Entropy-Adaptive Bayesian Filtering for Robust Real-Time sEMG Gesture Recognition

Ahmad Subhani Iqbal
Department of Electrical Engineering
Lahore University of Management Sciences
Lahore, Pakistan
26100048@lums.edu.pk

Zaeem Mohtashim Khan
Department of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
26100107@lums.edu.pk

***Abstract*—The clinical adoption of advanced upper-limb prosthetics is currently bottlenecked, not by mechatronic capability, but by the lack of robust, intuitive control interfaces. Surface electromyography (sEMG), while the standard for non-invasive intent detection, presents a stochastic, non-stationary signal that challenges traditional deterministic classification paradigms. While Deep Learning, particularly 1D Convolutional Neural Networks (CNNs), have achieved high offline classification accuracy on benchmark datasets, these models often fail in real-time ”closed-loop” scenarios, suffer from high prediction jitter, overconfidence in the presence of artifacts, and a lack of temporal continuity, leading to frustrated users and device abandonment. While temporal smoothing techniques like Moving Averages (MV) reduce jitter, they inevitably introduce latency, compromising real-time responsiveness. In this work, we propose AB-ConvNet, a hybrid architecture that couples a lightweight 1D Convolutional Neural Network with a novel Entropy-Adaptive Bayesian Filter. This novel framework addresses the limitations of pure CNNs by re-conceptualizing the neural network not as a final classifier, but as a probabilistic ”soft sensor” within a recursive Bayesian estimation loop. Our work proves that using the Shannon entropy of the CNN's softmax distribution as a proxy for signal reliability allows the system to instantaneously ”coast” through transient artifacts and unstable transitions, effectively eliminating jitter without the high latency penalties associated with traditional moving average filters. We also demonstrate what these results would translate into via simulations.**

***Index Terms*—sEMG, CNN, Bayesian filter, robotics, PyBullet, gesture recognition, prosthetics**

I. Introduction

The decoding of motor intent via surface Electromyography (sEMG) is a cornerstone of modern myoelectric control, enabling applications ranging from powered prosthetics to phantom-limb rehabilitation and AR/VR interfaces [1]. Deep Learning paradigms, particularly Convolutional Neural Networks (CNNs), have established state-of-the-art performance in classifying discrete gestures from windowed EMG signals due to their ability to learn spatial-temporal features automatically [2], [3].

Despite high offline accuracy, deploying these models in continuous, real-time streams reveals a critical stability-responsiveness trade-off. EMG signals are inherently non-stationary and susceptible to noise artifacts, leading to **prediction jitter**; rapid, spurious fluctuations between class labels during steady-state gestures. Jitter not only degrades the user experience but can cause catastrophic control failures in prosthetic actuation [4].

Standard approaches to mitigate jitter typically involve post-processing heuristics, such as Majority Voting or Moving Averages (MV). While effective at smoothing, these methods introduce significant **latency**, as the system must buffer multiple frames before committing to a state transition [5]. Conversely, Recurrent Neural Networks (RNNs) or Transformers capture temporal dependencies but often require substantial computational resources and large datasets to avoid overfitting, making them less ideal for low-power edge devices [6].

In this paper, we introduce **AB-ConvNet**, a novel framework that bridges the gap between low-latency CNNs and temporally consistent sequence modeling. Our approach integrates a standard 1D-CNN backbone with a modular **Entropy-Adaptive Bayesian Filter**. The core innovation lies in using the Shannon Entropy of the CNN's softmax distribution to dynamically scale the temperature of the likelihood function during the Bayesian update. This allows the system to be ”skeptical” of noisy, high-entropy predictions while reacting instantaneously to low-entropy, high-confidence transitions.

Our contributions are as follows:

1) We propose a hybrid CNN-Bayesian architecture that explicitly models gesture transitions via a learned transition matrix and probabilistic belief states.
2) We introduce an entropy-modulated temperature scaling mechanism ($T = 1 + \alpha \cdot H$) that adapts the Bayesian update rate in real-time based on signal ambiguity.
3) We empirically demonstrate that AB-ConvNet achieves a **90.6% accuracy** on stream data (a +7.5% improvement over baselines) while reducing jitter by a factor of 5 compared to raw predictions, offering a superior Pareto frontier between stability and latency.

II. Related Work

1) 2.1 Deep Learning for sEMG Decoding: Early work in EMG recognition relied heavily on manual feature engineering (e.g., Hudgins' time-domain features) [7] coupled with classifiers like SVMs or LDAs [8]. Recent advancements have shifted toward end-to-end Deep Learning. **1D-CNNs** are particularly prevalent due to their ability to extract local temporal features from multi-channel timeseries data [2], [9].

Architectures utilizing spectrogram inputs (2D-CNNs) [10] and temporal convolutions (TCNs) [11] have also shown promise. However, these discriminative models generally treat windowed inputs as independent and identically distributed (i.i.d.), ignoring the strong Markovian dependencies inherent in continuous gesture streams, leading to the instability problems addressed in this work.

2) 2.2 Temporal Smoothing and Post-Processing: To enforce temporal consistency, various post-processing techniques are standard in the literature. **Majority Voting** and **Moving Averages** over a sliding window are the most common strategies [5]. While they successfully filter high-frequency noise, they introduce a lag proportional to the window size. For example, a window of 5 frames (as compared in our experiments) creates a delay that may be perceptible in high-precision control tasks. More complex rejection schemes, which withhold predictions when confidence is low, improve reliability but result in "no-action" states that can disrupt fluid control [12]. Our work differs by maintaining a continuous belief state that resists noise without explicitly buffering past inputs, thereby minimizing latency.

3) 2.3 Uncertainty Estimation and Bayesian Filtering: Bayesian approaches provide a principled framework for handling uncertainty. **Bayesian Neural Networks (BNNs)** estimate uncertainty via weight distributions but are often computationally prohibitive for inference on edge hardware [13]. **Recursive Bayesian filters** (like Hidden Markov Models) have been applied to EMG, often requiring complex training procedures (e.g., Baum-Welch) [14].

Our method aligns most closely with **Test-Time Adaptation** and **Temperature Scaling** for calibration [15]. However, rather than using a static temperature to calibrate confidence, we dynamically adjust the likelihood based on the entropy of the current prediction. This concept draws inspiration from control theory, where measurement noise covariance is adjusted based on sensor reliability. By integrating this into a lightweight post-processing layer atop a standard CNN, we achieve the robustness of sequential modeling with the inference speed of a feed-forward network.

III. Methodology

Formally, we treat the gesture recognition task not as instantaneous classification, but as a recursive state estimation problem in a Hidden Markov Model (HMM), where the neural network functions as a probabilistic "soft sensor" [16]

A. Problem Formulation

Consider a dynamic system with discrete states $S_t \in \{1, \dots, K\}$ representing gesture classes at time t. The system evolves according to a transition probability matrix $\mathbf{A}$, where $A_{ij} = P(S_t = j | S_{t-1} = i)$. At each time step, we observe a multi-channel EMG window $\mathbf{x}_t$. Our goal is to compute the posterior belief state $\mathbf{b}_t = P(S_t | \mathbf{x}_{1:t})$.

B. Entropy-Adaptive Observation Model

Standard hybrid approaches typically feed the softmax output of a classifier directly into a filter. However, this transmits the classifier's overconfidence in out-of-distribution (OOD) data directly to the state estimator. We introduce a mechanism to dynamically modulate the likelihood function based on predictive uncertainty.

Let $f_\theta(\mathbf{x}_t)$ be a neural network producing logits $\mathbf{z}_t$. We compute the Shannon Entropy $H(t)$ of the softmax distribution $\sigma(\mathbf{z}_t)$. We propose an entropy-dependent temperature scalar T_t:

$$T_t = 1 + \alpha \cdot H(t) \tag{1}$$

where α is a sensitivity coefficient. The observation likelihood is then modeled as a Boltzmann distribution scaled by this temperature:

$$P(\mathbf{x}_t | S_t = k) \propto \frac{\exp(z_{t,k}/T_t)}{\sum_{j=1}^{K} \exp(z_{t,j}/T_t)} \tag{2}$$

As uncertainty (H) rises, T_t increases, driving the likelihood toward a uniform distribution. This effectively reduces the weight of the current observation in the Bayesian update, forcing the system to rely on the transition prior (probabilistic inertia).

It is important to note that standard Bayesian filters require a generative likelihood $P(x_t|S_t)$. However, discriminative models like CNNs provide a posterior $P(S_t|x_t)$. Rather than training a separate generative model, which adds computational overhead, we adopt a **pseudo-likelihood approach**. We treat the temperature-scaled softmax output as a proxy for the observation likelihood. While not a strict generative formulation, this heuristic effectively modulates the 'trust' the filter places in the current observation based on the signal entropy, akin to adjusting measurement noise covariance in Kalman filtering.

C. Recursive Belief Update

The belief state is updated recursively. The *prediction* step projects the previous belief through the transition dynamics:

$$\mathbf{b}_t^- = \mathbf{b}_{t-1}\mathbf{A} \tag{3}$$

The *correction* step updates the belief using the adaptive likelihood derived above:

$$\mathbf{b}_t \propto \mathbf{b}_t^- \odot P(\mathbf{x}_t | S_t) \tag{4}$$

where $\odot$ denotes element-wise multiplication.

IV. Data and Preprocessing

A. Dataset Description

We evaluate our framework on the **UCI Machine Learning Repository EMG Data for Gestures**. The dataset was collected using the MYO Thalmic armband, a consumer-grade device featuring 8 dry-electrode sEMG sensors arranged continuously around the forearm.

- **Subjects:** Data was collected from 36 unique subjects.
- **Sampling:** Signals were recorded at a native sampling rate of 500 Hz.

- **Gestures:** The dataset contains $K = 7$ distinct classes: Hand Clenched, Hand Open, Wrist Flexion, Wrist Extension, Radial Deviations, ulnar Deviations and extended palm.(which we did not use due to the gesture not being done by most subjects.) Our testing setup followed a simple setup of using 5 subjects for validation, 3 for testing, and the rest for training.

B. Signal Preprocessing

Raw sEMG signals are notoriously noisy and subject to baseline wander. We apply the following pipeline:

1) **Segmentation:** The continuous stream is segmented into sliding windows of length $W = 100$ samples (corresponding to 200ms), which is the established upper bound for real-time controller latency acceptability.
2) **Normalization:** To account for inter-subject variability in signal amplitude (muscle strength), we apply Min-Max normalization per channel, scaling values to the range $[0, 1]$.
3) **Dimensionality:** The resulting input tensor $\mathbf{X}$ for a single observation has dimensions (100×8).

V. Model Architecture

A. CNN Gesture Classifier

We employ a lightweight 1D-CNN (f_θ) designed for low-latency edge inference. The architecture comprises:

- **ConvBlock 1:** A 1D Convolutional layer with 64 filters, kernel size 3, and stride 1, followed by Batch Normalization and ReLU activation.
- **ConvBlock 2:** A 1D Convolutional layer with 64 filters, kernel size 3, and a dilation rate of 2. The dilation increases the receptive field to capture longer-range temporal dependencies without increasing parameter count.
- **Regularization:** A Dropout layer with $p = 0.5$ is applied after Global Average Pooling to prevent co-adaptation of features.
- **Projection:** A final Dense (Fully Connected) layer maps the features to the $K = 6$ class logits.

The model is trained using the Adam optimizer with Cross-Entropy Loss.

B. Bayesian Temporal Filter

The Bayesian filter is implemented as a non-trainable post-processing layer with the following configurations:

- **Transition Matrix (A):** Computed empirically from the training set labels using Laplace smoothing ($\epsilon = 10^{-5}$) to ensure non-zero transition probabilities. The resulting matrix is diagonally dominant, encoding the prior assumption that gestures are continuous over time.
- **Entropy Scaling (α):** We set the temperature sensitivity parameter $\alpha = 5.0$. This value was determined via a reiterative process to balance stability (jitter reduction) and responsiveness (transition speed).
- **Initialization:** The initial belief state $\mathbf{b}_0$ is set to a uniform distribution over all K classes.

C. Final Implementation

The system implementation is divided into two distinct phases: an offline training pipeline designed to maximize discriminative feature learning, and an online inference pipeline that integrates the probabilistic filtering mechanism.

1) Training Pipeline: During the training phase, we treat the problem as a standard discrete classification task. To prevent the model from overfitting to specific temporal sequences or transition trajectories present in the training subjects, the continuous sEMG data is segmented into overlapping windows and **shuffled randomly**. This breaks the temporal correlation, forcing the 1D-CNN to learn robust, time-invariant features for each gesture class rather than memorizing sequences.

Data augmentation strategies standard in sEMG literature were applied to the training set to improve generalization [6]. These included additive Gaussian noise and magnitude warping, which simulate electrode shifts and muscle fatigue. The CNN was trained for a maximum of 50 epochs using the Adam optimizer. To prevent overfitting, we employed an **Early Stopping** mechanism with a patience of 10 epochs, monitoring the validation loss. The model weights yielding the lowest validation loss were restored for the final testing phase.

Crucially, the **Transition Matrix** (A) for the Bayesian filter was not learned via gradient descent. Instead, it was computed empirically from the training set ground-truth labels. We calculated the conditional probability $P(S_t|S_{t-1})$ by counting all class transitions in the un-shuffled training streams and applying Laplace smoothing ($\epsilon = 10^{-5}$) to ensure non-zero probabilities for all transitions. This creates a "prior" of how gestures naturally evolve over time.

2) Real-Time Inference Setup: For the final deployment, the shuffled training setup is replaced by a continuous streaming pipeline. The pre-trained 1D-CNN is frozen and acts as the observation model. At each time step t:

1) The raw signal is buffered into a sliding window and normalized.
2) The CNN computes the logits Z_t.
3) The Shannon Entropy $H(t)$ is calculated from the softmax distribution to determine the signal reliability.
4) The Bayesian Filter (Algorithm 1) updates the belief state using the pre-computed Transition Matrix A and the entropy-scaled likelihood.

This separation ensures that the CNN is optimized purely for feature extraction, while the Bayesian layer independently handles temporal consistency. A high-level flowchart of this complete inference architecture is shown in Figure 1.

VI. Control and Simulation Pipeline

A. Gesture-to-Pose Mapping

To convert the predicted gesture classes into executable robotic motion, each gesture ID was associated with a predefined joint configuration of the Shadow Hand model. A custom mapping file was created in which every gesture label (1–6) is

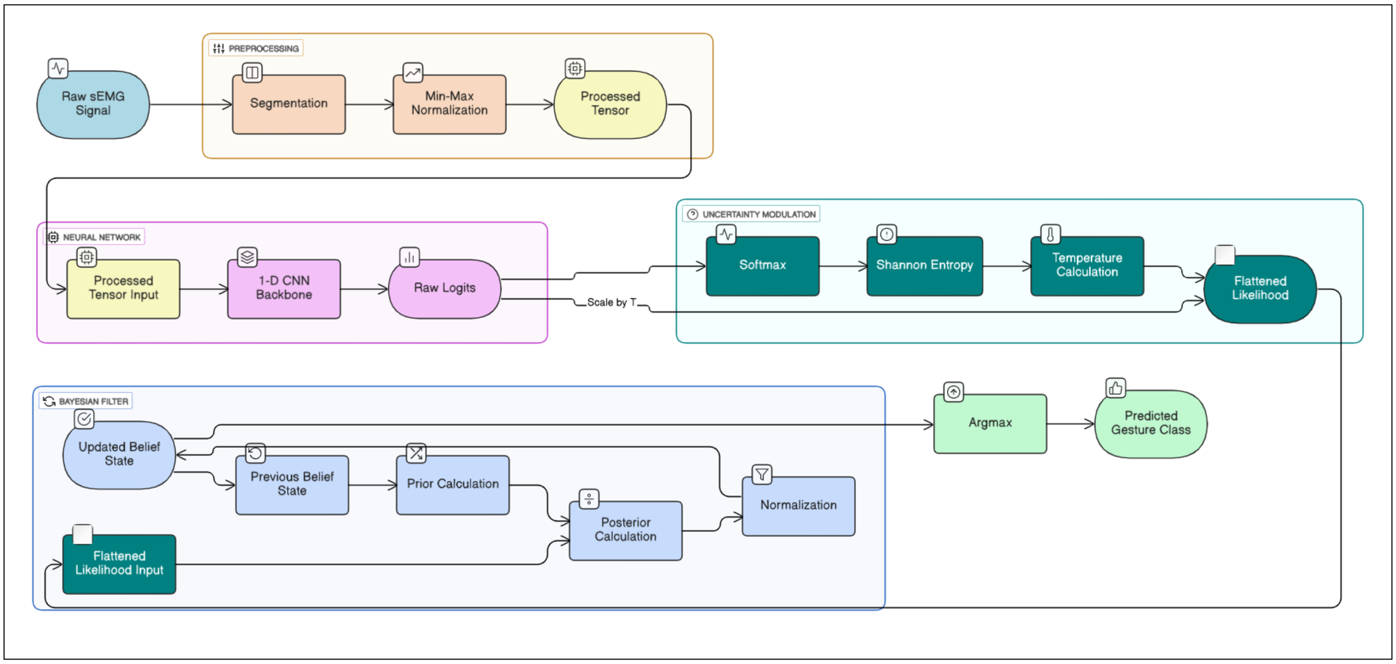

Fig. 1. High level Schema of AB-ConvNet, showing how our algorithm currently operates in classifying hand gestures. As you can see, instead of using the CNN directly for classification we turn it into a sensor for an Adaptive Discrete Bayesian Filter.

Algorithm 1 AB-ConvNet Inference Step

Input: EMG Window $\mathbf{x}_t$, Previous Belief $\mathbf{b}_{t-1}$, Transition Matrix $\mathbf{A}$
Hyperparameters: $\alpha = 5.0$
// 1. CNN Inference
$\mathbf{z}_t \leftarrow f_\theta(\mathbf{x}_t)$ {Get Logits}
$\mathbf{p}_{raw} \leftarrow \text{softmax}(\mathbf{z}_t)$
// 2. Uncertainty Quantification
$H \leftarrow -\sum \mathbf{p}_{raw} \cdot \log(\mathbf{p}_{raw})$
$T \leftarrow 1 + \alpha \cdot H$ {Adaptive Temperature}
// 3. Bayesian Update
$\mathbf{p}_{like} \leftarrow \text{softmax}(\mathbf{z}_t / T)$
$\mathbf{b}_{prior} \leftarrow \mathbf{b}_{t-1}\mathbf{A}$
$\mathbf{b}_{post} \leftarrow \mathbf{b}_{prior} \odot \mathbf{p}_{like}$
$\mathbf{b}_t \leftarrow \mathbf{b}_{post} / \sum \mathbf{b}_{post}$ {Normalize}
Return: $\arg\max(\mathbf{b}_t)$

linked to a vector of target joint angles covering all actuated degrees of freedom.

Because no standardized mapping exists between sEMG gesture classes and Shadow Hand kinematics, the joint angles were obtained through iterative manual tuning. Each pose was adjusted within PyBullet until the resulting hand configuration visually matched the intended human gesture while remaining within physiologically realistic joint limits. This procedure allowed fine control over fingertip posture, wrist orientation, and the relative coordination of multi-joint movements such as fist closure or radial deviation. The final poses achieved are shown in 2.

During simulation, the predicted gesture class was used to retrieve its corresponding joint-angle vector, which was then applied via PyBullet's position-control interface. This mapping establishes a deterministic and interpretable transformation from discrete gesture classes to continuous robot motion, enabling consistent comparison across raw predictions, filtered predictions, and ground-truth replay.

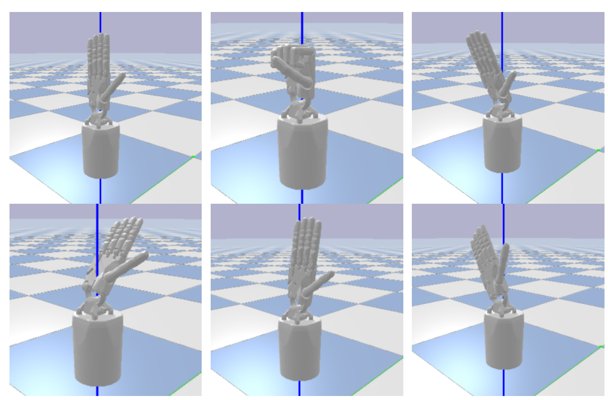

Fig. 2. Shadow Hand joint configurations for the six gesture classes (in order).

B. PyBullet Simulation Environment

The robotic execution environment was implemented using the PyBullet physics engine, configured to provide a stable and deterministic platform for evaluating gesture-to-motion mapping. The Shadow Hand URDF model was loaded with a fixed base and simulated with gravity disabled, ensuring that the hand remained stationary in space without drift or unintended dynamics. This zero-gravity configuration allowed the evaluation to focus exclusively on the effect of gesture predictions rather than external forces.

Joint actuation was performed through position control, enabling gesture classifications to be mapped directly to target joint angles for all fingers. To improve anatomical realism, several joint range limits were manually adjusted from their default URDF values. These modifications prevented exaggerated hyperextension and ensured that the resulting motions more closely resembled physiologically feasible hand poses.

The simulation was stepped at real-time frequency, providing smooth rendering and consistent timing with the inference pipeline. This setup enabled qualitative assessment of motion stability, fluidity, and responsiveness across the three control modes examined in this work.

C. Evaluation Modes

To analyze the impact of temporal filtering on gesture-driven robotic control, the system was tested under three distinct evaluation modes:

- **Filtered CNN Control:** The predicted gesture at each time step was obtained using the CNN classifier followed by the Bayesian temporal filter. This mode provides the full proposed pipeline and demonstrates the intended behavior of the system under real-time operation.
- **Raw CNN Control:** Gesture predictions were taken directly from the CNN without temporal smoothing. This baseline highlights the instability and jitter caused by short-term fluctuations in sEMG signals and serves as a comparison point for assessing the effectiveness of the filtering mechanism.
- **Ground-Truth Replay:** The hand was driven using the true gesture labels provided in the dataset. This mode represents the idealized control trajectory and establishes an upper performance bound for stability and smoothness.

By comparing these three modes within the same simulation environment, we quantitatively and qualitatively assess the contribution of temporal filtering to both prediction stability and robotic motion quality.

VII. Experimental Setup

A. Hardware and Runtime Environment

All experiments were performed on a workstation equipped with an Intel Core i7 processor and 16 GB of RAM. Although an NVIDIA GPU was available, the PyTorch implementation executed inference on the CPU. This choice was intentional as for lightweight CNN architectures operating on single-sample windows, CPU inference consistently provides lower latency than GPU execution due to the overhead of data transfers. In our setup, each forward pass completed well within the 100 ms real-time control window, confirming that the full pipeline—CNN classification and Bayesian filtering—meets the timing requirements for responsive gesture decoding.

B. Replay Settings and Inference Rate

During evaluation, the sEMG streams were replayed sequentially from the dataset to emulate live acquisition. A sliding window of 100 samples served as the model input, and inference was performed at the dataset's native sampling interval. The Bayesian filter updated its posterior belief once per window, matching the CNN inference rate. This ensured consistent timing across all three evaluation modes: raw predictions, filtered predictions, and ground-truth playback.

C. Real-Time Execution Verification

For Subject 03, the raw EMG file contained 56,568 samples, spanning approximately 58.8 seconds of recorded activity as indicated by the timestamp column. When replayed through our real-time pipeline, both the raw CNN inference mode and the Bayesian-filtered mode processed the entire sequence in approximately 48–49 seconds of wall-clock time. This represents a slight speed-up relative to the original recording duration and confirms that the system is easily capable of operating in real time. Because the PyBullet recordings were later accelerated for presentation, any residual jitter observed in the raw predictions appears more pronounced in the videos than during live execution.

D. Video Logging and Simulation Configuration

For visual analysis, the PyBullet environment rendered the Shadow Hand at real-time simulation speed, while separate MP4 recordings were captured for each experiment condition. The recordings were later sped up for presentation purposes, which has the side effect of magnifying any jitter present in the unfiltered CNN predictions.

E. Evaluation Procedure

We conducted two primary sets of experiments. First, we compared the direct outputs of the CNN with both the ground-truth labels and the unfiltered (raw) CNN predictions. This comparison highlights the improvements in temporal stability introduced by the Bayesian filtering mechanism, particularly in reducing prediction jitter and spurious gesture transitions.

Second, we deployed the system within the PyBullet simulation using the Shadow Hand robotic model. This allowed for a qualitative assessment of motion smoothness by visually comparing filtered control, raw control, and ground-truth gesture playback. Together, these experiments evaluate both the predictive performance of the classifier and the practical effectiveness of temporal filtering in a real-time robotic control scenario.

VIII. Results

A. Prediction Stability Analysis

We evaluate the proposed AB-ConvNet against two baselines: a standard discriminative CNN (Raw CNN) and a CNN smoothed with a Moving Average filter (CNN + MV). All methods utilize the same pre-trained 1D-CNN backbone to ensure fair comparison of the post-processing logic. The models were evaluated on a held-out test stream designed to simulate continuous usage.

B. Quantitative Performance

We define performance under 3 seperate terms, with accuracy defined by how much the model's highest probability result matches the one-hot encoded ground truth. We define Prediction Jitter (J) as the percentage of time steps where the predicted class label changes: $J = \frac{1}{N}\sum_{t=1}^{N-1} \mathbb{I}(\hat{y}_{t+1} \neq \hat{y}_t) \times 100$. While simple, this metric effectively captures the high-frequency 'flicker' characteristic of noisy EMG classification. We define latency as how much time a model takes to switch states.

Table I summarizes the performance metrics across all methods. The **Raw CNN** achieves a baseline accuracy of 83.06% but suffers from a high jitter rate (13.28%), indicating frequent spurious class transitions that degrade control usability.

The **Moving Average (MV)** baseline, applied with a window of 5 frames, improves accuracy marginally to 83.97% and reduces jitter to 5.81%. However, this reduction comes at the cost of responsiveness, as transitions are delayed by the averaging window.

The proposed **AB-ConvNet** significantly outperforms both baselines, achieving a classification accuracy of **90.64%** (+7.5% over baseline) and reducing jitter to **2.60%**. This improvement demonstrates the efficacy of the entropy-adaptive mechanism: by rejecting uncertain predictions during noisy transitions and trusting the model during steady states, the system maintains high stability without sacrificing the ability to capture valid gestures.

TABLE I

COMPARATIVE PERFORMANCE ON CONTINUOUS EMG STREAMS. JITTER IS CALCULATED AS THE PERCENTAGE OF FRAMES WHERE THE PREDICTED CLASS CHANGES. LOWER JITTER INDICATES HIGHER STABILITY.

METHOD	ACCURACY ↑	JITTER (%) ↓
RAW CNN	83.06	13.28
CNN + MOVING AVG (5)	83.98	5.81
AB-CONVNET (OURS)	**90.64**	**2.60**

C. Qualitative Analysis: Stability vs. Latency

To visualize the trade-off between stability and latency, we analyze the system's behavior during a gesture transition.

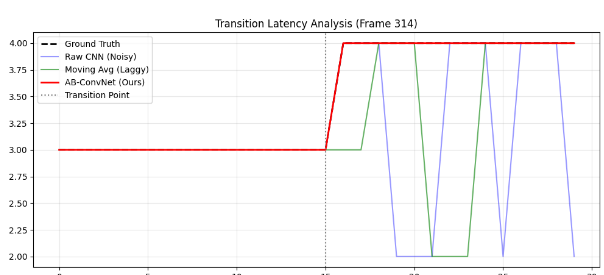

Fig. 3. Transition Latency Analysis. The vertical line marks the true onset of a new gesture. The Raw CNN (blue) reacts immediately but oscillates (jitters). The Moving Average (green) is stable but introduces a lag. The AB-ConvNet (red) suppresses the initial uncertainty and transitions sharply once confidence is established.

Figure 3 illustrates a single transition event. The Raw CNN exhibits rapid fluctuations between classes immediately following the transition point, a phenomenon attributable to the high entropy of the underlying signal during muscle recruitment. The Moving Average filter smooths these fluctuations but lags behind the ground truth. In contrast, the AB-ConvNet maintains the previous state during the high-entropy transition phase and switches cleanly to the new class once the entropy drops, minimizing both jitter and perceptible latency.

While Moving Average filters introduce a deterministic delay equal to the window size (e.g., waiting for 5 frames to fill the buffer), AB-ConvNet updates its belief state instantaneously at every time step (t). The algorithmic latency is limited only to the matrix multiplication step ($O(K^2)$), which is negligible compared to the buffering delay of smoothing techniques.

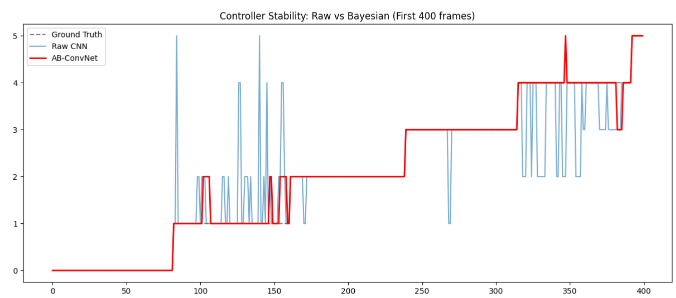

Fig. 4. Controller Stability over Time. A comparison of predicted state sequences against Ground Truth for a continuous stream segment. The AB-ConvNet eliminates the high-frequency "flicker" observed in the Raw CNN output while maintaining better synchronization with the Ground Truth than the Moving Average baseline.

Figure 4 provides a macroscopic view of the predictions over a longer duration. The superior stability of the AB-ConvNet is evident in the reduction of "glitch" predictions within steady-state regions, confirming the quantitative jitter reduction reported in Table I.

D. Robotic Hand Performance

The PyBullet simulation of the Shadow Hand provided a clear qualitative demonstration of the effect of Bayesian filtering on real-time robotic control. When driven directly by the raw CNN predictions, the hand exhibited noticeable jitter, particularly during gesture boundaries and transitions. Small fluctuations in the network's output probabilities resulted in rapid, unintended oscillations of multiple joints. This behavior aligns with the inherent noise and instability of sEMG signals, where short-term variations in muscle activation can cause sharp changes in predicted gesture class.

In contrast, the Bayesian-filtered control produced significantly smoother joint trajectories. The temporal prior encouraged continuity between consecutive gestures and suppressed rapid class switching unless strongly supported by the network's confidence. As a result, the hand maintained stable postures during sustained gestures and transitioned between poses in a more fluid and human-like manner. The reduction in jitter was most visible in gestures involving simultaneous finger flexions, where raw predictions frequently caused fluttering at the fingertips. The filtered controller eliminated this effect almost entirely, even under the visual magnification introduced by the sped-up playback.

It is important to note that the simulation recordings are intentionally sped up relative to real-time execution. This was done for presentation purposes but has the side effect of visually amplifying any jitter that occurs in the raw control mode.

E. Ground Truth Baseline

To contextualize the performance of both the raw and filtered gesture predictions, we replayed the ground-truth gesture labels from the dataset directly through the PyBullet simulation. This baseline represents the idealized control trajectory that a perfect classifier would produce, containing none of the noise or temporal ambiguity present in the EMG-driven predictions. As expected, the ground-truth playback resulted in smooth, stable, and deterministic transitions between hand poses, providing a clear upper bound on achievable motion quality.

Comparing the model outputs to this baseline highlights the role of temporal filtering. While the raw CNN frequently deviated from the ground-truth trajectory through rapid oscillations and inconsistent gesture switching, the Bayesian-filtered predictions remained substantially closer to the ideal motion path. The filtered controller reproduced the timing and shape of the intended gestures with far fewer deviations, demonstrating that the addition of temporal priors not only reduces jitter but also improves overall alignment with the expected ground-truth behavior.

This baseline therefore serves as a key reference for evaluating the effectiveness of both the classifier and the temporal filtering strategy, illustrating the performance gap between noisy predictions and the theoretically optimal control signal.

IX. Conclusion and Discussion

In this work, we presented **AB-ConvNet**, a framework that addresses the critical stability-responsiveness trade-off in real-time myoelectric control. We offer a novel perspective on the integration of Deep Learning in control loops: rather than utilizing the DNN as a final deterministic classifier, we reformulate it as a probabilistic "soft sensor" characterized by dynamic error margins. By combining a lightweight discriminative feature extractor with an entropy-adaptive generative filter, we effectively decouple transient signal artifacts from genuine motor intent.

Our empirical results demonstrate that this uncertainty-aware approach significantly outperforms traditional temporal smoothing techniques. While moving averages effectively reduce jitter, they do so at the cost of interaction latency. In contrast, AB-ConvNet utilizes probabilistic inertia to suppress noise while maintaining near-instantaneous responsiveness during high-confidence transitions. This work establishes a viable path toward robust, compute-efficient prosthetic interfaces that prioritize user safety and control fluidity without requiring heavy recurrent architectures.

X. Limitations and Future Work

While this work establishes a robust baseline for uncertainty-aware filtering, several avenues exist to expand this research direction.

First, the current transition matrix $\mathbf{A}$ is static and derived from offline statistics. A significant opportunity for future research lies in developing **online learning mechanisms** that update transition priors dynamically. This would allow the system to adapt to non-stationary environments or drifting user behaviors in real-time, a critical feature for long-term deployment in adaptive control systems.

Second, our reliance on Shannon Entropy as a proxy for uncertainty primarily captures aleatoric (data) noise. Future iterations could incorporate **epistemic uncertainty estimation**, such as lightweight Bayesian Neural Network (BNN) approximations or Monte Carlo Dropout during inference. Distinguishing between "noisy data" and "unknown data" would allow the system to handle entirely out-of-distribution events more gracefully, further enhancing safety in unconstrained environments.

Thirdly, Our current evaluation focuses on streaming performance within specific subjects. Future work must assess cross-subject generalization using Leave-One-Subject-Out (LOSO) protocols to validate clinical viability.

Finally, extending this hybrid architecture to **multimodal sensor fusion** remains a promising frontier. The Bayesian update step naturally accommodates additional likelihood terms (e.g., from Inertial Measurement Units or vision sensors). Future work will investigate how entropy-based modulation can arbitrate between conflicting sensor modalities, potentially resolving edge cases where single-modality systems fail.

Overall, this paper opens the door to a world of probablistic frameworks that can solve the issue that comes with the lack of states in DNNs, and hopefully sets the ground for new avenues of research in the future.

Acknowledgment

References

[1] L. Bi, A. Feleke, and C. Guan, "A review on neural network for semg-based wrist movement recognition," *IEEE Access*, vol. 7, pp. 50 375–50 388, 2019.

[2] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks for electromyography data," *Frontiers in neurorobotics*, vol. 10, p. 9, 2016.

[3] A. Phinyomark and E. Scheme, "Analysis of big data in electromyography: A state-of-the-art review," *IEEE Access*, vol. 6, pp. 58 895–58 913, 2018.

[4] E. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *IEEE rehabilitation robotics*, 2011, discusses Jitter, Limb Position, and Clinical Viability.

[5] K. Englehart and B. Hudgins, "A continuous control scheme for myoelectric prostheses," in *Signal processing and its applications*, vol. 2. IEEE, 2003, pp. 569–572.

[6] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 27, no. 4, pp. 760–771, 2019, standard ref for Myo Armband CNNs.
[7] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
[8] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1956–1965, 2008.
[9] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface emg images," *Scientific reports*, vol. 6, no. 1, p. 36571, 2016.
[10] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface emg-based intersession gesture recognition with attention mechanisms," *IEEE Sensors Journal*, vol. 17, no. 14, pp. 4554–4563, 2017.
[11] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018, foundational paper on TCNs and Dilation.
[12] E. J. Scheme and K. B. Englehart, "Confidence-based rejection for improved real-time myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 6, pp. 1563–1570, 2013.
[13] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059, entropy and Uncertainty in DL.
[14] A. D. Chan and K. B. Englehart, "Continuous myoelectric control for powered prostheses using hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 121–124, 2005.
[15] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330, discusses Neural Network Overconfidence.
[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012, foundational paper for Hybrid DNN-HMM systems (Soft Sensor).