed and 4 N for force. These boundaries were applied during rendering to maintain consistent comparison conditions across all evaluated models and to ensure stable system operation. However, interaction speeds for smooth textures may sometimes exceed these ranges during natural exploration, which may further influence perceptual accuracy for such surfaces. Future work should therefore explore integration with advanced actuation technologies/algorithms capable of modulating friction and compliance in addition to vibration. In addition, failure scenarios may occur when interaction conditions fall outside the training distribution or when abrupt changes in speed and force occur. Such situations may arise due to natural variability in human exploration, where each user interacts with the surface differently. Consequently, certain interaction patterns may deviate from those observed during training, which can reduce prediction stability under these conditions. The current reliance on a warm start initialization, while effective, could be reduced or eliminated through alternative model designs. Finally, the framework still assumes fixed tool geometry and rigid contact conditions. Extending FoTEN to support variable tools, user-specific adaptations, and multimodal priors such as images or text [19] could further broaden applicability and enhance perceptual realism.

IX. Conclusion

This paper presented the Fourier-enhanced Transformer Encoder Network (FoTEN), a lightweight framework that combines Fourier decomposition with Transformer encoders for real-time vibrotactile texture synthesis. The use of fixed-size sliding windows simplified modeling by removing the need for complex segmentation, while the integration of Huber loss improved reconstruction accuracy and training stability. Results showed consistent improvements in both time- and frequency-domain fidelity, as well as perceptual similarity validated through user studies, marking clear advances over existing methods. Remaining challenges are tied to actuator limitations in rendering smooth or compliant textures. Future work will investigate advanced actuation, variable tool conditions, and multimodal priors to further broaden applicability.

REFERENCES

[1] G. S. Giri, Y. Maddahi, and K. Zareinia, "An application-based review of haptics technology," *Robotics*, vol. 10, no. 1, p. 29, 2021.

[2] P. Strohmeier and K. Hornbæk, "Generating haptic textures with a vibrotactile actuator," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 4994–5005.

[3] T. H. Massie, "Initial haptic explorations with the phantom: Virtual touch through point interaction," Ph.D. dissertation, Massachusetts Institute of Technology, 1996.

[4] Fritz and K. E. Barner, "Stochastic models for haptic texture," in *Telemanipulator and Telepresence Technologies III*. SPIE, 1996.

[5] K. Tozuka, B. Poitrimol, G. Sasaki, K. Kobayashi, and H. Igarashi, "Integrating texture models through regression of vibration and texture characteristics," *ROBOMECH Journal*, vol. 12, no. 1, p. 25, 2025.

[6] L. Tao, F. Wang, Y. Li, J. Wu, X. Jiang, and Q. Xi, "A cross-texture haptic model based on tactile feature fusion," *Multimedia Systems*, vol. 30, no. 3, pp. 1–12, 2024.

[7] S. D. Cranstoun, H. C. Ombao, R. Von Sachs, Guo, and B. Litt, "Time-frequency spectral estimation of multichannel eeg using the auto-slex method," *IEEE transactions on Biomedical Engineering*, vol. 49, 2002.

[8] R. A. Davis, T. C. M. Lee, and G. A. Rodriguez-Yam, "Structural break estimation for nonstationary time series models," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 223–239, 2006.

[9] A. Abdulali, I. R. Atadjanov, and S. Jeon, "Visually guided acquisition of contact dynamics and case study in data-driven haptic texture modeling," *IEEE Transactions on Haptics*, vol. 13, no. 3, pp. 611–627, 2020.

[10] J. M. Romano and K. J. Kuchenbecker, "Creating realistic virtual textures from contact acceleration data," *IEEE Transactions on haptics*, vol. 5, no. 2, pp. 109–119, 2011.

[11] M. I. Awan, T. Ogay, W. Hassan, D. Ko, S. Kang, and S. Jeon, "Model-mediated teleoperation for remote haptic texture sharing: Initial study of online texture modeling and rendering," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023.

[12] H. Culbertson, J. Unwin, and K. J. Kuchenbecker, "Modeling and rendering realistic textures from unconstrained tool-surface interactions," *IEEE transactions on haptics*, vol. 7, no. 3, pp. 381–393, 2014.

[13] S. Shin, R. H. Osgouei, K.-D. Kim, and S. Choi, "Data-driven modeling of isotropic haptic textures using frequency-decomposed neural networks," in *2015 IEEE World Haptics Conference (WHC)*, 2015.

[14] N. Heravi, W. Yuan, A. M. Okamura, and J. Bohg, "Learning an action-conditional model for haptic texture generation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 088–11 095.

[15] J. B. Joolee and S. Jeon, "Data-driven haptic texture modeling and rendering based on deep spatio-temporal networks," *IEEE Transactions on Haptics*, vol. 15, no. 1, pp. 62–67, 2021.

[16] D. Chen, D. Zhu, J. Liu, G. Chen, Y. Fang, and Y. Zhang, "Research on texture haptic reconstruction method based on informer model," in *Proceedings of the 2023 3rd International Conference on Robotics and Control Engineering*, 2023, pp. 161–165.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[19] M. Naeem, M. I. Awan, and S. Jeon, "Text-driven generative framework for multimodal visual and haptic texture synthesis," in *2025 IEEE World Haptics Conference (WHC)*. IEEE, 2025, pp. 140–146.

[20] D. Chen, Y. Ding, G. Chen, T. Fan, J. Liu, and A. Song, "Low-delay haptic texture display method based on user action information and texture image," *International Journal of Human-Computer Studies*, vol. 199, p. 103500, 2025.

[21] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[22] C.-c. Jin and X. Chen, "An end-to-end framework combining time–frequency expert knowledge and modified transformer networks for vibration signal classification," *Expert Systems with Applications*, 2021.

[23] X. Shi, L. Wang, X. Liu, J. Wu, and Z. Shao, "Scene-aware foveated neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[24] E. G. S. Nascimento, T. A. de Melo, and D. M. Moreira, "A transformer-based deep neural network with wavelet transform for forecasting wind speed and wind energy," *Energy*, vol. 278, p. 127678, 2023.

[25] Y. Zou, P. Zou, Y. Zhao, K. Zhang, R. Zhang, and X. Wang, "Melons: generating melody with long-term structure using transformers and structure graph," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

[26] M. I. Awan and J. Seokhee, "Surface texture classification based on transformer network," *Korean HCI Society Conference*, pp. 762–764, 2023.

[27] S. Okamoto, H. Nagano, and Y. Yamada, "Psychophysical dimensions of tactile perception of textures," *IEEE transactions on haptics*, vol. 6, no. 1, pp. 81–93, 2012.

[28] M. I. Awan and S. Jeon, "Estimating perceptual attributes of haptic textures using visuo-tactile data," *IEEE Access*, 2025.

[29] W. McMahan and K. J. Kuchenbecker, "Dynamic modeling and control of voice-coil actuators for high-fidelity display of haptic vibrations," in *2014 IEEE Haptics Symposium (HAPTICS)*. IEEE, 2014, pp. 115–122.

[30] K. Tozuka and H. Igarashi, "A simplified texture modeling using a physical and perceptual rule-based approach," *IEEE Access*, 2024.

[31] A. Abdulali and S. Jeon, "Data-driven modeling of anisotropic haptic textures: Data segmentation and interpolation," in *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*. Springer, 2016, pp. 228–239.

[32] D. Nie, J. Liu, and X. Sun, "Influence of surface tactile data quantity on material classification in unstructured environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[33] A. Jadon, A. Patil, and S. Jadon, "A comprehensive survey of regression based loss functions for time series forecasting," *arXiv preprint arXiv:2211.02989*, 2022.

[34] Y. Ujitoko and Y. Ban, "Vibrotactile signal generation from texture images or attributes using generative adversarial network," in *Haptics: Science, Technology, and Applications: 11th International Conference, EuroHaptics 2018, Pisa, Italy, June 13-16, 2018, Proceedings, Part II 11*. Springer, 2018, pp. 25–36.

Mudassir Ibrahim Awan received the B.E. degree in electronics engineering from the Karachi Institute of Economics and Technology (KIET), Karachi, Pakistan, in 2016, and the integrated M.S. and Ph.D. degrees in computer science and engineering from Kyung Hee University, Yongin, South Korea, in 2025. He is currently a Postdoctoral Researcher with Kyung Hee University. His research interests include haptics, virtual and augmented reality, human-centered artificial intelligence, immersive media, and intelligent interactive systems. His work also focuses on perception-driven computing, multimodal interaction, and adaptive frameworks for realistic sensory experiences in digital environments.

Sungjoo Kang received the B.S. and M.S. degrees in electrical and computer engineering from Hanyang University, Seoul, Korea, in 2003 and 2005, respectively. He received an M.S.E. in software engineering from Carnegie Mellon University, USA, in 2010, and an M.S. in software engineering from KAIST, Korea, in 2011. He received the Ph.D. degree in computer engineering from Chungnam National University, Daejeon, Korea, in 2018. Since 2005, he has been with the Electronics and Telecommunications Research Institute (ETRI), where he is currently a Principal Researcher and Director of the On-device System Software Research Section. He also served as a Visiting Scientist at the University of Texas at Dallas, USA, in 2022. His research interests include on-device and edge AI, cyber-physical systems (CPS), digital twins, VR, and haptics.

Dongbeom Ko received the B.S. and Ph.D. degrees in computer engineering from Tech University of Korea, South Korea, in 2016 and 2021, respectively. Since 2021, he has been with the Electronics and Telecommunications Research Institute (ETRI), South Korea, where he is currently a Senior Researcher. His research has focused on the design and development of intelligent on-device systems for efficient AI processing in resource-constrained environments. His research interests include on-device systems, on-device AI, digital twins, edge intelligence, and embedded AI systems.

Waseem Hassan received the B.S. degree in electrical and telecommunication engineering from the National University of Sciences and Technology (NUST), Pakistan, in 2012, and the M.S. and Ph.D. degrees in computer engineering from Kyung Hee University, Republic of Korea, in 2016 and 2022, respectively. From 2022 to 2023, he was a Postdoctoral Researcher with Kyung Hee University, Republic of Korea. From 2023 to 2025, he was a Postdoctoral Researcher with the University of Copenhagen, Denmark. Since 2025, he has been a Marie Curie Fellow with UpnaLab, Public University of Navarre, Spain. His research interests include psychophysics, haptic content generation, volumetric haptic displays, multimodal interaction, and novel haptic interfaces.

Seong Tae Kim (Member, IEEE) received the Ph.D. degree from KAIST, South Korea, in 2019.

In 2015, he was a Visiting Researcher with the University of Toronto, Canada. From 2019 to 2021, he was a Senior Research Scientist with the Chair for Computer Aided Medical Procedures, Technical University of Munich, Germany. He is currently an Associate Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea. He has authored or co-authored more than 90 peer-reviewed journal and conference papers. His current research interests include deep learning, spatio-temporal learning, explainable artificial intelligence, and medical image analysis. He received the Best Student Paper Award at SPIE Medical Imaging in 2018.

Seokhee Jeon He received his B.S. and Ph.D. degrees in Computer Science and Engineering from Pohang University of Science and Technology (POSTECH) in 2003 and 2010, respectively. He then worked as a Postdoctoral Research Associate at the Computer Vision Laboratory, ETH Zurich. In 2012, he joined the Department of Computer Engineering at Kyung Hee University as an Assistant Professor and became a Full Professor in 2024. He is also a co-founder and faculty member in the Department of Immersive AX Convergence at Kyung Hee University. His research interests include data-driven haptic modeling and rendering, hyper-realistic multimodal feedback in virtual, augmented, and remote environments, and the development of modular wearable haptic interfaces with enhanced applicability.