Trading Sensing for Structure: Sparse IMU-EMG Fingertip Force Estimation via Neuro-inspired Structured Modeling

Yang Gao
South China University of Technology
Guangzhou, China
gaoyang2025@scut.edu.cn

Yingjing Xiao
East China Normal University
Shanghai, China
51275901140@stu.ecnu.edu.cn

Junbin Ren
East China Normal University
Shanghai, China
renjunbin001@gmail.com

Chenxu Zhang
South China University of Technology
Guangzhou, China
ftcxzhang@mail.scut.edu.cn

Wenbo Zhang
South China University of Technology
Guangzhou, China
202310193158@mail.scut.edu.cn

Zhanpeng Jin*
South China University of Technology
Guangzhou, China
zjin@scut.edu.cn

Abstract

We present a Neuro-inspired Structured Model (NiSM) for fine-grained finger-level contact force estimation during dexterous grasping from sparse wearable sensing. NiSM uses a thumb-mounted inertial measurement unit (IMU) and a single-channel wrist electromyography (EMG) sensor, while deriving grasp context from an IMU-based hand-shape latent. This work studies the trade-off between sensing density and structural inductive bias: when sensor coverage is limited, inferred grasp context, kinematic coupling, and muscle-effort modulation can provide useful constraints for force prediction. Using human sensorimotor control as an architectural analogy, NiSM organizes computation into complementary pathways: a context-conditioned feedforward pathway maps thumb kinematics to finger-specific estimates through sparse learned gating, while an EMG-conditioned modulation pathway provides effort-dependent scaling. By explicitly encoding these structural priors, NiSM reduces reliance on dense sensor arrays and large generic black-box models. Evaluations on multi-user, multi-object datasets show that NiSM achieves performance competitive with a dense-sensing reference where comparable dense inputs are available, while using substantially fewer parameters, and is more robust than generic models under limited-data settings. These results indicate that structural priors can partially recover force-relevant information under sparse sensing, offering a practical direction for wearable finger force estimation under real-world deployment constraints.

1 Introduction

Estimating finger-level contact forces is essential for physically grounded hand interaction, enabling tasks such as dexterous manipulation, rehabilitation, and teleoperation. Despite substantial progress in hand pose estimation, force estimation remains less developed, largely because contact force depends on inter-finger coordination, muscle activation, and object interaction dynamics that are difficult to observe with wearable sensors of limited coverage.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing methods often rely on dense sensing, such as multi-channel EMG [34, 36, 28] or data gloves [2, 15, 26, 3], which can be impractical for everyday deployment due to bulk, discomfort, or cost. Recent lightweight alternatives, including camera-based systems [38, 21, 29, 14], reduce the need for dense wearable instrumentation, but often remain sensitive to visibility conditions or rely on generic black-box regression. These approaches provide useful sensing channels, yet they do not explicitly model the structured motor constraints that shape finger coordination and force generation.

This setting raises an inductive bias–information trade-off: when sensing density is reduced, what structure can be encoded in the model to recover useful force-relevant information? Biological sensorimotor systems provide a natural source of such structure, using task context, coordinated motor patterns, and muscle-dependent modulation to generate dexterous forces from limited and distributed signals [5]. Motivated by these principles, we study whether neuro-inspired structural inductive biases can improve sparse-input force estimation by constraining how kinematic, intent, and muscle-activation information are integrated.

In this work, we present a neuro-inspired structured model for finger force estimation using a thumb-mounted IMU, a single-channel wrist EMG sensor, and a grasp-context inferred from IMU-derived hand-shape latent. This configuration is highly constrained yet practical for wearable deployment. Our model organizes computation into complementary pathways inspired by functional components of human sensorimotor control. A context-modulated feedforward pathway maps thumb kinematics to finger-specific force estimates through sparse learned gating, while an EMG-conditioned modulation pathway provides effort-dependent scaling. This design explicitly encodes task context, kinematic coupling, and muscle-effort modulation as structural priors, in contrast to generic black-box fusion models.

We evaluate the proposed model on multi-user, multi-object force estimation datasets and compare it with sparse-input sequence models, black-box fusion models, and dense-sensing baselines. The results show that NiSM achieves performance competitive with dense-sensing baselines while using substantially fewer parameters, and exhibits stronger robustness than generic models under limited-data settings. Through ablation analysis, we further quantify the contribution of each structural component, highlighting how model structure can recover part of the information typically supplied by denser sensing.

Our main contributions are:

- We propose NiSM, a neuro-inspired structured model that studies how three inductive biases can recover part of the force-relevant information lost to missing sensing channels in sparse IMU-EMG force estimation: inferred grasp context, sparse kinematic coupling, and EMG-conditioned effort scaling.
- We build and evaluate a diverse multi-user, multi-object dataset covering realistic grasp types and force levels, enabling systematic evaluation of sparse-sensing force estimation models.
- We show that NiSM achieves performance competitive with dense-sensing references using substantially fewer parameters, and ablations reveal how inferred grasp context, sparse kinematic coupling, and EMG-driven effort scaling contribute force-relevant information.

2 Related Work

2.1 Hand Interaction Sensing via Vision

Visual cues, such as fingernail color changes, deformation patterns, and skin displacement, have been widely explored as non-contact proxies for estimating fingertip pressure and interaction forces. Mascaro and Asada [31] reconstructed three-axis fingertip forces and finger postures from subnail blood volume variations. Chen et al. [7] further estimated force, torque, and surface curvature using visual features from the fingernail and surrounding skin. Fallahinia et al. [10] employed deep neural networks for real-time tactile force estimation from video inputs. These methods reduce the need for wearable force sensors, but their accuracy depends heavily on high-resolution visual observations and remains sensitive to lighting, occlusion, and hand-object visibility.

Other vision-based approaches estimate pressure or interaction forces directly from RGB or egocentric observations. PressureVision++ [16] predicts fingertip pressure from a single RGB image, while

EgoPressure [45] introduces synchronized RGB-D data for hand pose and pressure estimation. Object-centric methods infer forces from observed object motion or physical simulation, including works by Ehsani et al. [9], Li et al. [24], and Pham et al. [33]. Although these approaches provide valuable external sensing channels, they can struggle with static objects, limited viewpoints, or cluttered interactions. More importantly, many rely on end-to-end visual regression rather than explicitly modeling the internal sensorimotor structure that links intent, kinematics, muscle activation, and force generation.

2.2 Hand Interaction Sensing via Wearable Sensors

2.2.1 Single-Modality Approaches

Wearable surface electromyography (sEMG) and inertial measurement units (IMUs) are commonly used for hand interaction sensing because they are compact and compatible with mobile deployment. sEMG captures neuromuscular activation and has been used for coarse gesture classification [27, 22, 11, 35] and global grip force estimation [13, 23, 30]. However, sEMG alone lacks explicit kinematic information and has limited observability for spatially resolved fingertip forces. IMUs, in contrast, are effective for tracking hand and finger motion [42, 25, 46, 18], but they primarily measure kinematic states and do not directly capture muscle effort or contact dynamics [1]. Thus, each modality provides only partial information: sEMG reflects internal effort, while IMUs encode motion. This observability gap motivates multimodal sensing and structured integration.

2.2.2 Multimodal Fusion Approaches

Several studies combine IMU and sEMG signals for gesture recognition or force estimation. Multimodal hand gesture classification has been demonstrated with forearm-mounted sensors and deep Q-networks [40], as well as with multi-channel wrist-worn systems [20]. For force-related tasks, prior work has used high-density sEMG and IMU to estimate wrist angle and grasp force [23], or to predict endpoint force during muscle contractions [19]. These methods show the value of complementary sensing, but often operate at the level of discrete gestures, global force, or endpoint force rather than fine-grained per-finger contact estimation.

Recent dense multimodal systems provide richer signals for detailed hand interaction modeling. PiMForce [37], for example, integrates dense sensing and posture information to estimate spatially distributed hand pressure under free-form interactions. Such systems demonstrate the benefit of high sensing density, but their instrumentation can be cumbersome for everyday deployment. In addition, many fusion models rely on generic deep architectures, leaving open the question of how much sensing density can be reduced when task structure, kinematic coupling, and effort-dependent modulation are explicitly encoded. Our work addresses this question by studying sparse wearable inputs together with structural inductive biases.

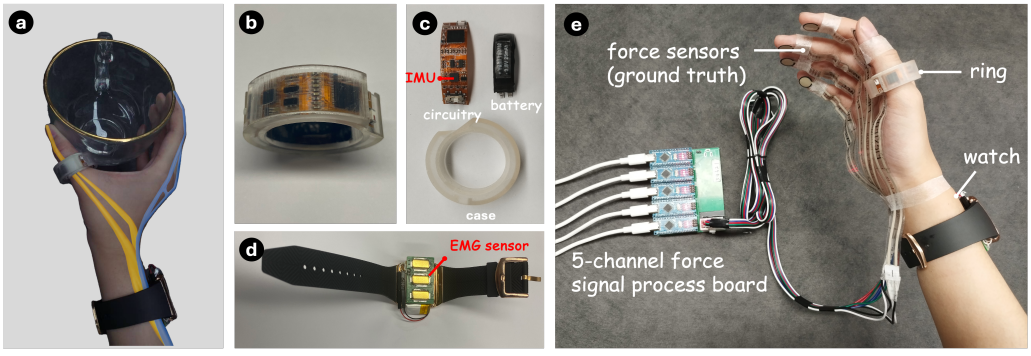


Figure 1: Minimalist Ring-Watch system for finger force sensing.(a) Hand interaction during grasping. (b–c) Custom ring with embedded IMU. (d) Watch with integrated EMG sensor. (e) Full setup with ground-truth force sensors and signal board.

2.3 Neuro-inspired Models for Motor Control

Biological motor control has motivated many models for robot locomotion, arm movement, manipulation, and navigation. Examples include central pattern generator models for locomotion [6], neural controllers for arm movement [4], and bio-inspired sensorimotor control frameworks using artificial or spiking neural networks [43, 39, 12]. These studies show that structured computation can improve efficiency, adaptability, and robustness in embodied control.

However, most bio-inspired models target control generation rather than fine-grained per-finger force estimation from sparse wearable biosignals. They also often rely on dense sensory inputs, such as vision or detailed sensor arrays [43, 17], which differs from the practical constraints of minimal wearable sensing. Although structured neural architectures have been studied as inductive priors for embodied control [5, 39], their use for recovering force-relevant information from sparse internal signals remains underexplored. In particular, task-dependent modulation, sparse synergy-like mappings, and EMG-conditioned effort scaling have not been systematically integrated for sparse finger force prediction.

To address this gap, NiSM uses the functional organization of sensorimotor control as an architectural template, combining context-conditioned feedforward prediction with EMG-conditioned scaling, and studies how this structural inductive bias can compensate for reduced sensing density in wearable force estimation.

3 Methodology

3.1 Minimalist Ring-Watch Sensing System

To support force-aware hand interaction sensing in an everyday wearable form factor, we design a minimalist ring-watch sensing system consisting of a finger-worn ring with a 9-axis Inertial Measurement Unit (IMU) and a smartwatch-like device with a single-channel surface Electromyography (sEMG) sensor (Figure 1). The thumb-mounted IMU captures local motion and orientation, while the wrist EMG records coarse muscle activation related to grasp effort. The ring is worn on the thumb because the thumb participates in nearly all grasp types and serves as the opposition reference for the other fingers. This sparse configuration balances sensing capability with comfort, portability, and social acceptability, mitigating the deployment limitations of bulkier data gloves or dense IMU/EMG arrays. The time-synchronized IMU and EMG streams provide complementary but partial observations—local thumb kinematics and global muscle-effort cues—which form the sparse sensing basis for our neuro-inspired structured model; task-level grasp intent is incorporated at the model level, as described in Section 3.2, to estimate per-finger contact pressure.

3.2 Neuro-inspired Structured Model

To study how structural inductive bias can compensate for reduced sensing density in wearable force estimation, we design NiSM, a neuro-inspired structured architecture for our minimalist *ring-watch system* (Section 3.1), as shown in Figure 2. Rather than relying on dense sensing or generic black-box fusion, NiSM encodes task context, kinematic coupling, and effort-dependent modulation as explicit structural priors.

The model predicts per-finger contact pressure from two sparse sensing channels — thumb kinematics from an IMU and forearm muscle activation from a single-channel EMG, with a grasp-context feature inferred from the IMU stream. These inputs are processed through interpretable modules inspired by human sensorimotor control, including context-conditioned kinematic processing, sparse finger-wise projection, and EMG-conditioned gain modulation.

We formulate sparse-input force estimation as:

$$\mathbf{P}(t) = f\left(\mathbf{K}(t), \mathbf{E}(t)\right), \text{with } \hat{c}(t) = g_{ctx}(\mathbf{K}(t)) \tag{1}$$

where $\mathbf{K}(t)$ represents the thumb IMU-derived kinematic state, $\mathbf{E}(t)$ denotes the synchronized EMG signal reflecting muscle-effort information, and the grasp-context feature $\hat{c}$ is inferred from $\mathbf{K}(t)$ via a hand-shape latent encoder g_{ctx}. In Figure 2 and below, $\hat{c}(t)$ is denoted $\hat{\mathbf{z}}_{\text{intent}}$.

Given synchronized thumb IMU and EMG streams, NiSM outputs a five-dimensional vector $\mathbf{p}_{\text{final}} \in \mathbb{R}^5$ representing finger-level contact pressure estimates.

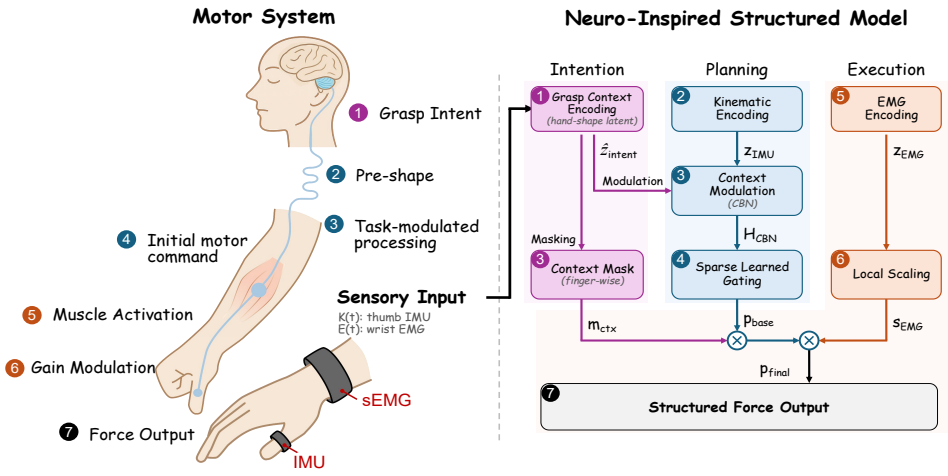


Figure 2: Overview of the Neuro-inspired Structured Model (NiSM) for Finger Force Prediction. Left: Biological Inspiration: A simplified schematic of the human motor system illustrating the hypothesized information flow from grasp intent (1) through sensorimotor processing (2-4) to muscle activation and gain modulation (5–6) and force output (7). Right: NiSM follows this organization with three stages: intention, planning, and execution. The thumb IMU stream $\mathbf{K}(t)$ is encoded by modules 1 and 2, and the wrist EMG $\mathbf{E}(t)$ by module 5. Module 1 infers the hand-shape latent $\hat{\mathbf{h}}$ and maps it to the context feature $\hat{\mathbf{z}}_{\text{intent}}$ ($\hat{c}(t)$ in Eq. 1), which conditions the CBN modulation and the finger-wise context mask $\mathbf{m}_{\text{ctx}}$.

3.2.1 Input Encoding

This stage encodes sparse sensor measurements into compact feature representations for subsequent structured force prediction.

Thumb Kinematic Encoding The thumb IMU sequence $\mathbf{K} \in \mathbb{R}^{T \times d_{\text{IMU}}}$, where $T = 30$ and $d_{\text{IMU}} = 12$ (3-axis acceleration + 9-element rotation matrix), is processed by an LSTM followed by a linear layer. This captures temporal dependencies in thumb motion and yields a fixed-size kinematic feature vector $\mathbf{z}_{\text{IMU}} \in \mathbb{R}^{d_z}$, providing motion context for inferring coupled finger-force patterns from sparse thumb observations.

Inferred Grasp-Context Encoding. NiSM does not directly embed ground-truth grasp-intent or object labels. Instead, it derives a continuous grasp-context feature from an IMU-based hand-shape latent. The thumb IMU sequence is encoded into a hand-shape representation $\hat{\mathbf{h}}$, which is then mapped by an MLP context encoder into $\hat{\mathbf{z}}_{\text{intent}}$. Object/action labels, when available, are used only as auxiliary training supervision and are not provided to the model at inference. Both $\hat{\mathbf{z}}_{\text{intent}}$ and $\mathbf{z}_{\text{IMU}}$ are derived from the thumb IMU, so $\hat{\mathbf{z}}_{\text{intent}}$ serves as an auxiliary-supervised continuous task condition computed from the same observation.

EMG Activation Encoding Each single-channel wrist EMG window $\mathbf{E}_{\text{window}} \in \mathbb{R}^{T \times 1}$ is first converted into a compact 10-dimensional feature vector $\mathbf{f}_{\text{EMG}} \in \mathbb{R}^{10}$ via time- and frequency-domain feature extraction. This vector is then passed through an MLP-based encoder E_{EMG}, producing the muscle activation feature $\mathbf{z}_{\text{EMG}} \in \mathbb{R}^{d_z}$ after a final `tanh` activation, which serves as an effort-related feature for EMG-conditioned force modulation.

3.2.2 Planning Pathway: Context-Modulated Command Synthesis

This pathway implements an hand-shape-conditioned feedforward prediction from sparse thumb kinematics. Its design uses the inferred grasp context to condition how kinematic information is interpreted, and then applies sparse learned gating to produce finger-specific base pressure estimates.

Context-Conditioned Modulation: The kinematic feature $\mathbf{z}_{\text{IMU}} \in \mathbb{R}^{d_z}$ is modulated by the context feature $\hat{\mathbf{z}}_{\text{intent}} \in \mathbb{R}^{d_z}$ using Conditional BatchNorm (CBN) [8]. The CBN layer dynamically adjusts

its normalization parameters based on $\hat{\mathbf{z}}_{\text{intent}}$:

$$\mathbf{H}_{\text{CBN}} = \gamma(\hat{\mathbf{z}}_{\text{intent}}) \odot \frac{\mathbf{z}_{\text{IMU}} - \mu_{\text{B}}}{\sigma_{\text{B}}} + \beta(\hat{\mathbf{z}}_{\text{intent}}) \tag{2}$$

where μ_B and σ_B are batch statistics of $\mathbf{z}_{\text{IMU}}$ over the current mini-batch, and $\odot$ denotes element-wise multiplication. The scaling parameters $\gamma(\hat{\mathbf{z}}_{\text{intent}}) \in \mathbb{R}^{d_z}$ and shifting parameters $\beta(\hat{\mathbf{z}}_{\text{intent}}) \in \mathbb{R}^{d_z}$ are affine transformations of $\hat{\mathbf{z}}_{\text{intent}}$, learned by separate linear layers:

$$\gamma(\hat{\mathbf{z}}_{\text{intent}}) = \mathbf{W}_\gamma \hat{\mathbf{z}}_{\text{intent}} + \mathbf{b}_\gamma \tag{3}$$

$$\beta(\hat{\mathbf{z}}_{\text{intent}}) = \mathbf{W}_\beta \hat{\mathbf{z}}_{\text{intent}} + \mathbf{b}_\beta \tag{4}$$

where $\mathbf{W}_\gamma, \mathbf{W}_\beta \in \mathbb{R}^{d_z \times d_z}$ and $\mathbf{b}_\gamma, \mathbf{b}_\beta \in \mathbb{R}^{d_z}$ are learnable parameters. This modulation encodes the inferred grasp context as a task condition, allowing the same sparse thumb motion to be interpreted differently under different grasp conditions and producing a task-contextualized feature $\mathbf{H}_{\text{CBN}} \in \mathbb{R}^{d_z}$.

Sparse Learned Gating: The context-modulated kinematic feature $\mathbf{H}_{\text{CBN}}$ is projected to the base pressure prediction $\mathbf{p}_{\text{base}} \in \mathbb{R}^{N_f}$ through a learned sparse weight matrix $\mathbf{W}_{\text{sparse}} \in \mathbb{R}^{d_z \times N_f}$. Each column $\mathbf{W}_{\text{sparse},j} \in \mathbb{R}^{d_z}$ corresponds to the feature-to-pressure mapping for the j-th finger:

$$p_{\text{base},j} = \mathbf{H}_{\text{CBN}}^\top \mathbf{W}_{\text{sparse},j}. \tag{5}$$

The sparse projection acts as a structural constraint on the mapping from shared kinematic features to finger-specific pressure estimates. Its learned non-zero weights identify salient feature–finger dependencies, reflecting synergy-like coordination patterns while avoiding a fully dense black-box mapping. Sparsity in $\mathbf{W}_{\text{sparse}}$ is encouraged through L_1 regularization during training (Section 3.3).

3.2.3 Execution Pathway: EMG-Conditioned Effort Modulation

Complementary to the feedforward pathway, this module uses the EMG signal to provide effort-dependent, finger-wise modulation of the base pressure prediction. This pathway introduces muscle-activation information that is not directly observable from sparse thumb kinematics, serving as a structural prior for compensating reduced sensing density.

EMG-Conditioned Local Scaling: In parallel with the feedforward pathway, the encoded EMG feature $\mathbf{z}_{\text{EMG}}$ is mapped to finger-wise gain factors. A linear layer followed by a sigmoid activation produces $\mathbf{s}_{\text{EMG}} \in [0, 1]^5$, with one scaling factor for each finger. Rather than directly predicting pressure from EMG alone, this module uses muscle effort to modulate the base prediction generated from context-conditioned thumb kinematics. In this way, EMG provides an effort-related constraint that reweights each finger's base pressure estimate, allowing NiSM to incorporate peripheral muscle-activation cues while avoiding a fully dense black-box fusion.

3.2.4 Structured Force Output

The final finger pressure prediction $\mathbf{p}_{\text{final}}$ is obtained by multiplicatively combining the feedforward base prediction $\mathbf{p}_{\text{base}}$, the inferred context-dependent finger mask $\mathbf{m}_{\text{ctx}}$, and the EMG-conditioned scaling factors $\mathbf{s}_{\text{EMG}}$:

$$\mathbf{p}_{\text{final}} = \text{ReLU}(\mathbf{p}_{\text{base}} \odot \mathbf{m}_{\text{ctx}} \odot \mathbf{s}_{\text{EMG}}) \tag{6}$$

where $\odot$ denotes element-wise multiplication. $\mathbf{m}_{\text{ctx}} = M_{\text{ctx}}(\hat{\mathbf{z}}_{\text{intent}})$ provides a finger-wise task prior inferred from IMU-derived hand-shape context, while $\mathbf{s}_{\text{EMG}}$ introduces effort-dependent modulation from muscle activation. This structured interaction combines task, motion, and effort cues to constrain pressure prediction under sparse sensing, and the ReLU activation enforces non-negative outputs.

3.3 Training Objective with Structural Regularization

The model parameters are optimized with a loss function that balances pressure prediction accuracy and structural regularization, encouraging NiSM to learn an information-efficient mapping under sparse sensing.

The primary objective is to minimize the Mean Squared Error (MSE) between the predicted pressures $\mathbf{p}_{\text{final}}$ and the ground-truth pressures $\mathbf{p}_{\text{GT}}$:

$$L_{\text{MSE}} = ||\mathbf{p}_{\text{final}} - \mathbf{p}_{\text{GT}}||_2^2 \tag{7}$$

To promote sparsity in the learned gating matrix $\mathbf{W}_{\text{sparse}}$ within the feedforward pathway, we add an ℓ_1 regularization term:

$$L_{\text{sparse}} = ||\mathbf{W}_{\text{sparse}}||_1 \tag{8}$$

This encourages many weights in $\mathbf{W}_{\text{sparse}}$ to approach zero, forcing the model to rely on the most relevant context-modulated thumb kinematic features for each finger's base pressure prediction. This ℓ_1 regularization acts as a structural prior, guiding the model toward a sparse and interpretable feature-to-finger mapping rather than an unconstrained dense projection. Additionally, to improve the reconstruction of critical peak pressure magnitudes, we incorporate a peak-pressure error term L_{peak}. This term penalizes discrepancies between predicted and ground-truth peak pressures, defined as the sum of squared differences between the maximum values of $\mathbf{p}_{\text{final},i}$ and $\mathbf{p}_{\text{GT},i}$ for each finger i within the training window.

The total loss function is a weighted sum:

$$L_{\text{total}} = L_{\text{MSE}} + \lambda_{\text{sparse}} L_{\text{sparse}} + \lambda_{\text{peak}} L_{\text{peak}} + \lambda_{\text{shape}} \mathcal{L}_{\text{shape}} + \lambda_{\text{ctx}} \mathcal{L}_{\text{ctx}} \tag{9}$$

where λ_{sparse}, λ_{peak}, λ_{shape}, and λ_{ctx} control sparsity regularization, peak-pressure reconstruction, auxiliary hand-shape supervision, and auxiliary context regularization, respectively. The last two terms are used only during training and do not require hand-shape targets or object/action labels at inference. Minimizing L_{total} promotes accurate pressure estimation while maintaining sparse structural bias under reduced sensing density.

4 Experiments

4.1 Dataset

Self-Collected Dataset We constructed a dataset for finger-level contact pressure estimation under sparse wearable sensing conditions with IRB approval. In this work, fingertip contact pressure is used as a proxy for finger-level normal contact force. The dataset contains recordings from 20 participants performing 20 diverse hand gestures and grasp actions (e.g., power grasp, pinch, tool use), each repeated five times. Participants were encouraged to freely vary their force during repetitions, resulting in high intra-action variability and more natural grasping behavior. They also put on the ring and watch themselves without a mounting fixture, so device placements reflect natural everyday wear. The dataset comprises over 1 million time-aligned frames. Ground-truth fingertip normal-force readings were recorded at 100 Hz using FlexiForce sensors on all five fingers, with measurements up to 25 N, and were normalized as contact-pressure labels. These labels were synchronized with a thumb-mounted IMU and a single-channel wrist EMG using shared timestamps. Additionally, 3D hand poses were extracted for auxiliary analysis.

This dataset provides a sparse-sensing testbed for evaluating how thumb kinematics and muscle-effort cues complement each other under realistic grasp variability. We split it by participant into 16 training and 4 test users. A quantitative comparison with public benchmark data is provided in the Appendix.

Public Benchmark Dataset To evaluate NiSM under a densely instrumented benchmark setting, we use the PiMForce dataset [37], a large-scale multimodal corpus for whole-hand pressure estimation. It includes 21 participants performing predefined grasp and manipulation tasks, recorded with 3D hand posture, 8-channel sEMG, and high-resolution pressure measurements covering fingertips and palm patches, totaling over 80 million multimodal frames.

Unlike our sparse ring-watch setup, PiMForce provides dense sensing and structured interaction protocols. We hold out 4 of its 21 participants for testing (Appendix A.1.2). For fair comparison, we project both predicted and ground-truth outputs to a unified five-finger pressure space. This enables a comparison in a common output space between sparse structured modeling and dense-sensing baselines, allowing us to examine how much force-relevant information can be recovered under reduced sensing density.

4.2 Baselines

To evaluate NiSM under the sensing density–structural inductive bias trade-off, we compare it with baselines that differ in model structure and input density. These include generic time-series models trained on the same sparse inputs, a sparse-input black-box fusion model, and a dense-sensing

reference model. Together, they assess NiSM's prediction performance, data efficiency, and the contribution of its structural priors under reduced sensing density.

Time-Series Models. To compare NiSM with generic sequence modeling approaches, we adapt representative time-series models to the same sparse IMU and EMG inputs. These include: **Informer [47]**, a Transformer-based encoder-decoder model for long-range temporal dependency modeling, configured for concurrent finger pressure prediction; **LSTM-Transformer [32]**, a hybrid recurrent-attention model adapted for per-frame pressure regression from sparse inputs; and **Crossformer [44]**, a Transformer-based model that explicitly captures cross-dimension dependencies in multivariate time series, adapted to our multimodal pressure estimation task. These baselines test whether generic temporal modeling alone can extract force-relevant information from limited wearable signals.

Sparse Input Black-Box Deep Learning Model. This baseline evaluates whether a more complex generic fusion architecture can achieve comparable performance without NiSM's structural inductive biases. We include a generic encoder-decoder with Cross-Attention Transformer (CA-T) [41], which jointly predicts finger pressure and 3D hand posture from sparse IMU/EMG inputs. Although CA-T can exploit correlations between pose and pressure, it does not explicitly encode task-dependent modulation, sparse feature-to-finger mapping, or EMG-conditioned effort modulation, serving as a black-box sparse-input baseline.

Dense Input Reference Model. This baseline provides a data-rich reference using denser sensing than our ring-watch setup. Comparing NiSM with this model allows us to examine whether structural inductive biases can recover part of the force-relevant information typically supplied by denser sensing. We implement and retrain the core encoder-decoder architecture from PiMForce [37], a state-of-the-art model for hand pressure estimation. This baseline is trained with dense sensing modalities available in PiMForce, such as multi-channel EMG and 3D hand posture, while pressure measurements are used as supervision. For fair comparison, its predictions are mapped to the same unified five-finger pressure space.

5 Results: How Much Force-Relevant Information Can Structure Recover?

We organize the results around a central question: how much force-relevant information can neuro-inspired structure recover under reduced sensing density? We examine this through three analyses: comparison with sparse black-box models and a dense-sensing reference, evaluation under limited user data, and component ablations that reveal which priors recover missing force-relevant information.

5.1 Can Structured Sparse Sensing Narrow the Gap to Dense Sensing?

We benchmark NiSM against representative models on both our self-collected dataset and the public PiMForce dataset. As shown in Table 1, NiSM achieves the best overall performance among the evaluated baselines while using significantly fewer parameters (1.71M). On the PiMForce benchmark, where a dense-sensing reference is available, NiSM achieves R^2 = 54.40% and PCC = 76.60%, slightly outperforming the dense PiMForce model (R^2 = 52.68%, PCC = 73.73%), which relies on denser inputs and substantially more parameters (66.39M). The standard deviations reported in Table 1 are computed over the four held-out test users, each evaluated separately, and thus reflect inherent inter-user variability. Because this test set is small, we also run 21-fold leave-one-subject-out cross-validation on PiMForce (in each fold, one participant is held out for testing and another for checkpoint selection), which gives $R^2 = 55.04 \pm 10.24\%$ and PCC = $76.23 \pm 5.70\%$, close to the fixed-split result. Paired comparisons over the same 21 folds show that NiSM improves R^2 by 0.300 (95% CI [0.226, 0.374]) over CA-T and by 0.167 ([0.128, 0.206]) over PiMForce (dense), with two-sided Wilcoxon signed-rank $p < 0.001$ for both R^2 and PCC (Appendix A.4.2).

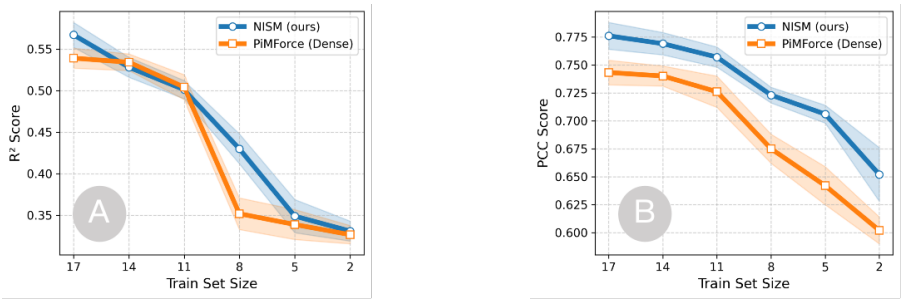
This result highlights the effectiveness of NiSM's structured inductive biases, including task-context modulation, sparse feature-to-finger mapping, and EMG-conditioned effort modulation, suggesting that explicit structure can recover part of the force-relevant information otherwise provided by denser sensing. Since PiMForce (dense) differs from NiSM in input modalities and architecture, and all models are retrained on PiMForce, this comparison evaluates the architectures under the same sparse-view setting and does not evaluate cross-dataset or cross-hardware generalization.

Across Users		Self-collected				PiMForce			
Models	**Para.**	**NRMSE(%)**	**MAE(%)**	**R²(%)**	**PCC(%)**	**NRMSE(%)**	**MAE(%)**	**R²(%)**	**PCC(%)**
LSTM-T [32]	5.07M	4.50±1.02	13.51±5.00	22.69±10.13	51.51±8.46	21.20±0.24	29.60±0.29	21.70±1.70	47.60±0.93
Informer [47]	11.02M	4.07±13.50	11.80±4.03	31.50±13.90	58.60±10.80	23.87±0.47	30.16±1.15	7.47±1.90	39.44±0.95
Crossformer [44]	26.34M	4.25±0.18	13.52±1.52	30.88±1.85	56.55±1.62	22.49±0.12	31.22±0.41	19.20±1.06	45.42±0.59
CA-T [41]	4.54M	6.55±0.42	8.29±1.28	31.91±2.12	63.96±4.89	28.42±0.76	38.43±3.71	25.38±4.50	52.87±4.43
PiMForce (dense)† [37]	66.39M	–	–	–	–	13.93±0.50	11.92±0.60	52.68±4.00	73.73±2.00
NiSM (ours)	**1.71M**	**3.20±0.50**	**7.60±1.20**	**47.00±4.40**	**68.50±3.10**	**11.10±0.20**	**8.69±0.30**	**54.40±2.90**	**76.60±1.40**

Table 1: Cross-user Performance Comparison of NiSM against Baselines on Self-collected and PiMForce Datasets. (lower values are better for NRMSE/MAE, higher values are better for R²/PCC). PiMForce model † was not evaluated on our self-collected dataset because it requires dense inputs (e.g., 8-channel sEMG) that our sparse setup does not provide; instead, we report PiMForce (dense) on the public dataset using its original 8-channel EMG and treat it as a strong dense EMG reference. All other sequence baselines use the 1-ch EMG view.

5.2 Does Structure Improve Data Efficiency under Limited User Data?

Sparse wearable force estimation is especially vulnerable to limited user coverage because both motion-force mappings and EMG-force relationships vary across individuals. Figure3(A, B) shows that NiSM degrades more gracefully as the number of training users decreases, suggesting that the proposed structural priors act as regularizers rather than merely increasing model capacity. These results reflect zero-shot generalization to unseen users; on PiMForce, calibrating with 5% of a test user's trials yields R^2 = 64.69% (Appendix A.4.4).



(a) R^2 Trend (b) PCC Trend

Figure 3: (A) R^2 and (B) PCC performance trends across varying train set sizes. Shaded bands denote the standard deviation across held-out test users.

5.3 Which Structural Priors Recover Missing Force-relevant Information?

Models	**NRMSE (%)**	**MAE (%)**	**R^2**	**PCC**
NiSM (full)	10.99	8.50	0.570	0.773
w/o Context Mod.	11.62	10.06	0.520	0.754
w/o SG Feedforward	11.35	8.81	0.541	0.771
w/o EMG Scaling	12.08	9.17	0.464	0.653

Table 2: Ablation study (cross-user). NRMSE and MAE are normalized and shown in percentage format.

We ablate NiSM's main structural components under the same sparse IMU-EMG test-time inputs. As shown in Table 2, removing context modulation lowers R^2 from 0.570 to 0.520, indicating that the IMU-derived grasp context helps disambiguate similar thumb motions across hand-shape and contact states. Removing sparse gating also degrades performance ($R^2 = 0.541$), supporting its role in modeling finger-specific kinematic coupling. The largest drop occurs without EMG-conditioned scaling ($R^2 = 0.464$, PCC $= 0.653$), showing that muscle-effort modulation provides force-relevant information beyond IMU kinematics.

Together, these results show that NiSM's gains come from complementary structural priors: inferred grasp context approximates unobserved task state, sparse gating encodes kinematic coupling, and EMG scaling supplies effort-dependent information. Appendix Fig. 7 shows stable feature-to-finger patterns in W_{sparse}, supporting that sparse gating learns structured dependencies.

Because the task condition is derived from the same thumb IMU as the kinematic feature, we further test whether it adds information beyond self-conditioning. When the task condition is swapped across actions with the other inputs fixed, learned target conditions redirect relative force allocation by 90.4%, compared with 8.5% for random vectors. On action pairs with matched thumb trajectories but different force patterns, task conditioning reduces force-difference recovery error by 14.74% (Appendix A.4.3).

6 Discussion and Limitations

Our results support the central trade-off studied in this work: structural inductive biases can partially compensate for reduced sensing density in finger-level force estimation. With a thumb-mounted IMU and a single-channel wrist EMG sensor, NiSM is competitive with dense-sensing references while using substantially fewer parameters, showing that explicitly encoding task context, kinematic coupling, and muscle-effort modulation recovers part of the force-relevant information typically provided by denser sensing.

Several limitations remain. First, context modulation depends on IMU-to-hand-shape inference: similar hand shapes may correspond to different object properties or contact conditions, especially for unseen objects, and tasks with weak thumb–finger coupling (e.g., typing or piano playing) lie outside NiSM's scope and may require additional sensors or explicit task cues. Second, NiSM estimates per-finger normal pressure rather than 3D forces or torques, which would require vector-valued supervision such as 3D tactile or six-axis force/torque sensing. Third, participants were healthy right-handed adults with a single self-donned placement per session; in a simulated ring rotation (Appendix A.4.5), R^2 drops by 2.6 points for random per-trial offsets up to 15° and by 23.4 points for a fixed 30° offset, so large mounting rotations call for orientation calibration; physical re-donning, wearing depth, calibration drift, and motion artifacts were not evaluated. Future work includes more expressive conditioning (e.g., cross-attention or mixture-of-experts), physically grounded priors such as joint, tendon, and contact constraints, and longer-term deployment in less constrained settings.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China (62302168, W2532047), the Guangdong S&T Programme (2025B0101120004), the Guangdong Provincial Department of Science and Technology (2023CX10X070), the Guangdong Basic and Applied Basic Research Foundation (2026A1515010254), the Guangdong Provincial Key Laboratory of Human Digital Twin (2022B1212010004), the Guangzhou Basic and Applied Basic Research Foundation (2024A04J6311), and the Shenzhen Holdfound Foundation Endowed Professorship. This work is partially supported by the High Performance Computing Platform of South China University of Technology.

The authors declare no competing interests.

References

[1] R. Aissaoui, A. De Lutiis, A. Feghoul, and F. Chénier. Handrim reaction force and moment assessment using a minimal imu configuration and non-linear modeling approach during manual wheelchair propulsion. *Sensors*, 24(19), 2024.

[2] E. F. Austin Jr, C. P. Kearney, P. J. Chacon, S. A. Winges, P. Acharya, and J.-W. Choi. A fabricated force glove that measures hand forces during activities of daily living. *Sensors*, 22(4):1330, 2022.

[3] L. Behnke, L. Sanchez-Botero, W. R. Johnson, A. Agrawala, and R. Kramer-Bottiglio. Dynamic hand proprioception via a wearable glove with fabric sensors. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 149–154, 2023.

[4] I. Bernabucci, S. Conforto, M. Capozza, N. Accornero, M. Schmid, and T. D'Alessio. A biologically inspired neural network controller for ballistic arm movements. *Journal of Neuro-Engineering and Rehabilitation*, 4:1–17, 2007.

[5] N. Bhattasali, A. M. Zador, and T. Engel. Neural circuit architectural priors for embodied control. *Advances in neural information processing systems*, 35:12744–12759, 2022.

[6] A. Billard and A. J. Ijspeert. Biologically inspired neural controllers for motor control in a quadruped robot. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 6, pages 637–641. IEEE, 2000.

[7] N. Chen, G. Westling, B. B. Edin, and P. van der Smagt. Estimating fingertip forces, torques, and local curvatures from fingernail images. *Robotica*, 38(7):1242–1262, 2020.

[8] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville. Modulating early visual processing by language. *Advances in neural information processing systems*, 30, 2017.

[9] K. Ehsani, S. Tulsiani, S. Gupta, A. Farhadi, and A. Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 221–230, 2020.

[10] N. Fallahinia and S. A. Mascaro. Real-time tactile grasp force sensing using fingernail imaging via deep neural networks. *IEEE Robotics and Automation Letters*, 7(3):6558–6565, 2022.

[11] B. Fang, C. Wang, F. Sun, Z. Chen, J. Shan, H. Liu, W. Ding, and W. Liang. Simultaneous semg recognition of gestures and force levels for interaction with prosthetic hand. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30:2426–2436, 2022.

[12] J. P. Fernández, M. A. Vargas, J. M. V. García, J. A. C. Carrillo, and J. J. C. Aguilar. A biological-like controller using improved spiking neural networks. *Neurocomputing*, 463:237–250, 2021.

[13] B. Fialkoff, H. Hadad, D. Santos, F. Simini, and M. David. Hand grip force estimation via emg imaging. *Biomedical Signal Processing and Control*, 74:103550, 2022.

[14] C. Gao, Y. Yang, and W. Li. 3d interacting hand pose and shape estimation from a single rgb image. *Neurocomputing*, 474:25–36, 2022.

[15] Y. Geng, Z. Yu, Y. Long, L. Qin, Z. Chen, Y. Li, X. Guo, and G. Li. A cnn-attention network for continuous estimation of finger kinematics from surface electromyography. *IEEE Robotics and Automation Letters*, 7(3):6297–6304, 2022.

[16] P. Grady, J. A. Collins, C. Tang, C. D. Twigg, K. Aneja, J. Hays, and C. C. Kemp. Pressure-vision++: Estimating fingertip pressure from diverse rgb images. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8683–8693, 2024.

[17] R. J. Guerrero-Criollo, J. A. Castaño-López, J. Hurtado-López, and D. F. Ramirez-Moreno. Bio-inspired neural networks for decision-making mechanisms and neuromodulation for motor control in a differential robot. *Frontiers in Neurorobotics*, 17:1078074, 2023.

[18] R. Guo, C. Chen, and X. Zhu. Simultaneous and proportional real-time control of three-dof movements based on motor unit discharges. *IEEE Journal of Biomedical and Health Informatics*, 29(4):2642–2653, 2025.

[19] G. Hajian, E. Morin, and A. Etemad. Multimodal estimation of endpoint force during quasi-dynamic and dynamic muscle contractions using deep learning. *IEEE Transactions on Instrumentation and Measurement*, 71:1–11, 2022.

[20] S. Jiang, B. Lv, W. Guo, C. Zhang, H. Wang, X. Sheng, and P. B. Shull. Feasibility of wrist-worn, real-time hand, and surface gesture recognition via semg and imu sensing. *IEEE Transactions on Industrial Informatics*, 14(8):3376–3385, 2018.

[21] D.-K. Ko, K.-W. Lee, D. H. Lee, and S.-C. Lim. Vision-based interaction force estimation for robot grip motion without tactile/force sensor. *Expert Systems with Applications*, 211:118441, 2023.

[22] F. Leone, C. Gentile, A. L. Ciancio, E. Gruppioni, A. Davalli, R. Sacchetti, E. Guglielmelli, and L. Zollo. Simultaneous semg classification of hand/wrist gestures and forces. *Frontiers in Neurorobotics*, Volume 13 - 2019, 2019.

[23] D. Li, P. Kang, Y. Yu, and P. B. Shull. Graph-driven simultaneous and proportional estimation of wrist angle and grasp force via high-density emg. *IEEE Journal of Biomedical and Health Informatics*, 28(5):2723–2732, 2024.

[24] Z. Li, J. Sedlar, J. Carpentier, I. Laptev, N. Mansard, and J. Sivic. Estimating 3d motion and forces of person-object interactions from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8640–8649, 2019.

[25] C. Liang, C. Yu, Y. Qin, Y. Wang, and Y. Shi. Dualring: Enabling subtle and expressive hand interaction with dual imu rings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–27, 2021.

[26] H. Liu, X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S.-C. Zhu. A glove-based system for studying hand-object manipulation via joint pose and force sensing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6617–6624, 2017.

[27] Y. Liu, S. Zhang, and M. Gowda. Neuropose: 3d hand pose tracking using emg wearables. In *Proceedings of the Web Conference 2021*, WWW '21, page 1471–1482, New York, NY, USA, 2021. Association for Computing Machinery.

[28] Y. Liu, S. Zhang, and M. Gowda. A practical system for 3-d hand pose tracking using emg wearables with applications to prosthetics and user interfaces. *IEEE Internet of Things Journal*, 10(4):3407–3427, 2023.

[29] K. Mahdikhanlou and H. Ebrahimnezhad. 3d hand pose estimation from a single rgb image by weighting the occlusion and classification. *Pattern Recognition*, 136:109217, 2023.

[30] H. Mao, Y. Zheng, C. Ma, K. Wu, G. Li, and P. Fang. Simultaneous estimation of grip force and wrist angles by surface electromyography and acceleration signals. *Biomedical Signal Processing and Control*, 79:104088, 2023.

[31] S. Mascaro and H. Asada. Measurement of finger posture and three-axis fingertip touch force using fingernail sensors. *IEEE Transactions on Robotics and Automation*, 20(1):26–35, 2004.

[32] V. Pentsos, S. Tragoudas, J. Wibbenmeyer, and N. Khdeer. A hybrid lstm-transformer model for power load forecasting. *IEEE Transactions on Smart Grid*, 2025.

[33] T.-H. Pham, A. Kheddar, A. Qammaz, and A. A. Argyros. Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2810–2819, 2015.

[34] N. Rubin, Y. Zheng, H. Huang, and X. Hu. Finger force estimation using motor unit discharges across forearm postures. *IEEE Transactions on Biomedical Engineering*, 69(9):2767–2775, 2022.

[35] A. Salatiello and M. A. Giese. Continuous decoding of daily-life hand movements from forearm muscle activity for enhanced myoelectric control of hand prostheses. *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[36] K. Seo, J. Seo, H. Jeong, S. Kim, and S. H. Yoon. Posture-informed muscular force learning for robust hand pressure estimation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[37] K. Seo, J. Seo, H. Jeong, S. Kim, and S. H. Yoon. Posture-informed muscular force learning for robust hand pressure estimation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 87831–87873. Curran Associates, Inc., 2024.

[38] I. H. Taylor, S. Dong, and A. Rodriguez. Gelslim 3.0: High-resolution measurement of shape, force and slip in a compact tactile-sensing finger. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10781–10787. IEEE, 2022.

[39] S. O. Verduzco-Flores and E. De Schutter. Self-configuring feedback loops for sensorimotor control. *Elife*, 11:e77216, 2022.

[40] J. P. Vásconez, L. I. Barona López, L. Valdivieso Caraguay, and M. E. Benalcázar. Hand gesture recognition using emg-imu signals and deep q-networks. *Sensors*, 22(24), 2022.

[41] Y. Xiao, Z. Huang, and Y. Gao. From wrist to finger: Hand pose tracking using ring-watch wearables. CHI EA '25, New York, NY, USA, 2025. Association for Computing Machinery.

[42] T. C. Yu, G. Hu, R. Zhang, H. Lim, S. Mahmud, C.-J. Lee, K. Li, D. Agarwal, S. Nie, J. Oh, F. Guimbretière, and C. Zhang. Ring-a-pose: A ring for continuous hand pose tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(4), Nov. 2024.

[43] O. Zahra, S. Tolu, and D. Navarro-Alarcon. Differential mapping spiking neural network for sensor-based robot control. *Bioinspiration & Biomimetics*, 16(3):036008, 2021.

[44] Y. Zhang and J. Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

[45] Y. Zhao, T. Kwon, P. Streli, M. Pollefeys, and C. Holz. Egopressure: A dataset for hand pressure and pose estimation in egocentric vision. *arXiv preprint arXiv:2409.02224*, 2024.

[46] H. Zhou, T. Lu, Y. Liu, S. Zhang, and M. Gowda. Learning on the rings: Self-supervised 3d finger motion tracking using wearable sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(2), July 2022.

[47] 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*, volume 35, pages 11106–11115, 2021.

A Technical Appendices and Supplementary Material

A.1 Dataset details

A.1.1 Self-collected Dataset

Participant Recruitment We recruited 20 healthy right-handed adult volunteers to participate in the data collection study. All participants provided written informed consent prior to enrollment. Demographic information was anonymized in compliance with privacy standards.

Inclusion criteria included the absence of neuromuscular or musculoskeletal disorders affecting the upper limbs, and the ability to complete the full range of gesture tasks. Exclusion criteria included recent wrist or hand injuries, or the use of assistive electronic devices.

Ethical Approval All participants provided written informed consent prior to enrollment, and all procedures were conducted in accordance with approved IRB protocols.

Experimental Protocol Participants were instructed to perform 20 distinct hand gestures and grasp actions, categorized into three groups: baseline postures, rhythmic finger motion, and object manipulation.

- **Basic Hand Gesture Tasks**
  - **Palm-up rest:** Hand placed palm-up on a table with fingers naturally extended for 30 seconds.
  - **Palm-down rest:** Hand placed palm-down with fingers loosely resting on the surface for 30 seconds.
  - **Suspended posture:** Hand held above the table, palm facing downward, with fingers slightly flexed and avoiding contact for 30 seconds.
  - **Rhythmic Finger Tapping:** Participants performed a rhythmic tapping task using the index, middle, and ring fingers, following a 1–2–3 pattern at a frequency of 2 Hz. Each cycle consisted of 6 seconds of tapping followed by 2 seconds of rest, repeated for 10 cycles (approximately 1 minute). The wrist was elevated to ensure motion originated exclusively from the fingers.
- **Object Interaction Tasks** Participants interacted with 16 real-world objects, each selected to represent diverse physical characteristics, as shown in Figure 4. For each object, participants performed five grasp-and-release repetitions, lifting the object from a table and placing it back naturally. Objects were grouped by functional context:
  - **Containers and packages:** Soda can, milk carton, thermos, toothpaste box, playing cards, adhesive tape
  - **Tools and electronics:** Screwdriver, electric shaver, smartphone, computer mouse
  - **Personal and soft items:** Black rubber ball, small ball, glasses case
  - **Utensils:** Large glass cup, spoon
  - **Other:** Cardboard game box

Sensor Setup We used a sparse wearable sensing setup for model inputs and separate fingertip sensors for ground-truth supervision. The model inputs consist of a thumb-mounted IMU and a single-channel wrist EMG sensor, while five fingertip pressure sensors provide synchronized supervision labels.

- **Thumb-mounted IMU.** The system incorporates a custom-designed ring that is comfortably worn on the thumb, housing an InvenSense ICM-20948 9-axis IMU (Inertial Measurement Unit). This device integrates both an accelerometer and a gyroscope, capturing detailed movement and orientation data across three axes. The IMU samples data at a rate of 55 Hz, providing a high level of precision for motion tracking. Data is wirelessly transmitted via Bluetooth Low Energy (BLE), a protocol selected for its low power consumption, ensuring efficient long-duration use without significant battery drain. This setup captures the dynamic motion of the thumb, a key component in hand interaction, given its frequent involvement in



Figure 4: The 16 real-world objects used in the study, selected to represent diverse physical characteristics.

most hand gestures. No instructions were given on ring orientation, rotational alignment, or wearing depth, and each session used a single donning.

- **Wrist-worn EMG sensor.** A single-channel dry-electrode EMG sensor is embedded in a smartwatch-like form factor and positioned on the dorsal forearm, near the wrist. The EMG signals, which measure electrical muscle activity, are amplified and band-pass filtered within 20–450 Hz, with notch filtering at 50 Hz to remove powerline interference. The signal is digitized at 1 kHz and transmitted via BLE to a nearby host. This sensor provides a coarse muscle-effort cue related to grasp force, complementing the thumb IMU without requiring multi-channel EMG arrays across the forearm or hand.
- **Fingertip pressure sensors.** Tekscan FlexiForce A402 sensors were placed on all five fingertips using a lightweight, flexible glove. These sensors recorded fingertip normal-force readings during interaction, which were used as ground-truth contact-pressure labels after normalization. The sensors sampled at 100 Hz and were integrated with an Arduino logger for data acquisition. These pressure labels were synchronized with the thumb IMU and wrist EMG streams using shared timestamps, alongside a host-side heartbeat signal at 100 ms intervals to maintain temporal alignment across all modalities.

Data Acquisition and Preprocessing

- **Logging and Synchronization.** IMU and EMG signals were streamed over Bluetooth Low Energy (BLE) to a host machine, while force data were acquired via USB serial from an Arduino logger. All data streams were timestamped based on host-side receipt time. The datasets were pre-aligned before processing, assuming synchronized start and consistent sampling rates across modalities.
- **Sliding Window Segmentation.** Multimodal sensor data were segmented into sliding windows of 30 frames with a stride of 2 samples, corresponding to approximately 250 ms per segment with high temporal overlap. For each segment, the final-frame pressure vector was used as the main regression target, while pressure values within the window were retained for computing the peak-pressure loss term. The window thus contains no frames after the target, and evaluation is window-based rather than zero-latency streaming. The length $T = 30$ was chosen empirically; on PiMForce, shorter windows ($T = 10, 20$) reduce latency but degrade accuracy (Table 7).
- **EMG Feature Extraction.** Each EMG segment was transformed into a fixed-length 10-dimensional feature vector using a standardized pipeline. This included common time-domain descriptors such as mean absolute value, root mean square (RMS), zero-crossing rate, waveform length, variance, amplitude sum, and standard deviation. In addition, frequency-domain features were computed via fast Fourier transform (FFT), including mean power spectral density (PSD), peak-frequency index, and total spectral energy. These features capture both transient and periodic components of muscle activity relevant to grasp and movement intention.
- **Normalization and Scaling.** To reduce inter-subject variability and enable generalization, each modality was independently normalized using scalers fit on the training set. IMU and EMG features were standardized using StandardScaler, while contact-pressure labels were scaled to $[0, 1]$ using MinMaxScaler per finger. The same transformations were applied during evaluation to ensure consistency.
- **Final Feature Composition.** Each processed sample included four components:
  - A 30-frame IMU sequence (with 12 dimensions per frame, including 3D acceleration and 9D orientation),
  - A 10-dimensional vector of EMG features,
  - A 5-dimensional normalized contact-pressure vector representing fingertip pressure labels.

  Optionally, the raw EMG value at the final timestep was retained for visualization or auxiliary analysis. These composite inputs formed the basis for multimodal supervised learning in our model pipeline.

Code and dataset Code is available at `https://github.com/3525starmoon-glitch/NiSM` under the MIT License. The dataset is available upon request for non-commercial research (CC BY-NC 4.0 terms); participants are identified only by anonymous IDs, raw video is not shared, and requesters must not attempt to re-identify participants.

A.1.2 Public Benchmark Dataset (PiMForce)

We evaluate our model on the publicly available PiMForce dataset, which contains synchronized multimodal recordings of 21 right-handed participants performing 22 grasp and manipulation tasks (indexed 0–21 in the released data). Each subject completed three sessions of repeated trials, with data stored hierarchically as `User{ID}/Session{ID}/data_{task_id}.npy`. Each file includes multimodal signals such as multi-channel EMG, inertial measurements, 3D hand pose, and fingertip/palm pressure measurements. In our evaluation, pressure measurements are used as supervision and are projected to a unified five-finger contact-pressure space.

To evaluate sparse-input models under the same sensing constraints as our ring-watch setup, we construct sparse inputs from a 12-dimensional IMU representation and a single-channel EMG signal. Specifically, the 3D acceleration and rotation matrix are concatenated to form the IMU representation, and one EMG channel is selected to match the single-channel wrist EMG setting. For the dense PiMForce reference, we use the dense modalities available in PiMForce, including multi-channel EMG and 3D hand posture, while pressure measurements remain the supervision target. All modalities are temporally aligned, normalized using training-set statistics, and segmented into overlapping windows of 30 frames.

A total of 4 users are held out for testing, while the remaining 17 users are used for training. For training users, only session 1 is included. For each test user, all three sessions are used. Out of consideration for training time, only half of the training data is used. This results in approximately 522,292 (half) training samples and 122,893 test samples. All processed data are stored as windowed numpy arrays and used consistently across model variants to ensure fair and reproducible evaluation.

A.2 Model Implementation Details

All NiSM experiments except the rotation simulation (Appendix A.4.5), which used one NVIDIA A800 GPU, were conducted on a system equipped with an Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz and an NVIDIA GeForce RTX 4090 GPU; baselines whose hardware differs are noted in their own paragraphs below. The deep learning framework utilized was PyTorch, version 2.5.1, running in a Python 3.9 environment.

A.2.1 NiSM Architecture Details

The neuro-inspired structured architecture projects sparse multimodal inputs into a shared 48-dimensional latent space ($d_z = 48$). The IMUEncoder processes 12-dimensional IMU sequences using a two-layer bidirectional LSTM and outputs a 48-dimensional kinematic feature. In parallel, a HandShapeLatentEncoder maps the IMU representation to an IMU-derived hand-shape latent, which is further transformed by a two-layer MLP-based ContextEncoder into a 48-dimensional inferred grasp-context feature. When object/action labels are available during training, an auxiliary classification head is attached to this context feature, but these labels are not embedded or provided to the force predictor at inference.

The structured prediction modules then use this inferred context feature to condition force estimation. Conditional Batch Normalization normalizes the IMU feature, with scale γ and shift β dynamically predicted from the grasp-context feature. Base pressure predictions for five fingers are generated through a sparse feature-to-finger projection regularized by the ℓ_1 sparsity term described in Section 3.3. The predictions are further gated by a ContextMask module conditioned on the inferred grasp-context feature. Finally, the EMGScalingModule maps the 48-dimensional EMG feature to five finger-wise scaling factors using a two-layer MLP with ReLU activation and 0.2 dropout, followed by a Sigmoid activation to produce $\mathbf{s}_{\mathrm{EMG}} \in [0, 1]^5$.

A.2.2 Baseline Architecture Details

LSTM-Trans

- **Model Implementation.**We adopt a hybrid Transformer-LSTM model as a baseline to infer multi-channel finger pressure signals from multimodal sensor inputs, including electromyography (EMG) and inertial measurement unit (IMU) signals. The model architecture and training configuration follow the original implementation without modification and serve as a reference point for evaluating our proposed method. Each input sample is a fixed-length sequence of 30 time steps, where each time step consists of concatenated EMG and IMU features. The model first processes the input sequence using two stacked LSTM layers (each with 128 hidden units), which extract temporal dependencies from the raw multimodal signals. The LSTM output is passed through two Transformer encoder blocks, each comprising a multi-head self-attention mechanism (4 heads, 64-dimensional each), dropout (rate = 0.2), and a position-wise feed-forward convolutional module (channels: 128 and 64) with ReLU activation. Layer normalization and residual connections are used to enhance stability and convergence. The output is then passed through a series of fully connected layers and a final linear projection layer to regress the target finger pressure sequence across 5 pressure channels.
- **Training Configuration.**The model is trained using the Adam optimizer with an initial learning rate of 1×10^{-3}. The loss function is mean squared error (MSE), and evaluation metrics include root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), coefficient of determination (R^2), Pearson correlation coefficient (PCC), and Spearman correlation. We use a batch size of 64 and train for up to 50 epochs with early stopping and learning rate scheduling based on validation loss. Input and output signals are min-max normalized before training, and the model's predictions are inverse-transformed for evaluation.This model is used solely as a baseline under the original parameter setting, without tuning or structural modifications.

Informer

- **Model Implementation.**We build upon the official Informer architecture, originally designed for long-term time-series forecasting. The source code implements an encoder-decoder framework with ProbSparse attention for $\mathcal{O}(L \log L)$ time complexity. We retain the original modular design, including the *Encoder*, *Decoder*, *DataEmbedding*, *AttentionLayer*, and *ConvLayer*, while only making minimal task-specific modifications to accommodate our finger-pressure inference task.

  Specifically, we adapt the model to the short-term forecast setting ($taskname = ' shorttermforecast'$) by using synchronized IMU and EMG sequences as input and finger pressure as target. Each time step of input uses the same sparse IMU and EMG view as the other baselines, and the model is trained to predict a 5-dimensional finger pressure vector for each output time step. The encoder and decoder are initialized with two layers each, 4 attention heads, 128-dimensional model embeddings, and 512 feed-forward dimensions. The sinusoidal positional embedding and convolutional distillation layers are preserved. Input and target sequences are scaled using per-sample z-score normalization before encoding, and inverse transformed after decoding.The original forward pass structure is preserved. The *short forecast* method is used as the primary inference function. The *long forecast*, imputation, and classification branches are disabled in our task. No attention visualization or output-attention flag is used in final evaluation.
- **Training Configuration.**The model is trained using the Adam optimizer with an initial learning rate of 5×10^{-4}, decayed by a factor of 0.1 upon validation loss stagnation. The training loss is mean squared error (MSE), and the evaluation metrics include RMSE, MAE, MAPE, coefficient of determination (R^2), Pearson correlation coefficient (PCC), and Spearman correlation coefficient. A batch size of 32 is used, and training is conducted for up to 80 epochs with early stopping (patience = 10) and checkpointing based on validation RMSE. Both input sensor data and ground truth finger pressure values are normalized to [0, 1] using min-max scaling before training, and inverse transformations are applied to predictions for evaluation. All experiments are conducted on a single NVIDIA RTX 3090 GPU. No parameter tuning or additional modules are introduced beyond the minimal modifications required for the task.

Crossformer

- **Model Implementation.**We build upon the official Crossformer architecture [44], which segments each input variable into patches and applies two-stage attention across the time and variable dimensions to capture cross-dimension dependencies. We retain the original modular design and only make minimal task-specific modifications so that the model takes the same synchronized EMG and IMU windows (30 time steps) as the other baselines and outputs the 5-dimensional finger pressure.
- **Training Configuration.**The model is trained with the default hyperparameters of the official implementation, without additional tuning, and is evaluated on the same data splits and metrics as the other baselines.

PiMForce-Dense.

- **Model Implementation.** As a dense-sensing reference, we retain most of the original PiMForce architecture with minimal adaptations to our five-finger pressure evaluation. The model estimates finger contact pressure from the dense sensing modalities available in PiMForce, including multi-channel EMG and 3D hand posture, while pressure measurements are used only as supervision. The model adopts a dual-stream structure combining a ResNet-50 backbone for pose encoding and a customized EMGNet module for EMG signal processing.

  In parallel, 3D hand-pose sequences are processed using a ResNet-50 network to extract a 512-dimensional pose representation. The two embeddings are concatenated to form a 1024-dimensional joint feature, which is passed through a three-layer MLP with hidden dimensions of 256 and 256, and finally projected to a vector of size $F_L \times F \times 32$, where F_L is the number of force levels and F is the number of finger force channels. All layers in the MLP use ReLU activations and are followed by Batch Normalization. The final output is reshaped to $(B, F_L, F, 32)$, where B denotes the batch size, providing a structured representation for downstream pressure regression tasks.
- **Training Configuration.** Starting from the published hyperparameters, we searched the learning rate and batch size on a validation split of the training data; the values below are the selected ones. The Adam optimizer is employed with an initial learning rate of 1×10^{-4}, and the learning rate is reduced by a factor of 0.2 if the monitored metric plateaus (the training loss in the fixed split and the validation RMSE in the leave-one-subject-out evaluation). Mean Absolute Error (MAE) is chosen as the loss function. The model is evaluated using metrics such as Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Pearson Correlation Coefficient (PCC), Spearman Rank Correlation, and the coefficient of determination R^2. The training process uses a batch size of 64 for up to 100 epochs. In the fixed split, the final-epoch model is used; in the leave-one-subject-out evaluation, training stops after 15 epochs without improvement, and the checkpoint with the lowest RMSE on the separate validation participant is used. Both input sensor data and the ground truth are normalized to the $[0, 1]$ range using min-max scaling prior to training, and inverse scaling is applied to predictions during evaluation. All experiments are performed on a single NVIDIA A800 GPU with 80GB of VRAM. No architectural adjustments are made beyond the minimal task-specific changes.

CA-T

- **Model Implementation.** The model is a multimodal Transformer-based architecture designed for joint prediction of fingertip force and hand pose from wearable IMU and EMG inputs. The IMU stream is processed using a 1D convolutional encoder followed by a 2-layer Transformer encoder with 4 attention heads and 128-dimensional hidden states. In parallel, the EMG signal is encoded using a 2-layer LSTM with the same hidden size. To enable cross-modal reasoning, we insert bidirectional cross-attention modules between the IMU and EMG branches. The resulting fused representations are decoded through two Transformer decoders—one for 3D hand pose and another for fingertip pressure—each comprising 2 layers and 4 heads. Final predictions are obtained through task-specific MLP heads. In addition to the main outputs, the model includes auxiliary branches for unimodal predictions (IMU-only, EMG-only), and a cross-attention-based predictor, enabling multi-level supervision across all intermediate stages.

- **Training Configuration.** The model is trained using the Adam optimizer with an initial learning rate of 1×10^{-3}, which is decayed by a factor of 0.5 every 30 epochs. During fine-tuning, a smaller learning rate of 1×10^{-4} and longer decay schedule (every 100 epochs) are applied. We use a batch size of 256 and train for up to 100 epochs. Training is supervised by a composite objective comprising MSE loss for pose prediction and Smooth L1 loss for pressure estimation. Auxiliary predictions from the unimodal branches and cross-attention module are also supervised with weighted losses. The loss weights are empirically set as: 1.0 for final pose, 3.0 for final pressure, 0.3 for unimodal pose heads, 0.6 for unimodal pressure heads, and 0.5 for the cross-attention output. All models are trained and evaluated on a single NVIDIA GPU using PyTorch. In the fixed split, the final-epoch model is used; in the leave-one-subject-out evaluation, early stopping (patience 50 epochs) is applied based on the loss of the separate validation participant.

A.3 Training Procedure

The model was trained utilizing the Adam optimizer, configured with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}; the Adam optimizer's beta parameters defaulted to standard PyTorch values. A ReduceLROnPlateau learning rate scheduling strategy was employed, which monitored the training loss in the fixed split, where no held-out data were used, and the loss of the separate validation participant in the leave-one-subject-out evaluation. The learning rate was reduced by a factor of 0.5 if the monitored metric failed to demonstrate an improvement of at least 1×10^{-3} over 3 consecutive epochs. Training was conducted with a batch size of 256 for 100 epochs, and the final-epoch model was used; in the leave-one-subject-out evaluation (Section 5.1), the checkpoint was instead selected on a separate non-test participant.

The composite loss function incorporated a Mean Squared Error (MSE) task loss, L_1 sparsity regularization terms for the SparseGatedFeedforward module ($\lambda_{SGF_{sparse}} = 0.0001$) and the context-mask module ($\lambda_{\text{mask}_{sparse}} = 0.0001$), alongside a peak-weighted error component. This peak-weighted error component, designed to penalize under-predictions at signal peaks—defined as instances where ground truth pressure exceeded 0.7—was assigned a weight of $\lambda_{peak} = 2$.

A.4 Evaluation Details

A.4.1 Evaluation Metrics

We report four standard regression metrics to quantitatively evaluate the accuracy of finger force prediction: Normalized Root Mean Square Error (**NRMSE**), Mean Absolute Error (**MAE**), Coefficient of Determination ($\boldsymbol{R}^2$), and Pearson Correlation Coefficient (**PCC**). These metrics are computed on a per-frame, per-finger basis and averaged over the entire test set.

Let $\hat{\mathbf{y}}_i \in \mathbb{R}^d$ denote the predicted force vector for sample i, and $\mathbf{y}_i \in \mathbb{R}^d$ the corresponding ground truth, where $d = 5$ is the number of fingers. Let N be the number of test samples.

MAE measures the average magnitude of absolute errors, regardless of direction:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{d} \sum_{j=1}^{d} |\hat{y}_{ij} - y_{ij}| \tag{10}$$

NRMSE provides a scale-invariant measure of prediction error, normalized by the dynamic range of the ground truth per finger:

$$\text{NRMSE} = \frac{1}{d} \sum_{j=1}^{d} \frac{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_{ij} - y_{ij})^2}}{\max(y_{\cdot j}) - \min(y_{\cdot j})} \tag{11}$$

$\boldsymbol{R}^2$ quantifies the proportion of variance in the ground truth that is explained by the model predictions:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \sum_{j=1}^{d} (\hat{y}_{ij} - y_{ij})^2}{\sum_{i=1}^{N} \sum_{j=1}^{d} (y_{ij} - \bar{y}_j)^2} \tag{12}$$

where $\bar{y}_j = \frac{1}{N}\sum_{i=1}^{N} y_{ij}$ is the mean force value for the j-th finger.

PCC evaluates the linear correlation between predicted and true values, averaged across fingers:

$$\text{PCC} = \frac{1}{d}\sum_{j=1}^{d} \frac{\sum_{i=1}^{N}\left(\hat{y}_{ij} - \bar{\hat{y}}_j\right)\left(y_{ij} - \bar{y}_j\right)}{\sqrt{\sum_{i=1}^{N}\left(\hat{y}_{ij} - \bar{\hat{y}}_j\right)^2}\sqrt{\sum_{i=1}^{N}\left(y_{ij} - \bar{y}_j\right)^2}} \tag{13}$$

Here, $\bar{\hat{y}}_j$ and $\bar{y}_j$ denote the mean predicted and ground-truth forces for the j-th finger, respectively.

These metrics provide a comprehensive view of both absolute accuracy (MAE, NRMSE) and relative prediction quality (R^2, PCC). All results reported in the main paper were computed using the above formulations without modification.

A.4.2 Paired Statistical Comparison

For the leave-one-subject-out comparison in Section 5.1, all compared models are retrained from scratch in every fold under the same partition and evaluated on the same held-out subject. Each model is trained once per fold with a fixed random seed, so the resulting statistics reflect between-subject rather than initialization variability. For R^2 and PCC, let d_i be the paired difference on subject i ($i = 1, \ldots, n$, $n = 21$), computed as NiSM minus the compared model so that positive values favor NiSM, and let $\bar{d}$ and s_d be their mean and standard deviation. We report the 95% confidence interval and the effect size

$$\bar{d} \pm t_{0.975,\,n-1}\,\frac{s_d}{\sqrt{n}}, \qquad d_z = \frac{\bar{d}}{s_d},$$

together with the number of subjects for which NiSM is better and the two-sided Wilcoxon signed-rank p-value. Results are summarized in Table 3.

Comparison	Metric	Improvement [95% CI]	d_z	Subjects improved	Wilcoxon p
NiSM vs. CA-T	R^2 gain	+0.300 [0.226, 0.374]	1.84	21/21	< 0.001
	PCC gain	+0.193 [0.157, 0.229]	2.41	21/21	< 0.001
NiSM vs. PiMForce (dense)	R^2 gain	+0.167 [0.128, 0.206]	1.96	20/21	< 0.001
	PCC gain	+0.113 [0.089, 0.137]	2.18	21/21	< 0.001

Table 3: Paired comparison over 21 leave-one-subject-out folds on PiMForce. Positive values favor NiSM.

A.4.3 Analysis of the Inferred Task Condition

Using the trained model without retraining, we test whether the inferred task condition has a target-specific effect and whether it helps when thumb trajectories are ambiguous.

Task-condition swaps. Across all $22 \times 21 = 462$ ordered non-identity action swaps on PiMForce, we fix the model, $\mathbf{z}_{\text{IMU}}$, and $\mathbf{z}_{\text{EMG}}$, and inject either the learned target condition or a task-agnostic random vector. The task-condition redirection ratio (TCR) measures source→target progress of the prediction (0% = source, 100% = target); gap weighting accounts for source–target separation. Learned swaps strongly redirect relative force allocation, whereas random vectors have little effect; the effect is weaker but still clear for total-force magnitude (Table 4).

Condition	Output	Gap-weighted TCR [95% CI]	Median	≥80%	≥95%
Learned target	Relative force allocation	90.4% [88.8%, 91.7%]	89.6%	81.2%	17.1%
Learned target	Total-force magnitude	56.9% [46.7%, 64.9%]	51.4%	21.2%	1.5%
Random vector	Relative force allocation	8.5% [4.0%, 13.2%]	5.8%	1.1%	0.0%
Random vector	Total-force magnitude	−1.3% [−13.3%, 9.7%]	−8.0%	5.6%	0.0%

Table 4: Learned task-condition swaps versus a random-vector control on PiMForce (462 ordered action swaps). TCR: 0% = source, 100% = target.

Matched thumb trajectories. We select action pairs whose thumb trajectories are nearly identical (palm-normalized DTW ≤ 0.015) but whose force patterns differ; all eligible pairs are included (3 pairs, 168 samples). Recovery error measures the mismatch between the predicted and true force difference within a pair. Task conditioning reduces recovery error from 0.81 to 0.69 (-14.74%), with improvement in every pair (Table 5).

Action pair	Samples	DTW	Recovery error (z_{IMU}-context $\rightarrow$ task condition)
TI-Pinch vs. TM-Pinch	58	0.01	$0.83 \rightarrow 0.71$ (-14.51%)
Sphere 3 Finger vs. Quadpod	59	0.01	$1.03 \rightarrow 0.87$ (-15.52%)
M-Press vs. P-Press	51	0.01	$0.52 \rightarrow 0.45$ (-13.37%)
Overall	168	0.01	$0.81 \rightarrow 0.69$ (-14.74%)

Table 5: Task conditioning improves force-difference recovery under matched thumb trajectories. DTW is palm-normalized (lower = more similar); lower recovery error is better.

A.4.4 Few-shot User Calibration

Because EMG is user-dependent, we complement the zero-shot cross-user evaluation with few-shot calibration on PiMForce: a fraction of each test user's trials is used for calibration, and non-overlapping trials of the same user are used for evaluation. Results are shown in Table 6, where the 0% row is the zero-shot setting. Most metrics improve with calibration, but MAE is not strictly monotonic from 5% to 20%.

Calibration	NRMSE (%)	MAE (%)	R^2 (%)	PCC (%)
0%	11.10 ± 0.20	8.69 ± 0.30	54.40 ± 2.90	76.60 ± 1.40
5%	9.07 ± 0.08	6.08 ± 0.68	64.69 ± 7.90	81.76 ± 3.99
20%	5.75 ± 0.71	6.84 ± 1.27	67.90 ± 8.46	83.37 ± 4.27

Table 6: Few-shot calibration on PiMForce using a fraction of each test user's trials.

T	NRMSE (%)	MAE (%)	R^2 (%)	PCC (%)
10	18.84 ± 0.27	11.36 ± 0.58	38.29 ± 0.49	70.55 ± 0.67
20	18.43 ± 0.28	10.42 ± 0.20	41.20 ± 0.60	72.83 ± 0.20
30	11.10 ± 0.20	8.69 ± 0.30	54.40 ± 2.90	76.60 ± 1.40

Table 7: Effect of the temporal window length T on PiMForce; $T = 30$ is the default setting.

A.4.5 Simulated Ring Rotation

We simulate ring rotation by rotating the IMU inputs of a trained model at test time with a mounting offset R_{off} ($\mathbf{a} \rightarrow R_{\text{off}}^{\top}\mathbf{a}$, $R \rightarrow R\,R_{\text{off}}$). The model is trained within participants on the self-collected data: each trial is split in time into 60% training, 10% checkpoint selection, and 30% testing; pressure is zeroed by each participant's no-contact offset, and EMG features are averaged over 100-sample sub-windows. Without rotation, the model reaches R^2 = 23.30%. Random per-trial offsets of up to 15°/30° reduce R^2 by 2.6/8.5 points, and fixed offsets of 10°/20°/30°/45° about random axes reduce it by 3.9/12.5/23.4/41.9 points (averaged over three random draws). NiSM thus tolerates small per-trial offsets, whereas large fixed offsets call for orientation calibration.

A.4.6 Per-Finger Performance Analysis

The provided performance metrics reveal variations in model accuracy across different fingers. Specifically, the thumb (Finger 1) achieves the lowest normalized MAE, suggesting highly accurate predictions, whereas the error gradually increases for fingers 2 through 5, peaking at Finger 5. The Pearson Correlation Coefficient (PCC) and coefficient of determination (R^2) consistently remain high across all fingers, indicating robust linear relationships and good explanatory power of the model, though slightly declining from thumb to pinky. NRMSE remains stable and relatively low, further

confirming the model's overall reliability in predicting finger-level forces, albeit with incremental degradation from the thumb towards the little finger.

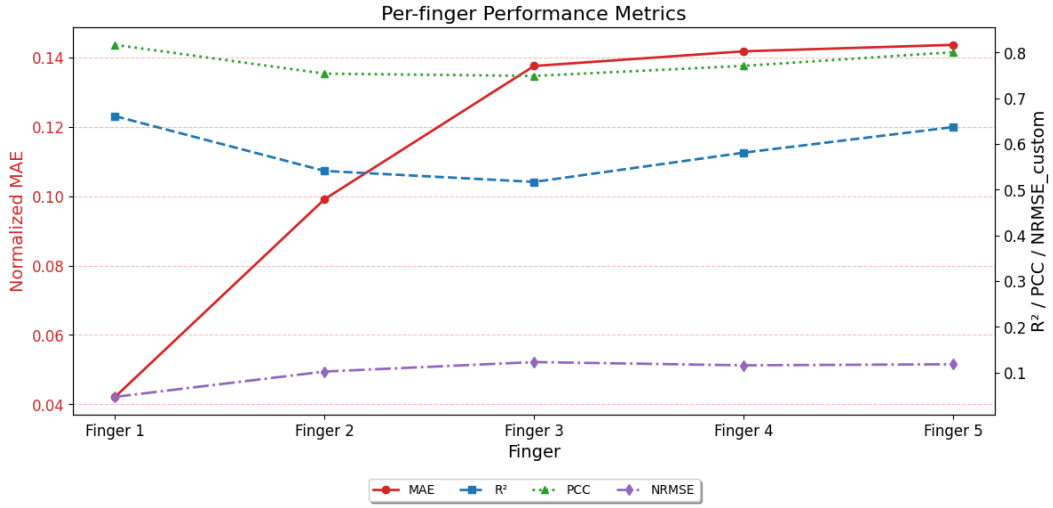


Figure 5: Per-finger performance metrics of the model for finger force prediction. The plot displays Normalized Mean Absolute Error (MAE) on the primary (left) y-axis, and R^2 score, Pearson Correlation Coefficient (PCC), and custom Normalized Root Mean Square Error ($NRMSE_{\text{custom}}$) on the secondary (right) y-axis, for each of the five fingers.

A.4.7 Qualitative Examples of Force Prediction

The Figure 6 presents a qualitative comparison between predicted and ground-truth finger forces for Finger 1 across a representative time segment. The predicted force closely tracks the general trend and dynamic variations of the ground truth, effectively capturing both the onset and offset phases of applied pressure. Although minor deviations are observable at peak forces, the model successfully reflects the temporal structure of finger pressure, underscoring its effectiveness in realistic force estimation scenarios.

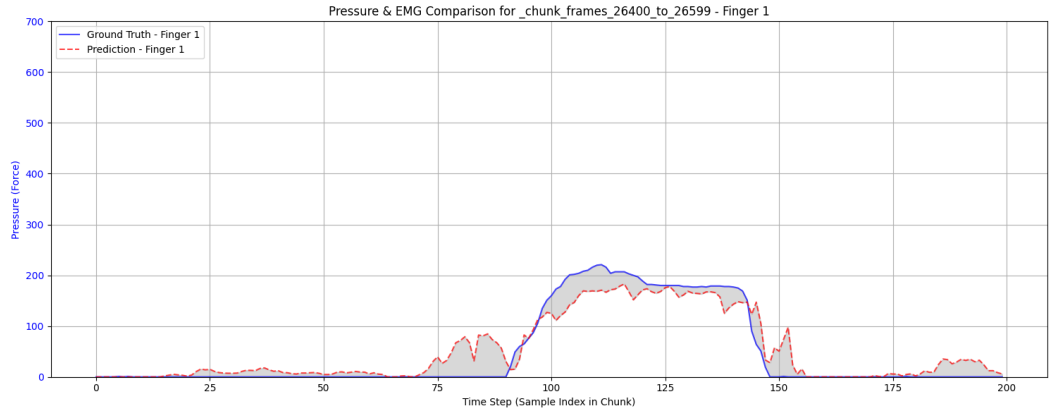


Figure 6: Example of predicted versus ground truth pressure for Finger 1 over a specific time window (frames 26400-26599). The ground truth is shown as a blue solid line, the model's prediction as a red dashed line, and the shaded area indicates the magnitude of the prediction error.

A.4.8 Visualization of Learned Synergies/Weights

Figure 7 shows the learned weight matrix W_{sparse} from the Sparse Gated Feedforward (SGF) module, which maps context-modulated kinematic features (H_{CBN}) to per-finger base force predictions (P_{base}). The matrix is notably sparse—many weights are near-zero (green)—as encouraged by L1 regular-

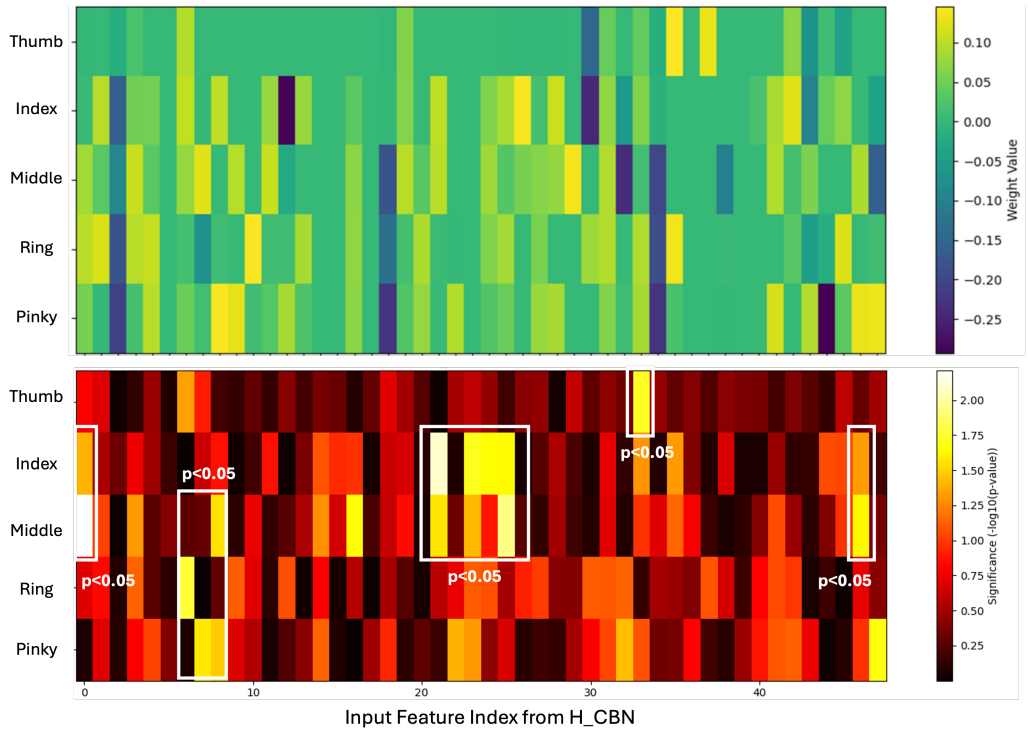


Figure 7: Analysis of the learned sparse weight matrix W_{sparse}. (Top) Mean weight values across all cross-validation folds. (Bottom) Statistical significance of each weight, visualized as -log10(p-value) from a one-sample t-test.

ization. This sparsity reveals that the model avoids dense, all-to-all mappings and instead identifies finger-specific subsets of relevant input features.

Such structure enables efficient and interpretable computation, consistent with sparse coding observed in biological motor circuits. For example, Finger 2 exhibits strong negative weights at feature indices 12–13 and strong positive weights at 29–30 and 43–44, indicating distinct, opposing influences. Meanwhile, features 32–34 negatively influence Fingers 1, 3, and 4, but have little effect on others. These patterns suggest that H_{CBN} encodes higher-level kinematic or intent features that coordinate multi-finger force outputs in a synergy-like fashion.

Overall, the interpretable and disentangled structure learned by W_{sparse} demonstrates the benefit of incorporating neuro-inspired structural inductive biases. It improves generalization from sparse data and offers transparency crucial for reliable human-AI interaction.

A.5 Broader Impact Statement

Our work advances the field of force-aware human-computer interaction by enabling accurate finger-level force estimation from sparse, wearable biosignals. This has the potential to significantly reduce the cost, intrusiveness, and power consumption of systems for dexterous input in domains such as virtual reality, assistive robotics, prosthetics, and ergonomic monitoring. By mimicking structural principles from biological motor control, our approach not only improves data efficiency and generalization but also enhances model interpretability—an important step toward trustworthy AI in embodied applications.

In the assistive technology space, our system could empower individuals with motor impairments by enabling intuitive control of prosthetic limbs or exoskeletons with minimal instrumentation. Similarly, in industrial and medical settings, the ability to monitor hand force non-invasively could help prevent repetitive strain injuries or enable remote physical therapy.

However, as with any sensing system that interfaces closely with the human body, there are potential risks related to privacy and fairness. For instance, wearable biosignal data could unintentionally reveal

sensitive health conditions if not properly safeguarded. Moreover, although our model demonstrates strong generalization, it may still underperform for users whose neuromuscular characteristics are underrepresented in the training data. Addressing this will require diverse participant recruitment and careful evaluation across demographics.

Finally, our work does not support applications that attempt to infer intent or behavior covertly, or in ways that violate user consent. We strongly advocate for transparent use, informed consent, and user control in all deployments of human sensing technologies.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 1 includes the main claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 includes the limitations of the work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and the appendix include all the information needed to reproduce the main experimental results of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is available at `https://github.com/3525starmoon-glitch/NiSM`, and the data are available upon request (see the appendix).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and the appendix include all the training and test details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 includes the main quantitative result with error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix includes information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed NeurIPS Code Of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The appendix includes societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not include data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We checked the licenses of assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The licenses and data-use terms of the released code and data are stated in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Section 4.1 and the appendix describe the study procedure, the tasks performed by participants, and the informed-consent process.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: Section 4.1 and the appendix includes the text on IRB approvals. Human study in this work does not give rise to clinical risks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for editing and formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.