electronics

MDPI

Article

CNN-Enhanced Transformer with Transfer Learning for Cross-Session Monkey Motor Neural Decoding

Miaoling Wu, Tianlu Gao, Yifei Ren, Yuxin Dai, Shuo Huang and Jun Zhang *

School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China; miaolingwu13@163.com (M.W.); tianlu@whu.edu.cn (T.G.); 2022302191484@whu.edu.cn (Y.R.); yuxindai@whu.edu.cn (Y.D.); shuo.huang@whu.edu.cn (S.H.)
* Correspondence: jun.zhang.ee@whu.edu.cn

Abstract

Cross-session variability in neural signal statistics often leads to distribution shifts that substantially degrade the generalization performance of motor decoding models. To address this challenge, this study proposes a CNN-enhanced Transformer with distribution alignment (CNN-Trans-DA), which integrates local temporal feature extraction with global sequence modeling and a lightweight domain adaptation strategy. A one-dimensional convolutional module is introduced prior to the Transformer encoder to capture local firing patterns, while multi-head attention is used to learn long-range temporal dependencies. For cross-session adaptation, we design a joint distribution alignment method that simultaneously minimizes marginal and conditional distribution discrepancies through maximum mean discrepancy (MMD) and correlation alignment (CORAL). Using four-direction monkey fingertip movement data, the proposed CNN-Trans-DA model achieves an accuracy of 84.64 percent under single-session decoding, outperforming the standard Transformer (81.22%), CNN (80.78%), and BiLSTM (79.29%). In cross-session transfer experiments, the model maintains an 82.42 percent decoding accuracy, showing clear improvements over models without distribution alignment. These results demonstrate that the proposed approach can effectively handle session-induced distribution drift and achieve robust motor intention decoding, providing a scalable and generalizable solution for cross-session brain–machine interface applications.

Keywords: neural decoding; Transformer; Brain–Machine Interface; Statistical Transfer Learning; domain adaptation

1. Introduction

Brain Machine Interfaces (BMIs), which establish a direct communication pathway between the brain and external devices, have become increasingly pivotal in motor function restoration, neurorehabilitation [1], and systems neuroscience [2]. They not only offer individuals with motor impairments the possibility of regaining voluntary control, but also provide an experimental platform for probing the neural mechanisms underlying motor intention [3,4]. In recent years, policy initiatives in the People's Republic of China, such as the "14th Five-Year National Key R&D Program" and the "New Generation Artificial Intelligence Development Plan," have further accelerated the integration of artificial intelligence with neuroscience, positioning BMI as a field of substantial technological and societal relevance [5,6].

Check for updates

Academic Editor: Jichai Jeong

Received: 13 May 2026
Revised: 3 June 2026
Accepted: 5 June 2026
Published: 1 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Among various BMI paradigms, monkey motor decoding tasks are widely used to evaluate the effectiveness and generalizability of decoding algorithms, owing to the close correspondence between non-human primate neural activity patterns and human motor control [7]. By modeling and decoding spiking activity from motor-related cortical regions (e.g., M1 and PMd), researchers can infer the mapping between population neural dynamics and movement intention, thereby providing theoretical and empirical foundations for the development of human BMI systems [8,9].

However, variations in experimental settings, neural states, recording devices, and task conditions often induce changes in the statistical properties of neural signals [10]. Such distributional shifts can markedly degrade model performance and hinder generalization across conditions. Addressing cross-condition domain shifts, therefore, requires effective transfer learning strategies to mitigate discrepancies in neural activity distributions [11,12].

The development of neural decoding methods has progressed from traditional linear models to modern deep learning approaches. Early studies employed linear methods—such as Support Vector Machines and linear regression—due to their simplicity and interpretability [13,14]. Nonlinear models, including neural networks and decision-tree-based models, later demonstrated enhanced flexibility in capturing complex neural relationships. With the rise in deep learning, convolutional neural networks (CNNs), recurrent neural networks, and long short-term memory (LSTM) networks have shown notable success in multichannel neural decoding by automatically extracting discriminative representations from high-dimensional neural data [15,16].

Despite these advances, most models emphasize either spatial or temporal feature extraction. CNNs are effective at capturing local spatial patterns, whereas LSTMs excel at modeling temporal dependencies. The absence of a unified spatiotemporal fusion mechanism limits the ability to fully exploit the coupled spatial coordination and temporal dynamics inherent in neural population activity [17]. Transformer architectures, driven by global attention mechanisms, have recently emerged as compelling alternatives for joint spatiotemporal modeling. Transformers capture long-range temporal dependencies and inter-channel interactions without relying on recurrent structures, aligning well with the coordinated dynamics of neural ensembles. Nevertheless, applications of Transformers to neural decoding remain in an early stage [18]. Their potential with respect to multichannel spike-train modeling, spatiotemporal feature integration, and interpretability has not been systematically explored. In particular, for monkey motor decoding, the theoretical advantages of Transformers have yet to be fully realized due to the lack of structural adaptations tailored to neural data characteristics.

In addition to feature modeling challenges, achieving cross-condition generalization remains a central difficulty. Because the statistical distribution of neural signals strongly depends on experimental conditions and physiological states, differences across recording sessions, task modes, or subjects can introduce substantial distribution shifts and lead to performance degradation. Transfer learning has been widely regarded as an effective means to address such issues by enabling knowledge transfer between source and target domains [19]. Prior studies have demonstrated its potential in cross-subject, cross-task, and pathological neural decoding scenarios [18,19]. However, most existing methods focus primarily on deep feature alignment or shared representation learning, while paying limited attention to discrepancies in the underlying statistical characteristics of neural signals, such as firing rate changes, noise variability, and session-dependent fluctuations [20]. Neglecting these low-level properties undermines model robustness under varying conditions [21,22].

Transfer learning techniques applied in BMI research can generally be categorized into three groups: recalibration-based methods, feature-selection-based methods, and deep-learning-based methods. Recalibration-based approaches aim to reduce inter-domain

discrepancies by adjusting feature distributions between source and target domains [20]. Feature-selection-based approaches improve performance by identifying the most informative features for the target task [21]. Deep-learning-based approaches leverage hierarchical representations to extract shared domain-invariant features, thereby enhancing transferability and robustness [22]. Despite their promise, existing approaches often overlook the statistical-level divergence between domains, which remains a key source of distribution shift and performance degradation in cross-condition decoding.

To address these challenges, this study proposes a CNN-enhanced Transformer architecture with Statistical Transfer Learning for cross-condition decoding of monkey motor neural signals. The method integrates a convolutional front-end with a Transformer encoder, enabling the joint extraction of local spatial features and global temporal dependencies. Furthermore, a statistical distribution alignment strategy is introduced to simultaneously constrain discrepancies in both marginal and conditional distributions between domains, ensuring consistency at both inter-class and intra-class levels. Compared with traditional approaches that consider only a single aspect of distribution mismatch, the proposed strategy enhances generalization and robustness under previously unseen conditions while preserving discriminative capability.

Experimental results on a monkey motor decoding dataset demonstrate that the proposed approach achieves superior cross-condition decoding performance compared with baseline models, including Bi-LSTM, the original Transformer, and standard CNN architectures, validating its effectiveness in modeling complex neural signals under distribution shifts.

The main contributions of this work are as follows: (1) A spatiotemporal neural signal processing framework is proposed to enhance feature representation consistency. (2) A CNN-enhanced Transformer architecture is designed for hierarchical spatiotemporal feature extraction. (3) A statistical transfer learning strategy is introduced to improve cross-condition decoding performance.

The remainder of this paper is organized as follows. Section 2 describes the dataset and preprocessing procedures. Section 3 details the model architecture and transfer learning mechanism. Section 4 presents experimental results and analyses. Section 5 concludes the paper and outlines future directions.

2. Data and Preprocessing

2.1. Dataset Description

This study employed a publicly available neural recording dataset collected from trained rhesus monkeys (*Macaca mulatta*) performing a four-direction fingertip movement task, as shown in Figure 1 [1]. During each trial, the monkey was instructed to move a cursor on a two-dimensional plane from an initial position toward one of four target directions, namely up, down, left, and right (Figure 2). Multichannel spike activity and the corresponding fingertip position trajectories were simultaneously recorded during each valid movement event. In this study, each complete movement event was treated as an independent sample, and a total of 1893 valid samples were retained after preprocessing.

The dataset was not collected by the authors of this study. Therefore, no new animal experiments were conducted in the present work. The original animal experimental procedures and ethical approvals were provided by the source study/dataset. This study only performed secondary analysis of the recorded neural signals for algorithm development and validation.

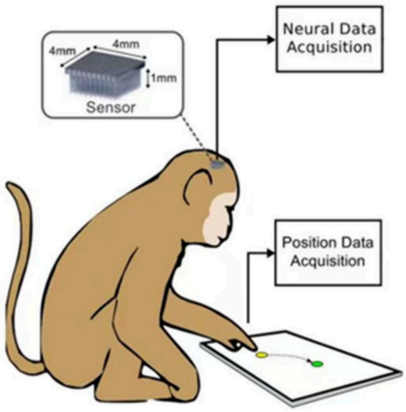


Figure 1. Schematic of the experimental paradigm.

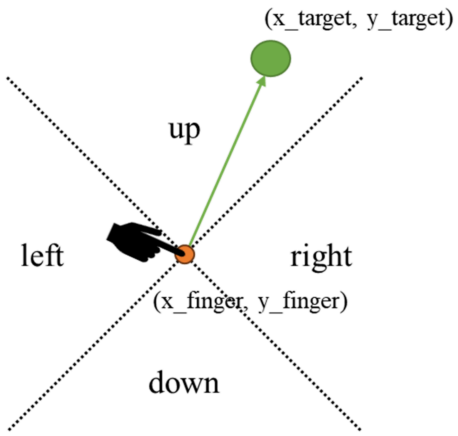


Figure 2. Illustration of cursor movement trajectories.

2.2. Sample Definition and Label Construction

Each movement event, defined from the onset at the previous static cursor position to the target position, was treated as an independent sample. Based on task design, movement directions were discretized into four classes. Labels were determined by computing the direction vector from the starting point to the endpoint of each trajectory $\{up, down, left, right\}$. When trajectories showed noise or cross-quadrant drift, labels were assigned according to the principal movement direction (i.e., the direction of maximum projection). To ensure label consistency and reliability, abnormal trials—such as those with excessively short/long movement times or highly irregular trajectories—were excluded.

2.3. Neural Signal Preprocessing Pipeline

To improve the signal-to-noise ratio and obtain a temporally structured representation suitable for model input, neural recordings were processed through the following steps (Figure 3):

(1) Artifact and Noise Removal: Raw signals were examined to remove segments containing overt artifacts (e.g., electrode detachment, transient power-line interference, movement artifacts). Such segments were either marked as missing or excluded from subsequent dataset partitioning.

(2) Spike Event Extraction (Spike Counting): High-resolution spike events were binned using a 40 ms window, yielding spike counts per channel at each time step. This process aggregates the original 4 ms resolution data into a sequence of length 10:

$$X = [x_1, x_2, \ldots, x_{10}]$$

where $x_t \in R^{192}$ represents the spike-count vector at time t.

(3) Normalization and Standardization: To eliminate baseline firing-rate differences across channels, Z-score normalization was applied per channel:

$$x' = \frac{x - \mu}{\sigma}$$

where μ and σ were computed using training-set data and reused for validation/testing to avoid information leakage.

(4) Handling Missing Channels: Missing or malfunctioning channels were first identified according to abnormal firing statistics or continuous signal absence. For sessions with a small proportion of missing channels, the missing values were interpolated using adjacent or statistically similar channels within the same session. If the proportion of missing channels exceeded 5%, the corresponding session was discarded. This threshold was selected to balance data retention and signal reliability: a small number of missing channels can be compensated by interpolation, whereas excessive channel loss may alter the population-level firing pattern and introduce unreliable spatial–temporal representations. After applying this criterion, 1893 valid samples were retained for model training and validation.

(5) Class-Balancing Strategy: To mitigate class imbalance during training, class-weighted cross-entropy was used so that minority classes contributed proportionally more to the loss function.

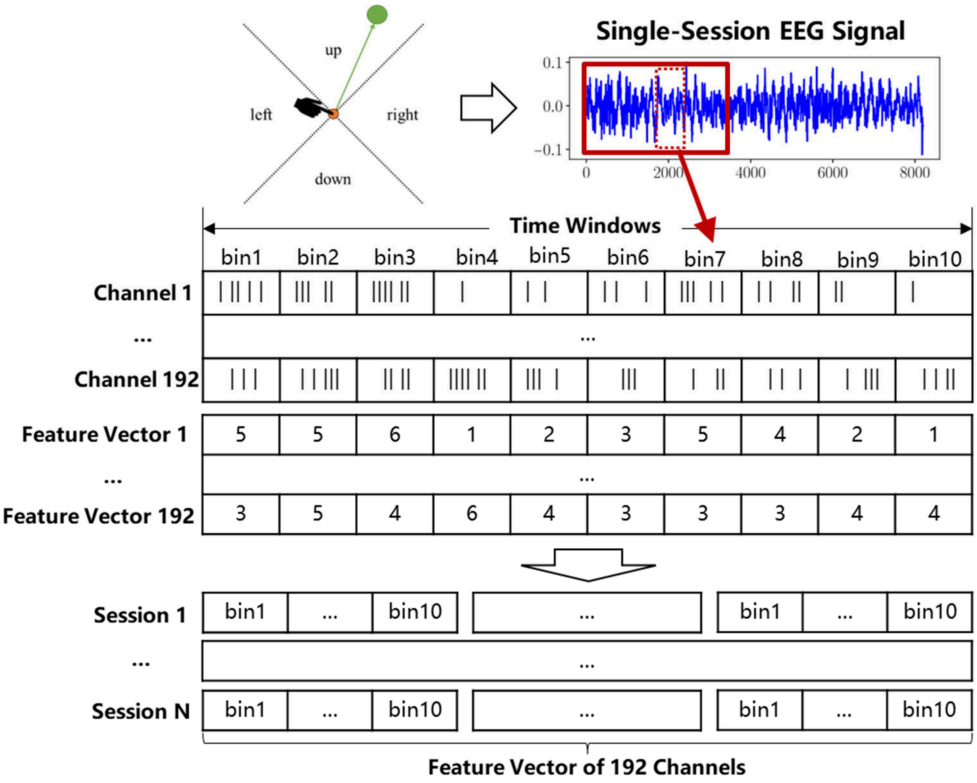


Figure 3. Overview of the data preprocessing pipeline.

3. Decoding Model and Transfer Mechanism

In this study, the CNN-Enhanced Transformer denotes the basic decoding architecture that integrates a one-dimensional convolutional front-end with a Transformer encoder. When the statistical distribution alignment module based on MMD and CORAL is further

incorporated for cross-session transfer learning, the complete framework is referred to as CNN-Trans-DA.

3.1. CNN-Enhanced Transformer Decoder

To effectively capture the multi-scale temporal and spatial structure of neural firing patterns associated with movement intention, we designed a hybrid neural decoding model consisting of a convolutional front-end, Transformer encoder layers, and a classification head (Figure 4). This section describes each component and its functional role within the overall architecture.

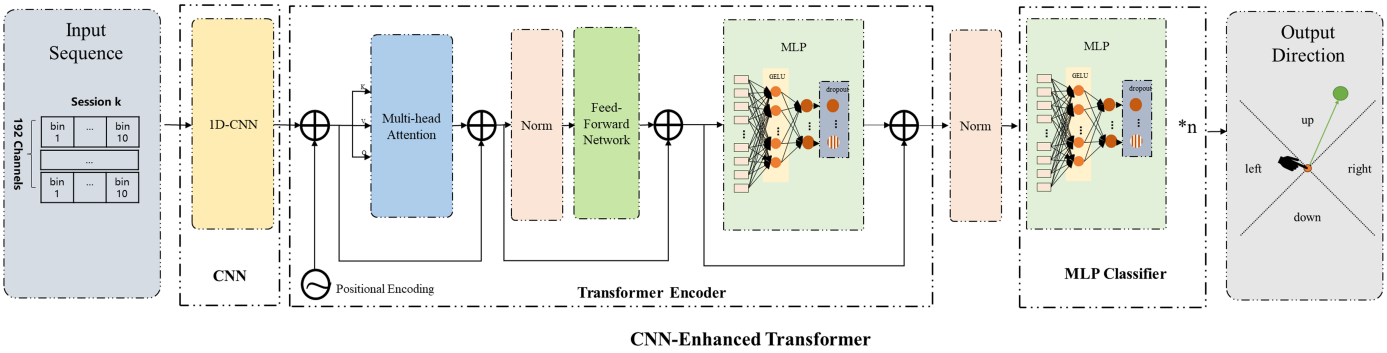


Figure 4. Schematic diagram of the CNN-enhanced Transformer architecture.

3.1.1. Convolutional Feature Extraction Module

Neural spike activity exhibits rich local temporal patterns and high-frequency fluctuations that are critical for intention decoding. While standard Transformers excel in modeling long-range dependencies, they are less effective in capturing short-term, high-resolution patterns. Therefore, a 1D convolutional front-end was employed to extract and enhance local spatiotemporal structure before attention-based processing.

1. 1D Convolution Front-end: Given an input sample represented as a T × C matrix, applying a 1D convolution across time is formulated as follows:

$$h_t^{(l)} = \sigma\left(\sum_{i=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} W_i^{(l)} x_{t+i} + b^{(l)}\right)$$

   where k is the kernel size and $\sigma(\cdot)$ is the ReLU activation. This operation aggregates local temporal neighborhoods, effectively emphasizing meaningful transient patterns and suppressing noise.
2. Input Layer: The raw T × C spike-count matrix is first normalized and optionally variance-stabilized and then reshaped as a time series for 1D CNN processing. When electrode arrays exhibit spatial structure, a channel remapping step may precede convolution; however, this work prioritizes temporal 1D convolution to preserve dynamical trends. Convolutional outputs are normalized (BatchNorm or LayerNorm), combined with residual connections, and projected via a linear layer to obtain the Transformer input embeddings:

$$z_t = Wh_t + b$$

The CNN module is placed before the Transformer encoder for two main reasons. First, neural spike trains usually contain short-term local firing patterns and transient fluctuations that are significantly related to movement intention. A one-dimensional convolutional layer can aggregate neighboring temporal bins and enhance local firing structures before

global attention modeling. Second, the convolutional front-end acts as a lightweight feature smoother and noise suppressor, reducing the influence of isolated abnormal spikes and improving the stability of the subsequent Transformer encoder. Therefore, the CNN module serves as a local temporal pattern enhancer, while the Transformer encoder further captures long-range dependencies across the entire spike-count sequence.

3.1.2. Transformer Encoder

After local feature extraction, global temporal dependencies are integrated through the Transformer encoder, which combines multi-head self-attention with position-wise feed-forward networks. The multi-head self-attention, feed-forward network, and layer normalization operations used in this section follow the standard Transformer formulation and normalization strategy reported in previous studies [23,24].

(1) Multi-Head Self-Attention (MHSA): Given the input embedding sequence, the following is calculated:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multiple heads capture relationships across different representational subspaces:

$$MHSA(E) = Concat(head_1, \ldots head_H)W_O$$

(2) Feed-Forward Network: Each attention layer is followed by a position-wise feed-forward network:

$$FFN(x) = W_2 GELU(W_1 x + b_1) + b_2$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{\text{model}}}$.

(3) Layer Normalization: To ensure training stability and accelerate convergence, Layer Normalization is applied before and after each sublayer (both the attention and FFN blocks). Given a feature vector x, LayerNorm is computed as follows:

$$LayerNorm(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta$$

where μ and σ^2 denote the mean and variance of x, while γ and β are learnable scaling and bias parameters. LayerNorm plays a particularly important role in this model: because neural signals may undergo fluctuations in scale and distribution after convolutional processing, LayerNorm stabilizes feature distributions at each timestep, reduces gradient oscillation, and enables deeper encoder stacking without training degradation.

3.1.3. Spatiotemporal Feature Fusion and Output Module

After the convolutional front-end and self-attention layers extract multi-scale spatiotemporal representations, these sequence-level features must be compressed into a sample-level vector for discriminative classification. This module performs feature vectorization, nonlinear transformation, and regularization, ultimately producing the probability distribution over the four movement directions [25,26].

(1) Multi-Layer Perceptron: The Transformer encoder outputs a temporal feature matrix $F \in R^{T \times d_{model}}$. To obtain a discriminative sample-level representation, features are first aggregated across time using global average pooling to form a vector f. The vector f is then passed through a two-layer MLP for further transformation:

$$u = GELU(W_a f + b_a)\hat{y} = softmax(W_b u + b_b)$$

The first MLP layer uses GELU activation to provide smooth nonlinear mapping, while the final layer produces the predicted probability distribution over the four fingertip movement directions. The MLP enables the model to learn a low-dimensional discriminative subspace optimal for classification based on the high-dimensional spatiotemporal features, and Dropout is used to enhance generalization.

(2) Output Layer: The output layer is trained using cross-entropy loss. Given the one-hot ground-truth label y, the classification loss is as follows:

$$L_{cls} = -\sum_{c=1}^{4} y_c log\hat{y}_c$$

(3) Multi-Level Dropout Mechanism: To prevent overfitting and improve robustness under cross-condition variability, Dropout is integrated at multiple levels, including attention matrices, FFN hidden layers, and the MLP classifier head. For a given activation z, Dropout is defined as follows:

$$\tilde{z} = \frac{z \odot m}{1-p}$$

where $m \sim Bernoulli(1-p)$ is an independent dropout mask and p is the drop probability. Using different p-values at different stages (e.g., $p = 0.1$ in attention layers, $p = 0.2$ in FFN, and $p = 0.3$ in the classifier head) preserves expressive capacity while improving tolerance to noise and distribution shift, thereby yielding more stable performance in cross-session transfer.

3.1.4. Model Processing Workflow

Based on the above modules, this section summarizes the full model workflow. The raw spike-count sequence is first normalized and windowed, then fed into the convolutional front-end. The 1D-CNN extracts local temporal patterns along the time axis and produces multi-scale representations. The convolutional outputs are then linearly projected into the Transformer embedding space and combined with positional encodings to preserve temporal order.

Within the Transformer encoder, multi-head self-attention models global dependencies among features, capturing movement-related structure across temporal windows through parallel attention heads. The subsequent feed-forward network applies nonlinear transformations at each timestep to enhance higher-order feature representations. Stacking multiple encoders yields hierarchical spatiotemporal integration, progressing from local to global and from low-level to high-level semantics.

The resulting temporal features are compressed via global average pooling to produce a sample-level vector, which is then fed into a two-layer MLP classifier. The GELU activation and multi-level Dropout enhance nonlinear expressiveness and generalization. Finally, a softmax layer outputs the probability distribution over the four fingertip movement directions.

Overall, the workflow implements a hierarchical "convolutional local enhancement → attention-based global integration → MLP discriminative mapping" architecture, enabling the model to maintain both discriminability and decoding stability under cross-session statistical variations. With these enhancements, the CNN-Enhanced Transformer achieves more accurate and robust neural decoding under limited sample conditions.

3.2. Statistical Transfer Learning Mechanism

Neural firing statistics differ substantially across experimental conditions, for example, across sessions, days, or sensory contexts. As a result, a model trained under a single condition often fails to generalize directly to new conditions. To mitigate this, we introduce a statistical distribution-alignment-based transfer learning mechanism to enhance cross-condition adaptability.

The key idea is to transfer feature knowledge learned in the source condition to a target condition with limited labeled data. Let the source and target domains be $D_S = \{X_S, P(X_S)\}$ and $D_T = \{X_T, P(X_T)\}$, with corresponding tasks $T_S = \{Y, P(Y|X_S)\}$ and $T_T = \{Y, P(Y|X_T)\}$. The source domain contains labeled neural samples, while the target domain contains unlabeled or sparsely labeled samples; both domains share the same label space.

During training, distribution-alignment constraints are imposed on the intermediate Transformer feature space to enforce statistical consistency between source and target features. We combine Maximum Mean Discrepancy (MMD) and Correlation Alignment (CORAL) to jointly align marginal and conditional distributions. MMD measures mean discrepancies in a high-dimensional kernel space, enforcing marginal distribution alignment. CORAL aligns covariance matrices, correcting feature correlations to balance intra-class distributions. The motivation for combining MMD and CORAL is that they characterize different aspects of cross-session distribution discrepancy. MMD mainly measures the difference between the mean embeddings of source and target features in a reproducing kernel Hilbert space, and is therefore effective for reducing marginal distribution mismatch. In contrast, CORAL aligns the covariance matrices of source and target features, which helps correct second-order statistical differences and feature-correlation shifts. In cross-session neural recordings, distribution drift is not limited to changes in average firing rates, but also includes variations in inter-channel correlation and temporal covariance structures. Using MMD or CORAL alone may only address one aspect of this shift, whereas their combination provides complementary first-order and second-order statistical alignment. Therefore, the joint MMD-CORAL strategy improves the robustness of cross-session neural decoding while maintaining a relatively simple and stable optimization process.

Under these complementary constraints, the model gradually learns domain-invariant representations that reduce statistical divergence between the two domains, mitigating the risk of negative transfer. The full loss combines classification and transfer losses:

$$L = \omega_1 L_{S-cls} + \omega_2 L_{MMD} + \omega_2 L_{CORAL} + \omega_4 L_{T-cls}$$

where L_{S-cls} and L_{T-cls} denote the cross-entropy classification losses for the source and target domain samples, respectively. The terms L_{MMD} and L_{CORAL} represent the marginal and conditional distribution alignment components. The coefficients ω_1 to ω_2 are weighting parameters that balance the contribution of each loss term. By jointly optimizing this composite objective function, the model achieves effective cross-condition distribution adaptation while maintaining discriminative decoding performance.

Compared with adversarial or manifold-alignment methods, this statistical transfer strategy is computationally simple and easy to optimize, making it well suited for high-dimensional neural data with limited sample sizes. Moreover, the Transformer's self-attention mechanism dynamically focuses on informative temporal segments, further enhancing robustness in cross-condition settings.

3.3. Implementation Details

The model is implemented in PyTorch 1.10.2. Adam is used as the optimizer with a learning rate of 1×10^{-4}, batch size of 64, and 200 training epochs. During training, the CNN-Enhanced Transformer decoder is first pretrained on source-condition data to learn the fundamental neural-to-movement mapping. It is then fine-tuned on the target condition using a small number of updates, modifying only selected attention and FFN parameters to reduce dependence on target samples and prevent overfitting.

The overall training pipeline is illustrated in Figure 5. Neural signals are processed by the 1D-CNN to extract local temporal features, which are then fed to the Transformer encoder for global dependency modeling. Intermediate features are aligned across conditions under statistical constraints. Finally, the two-layer MLP maps the aligned features to the four fingertip movement classes.

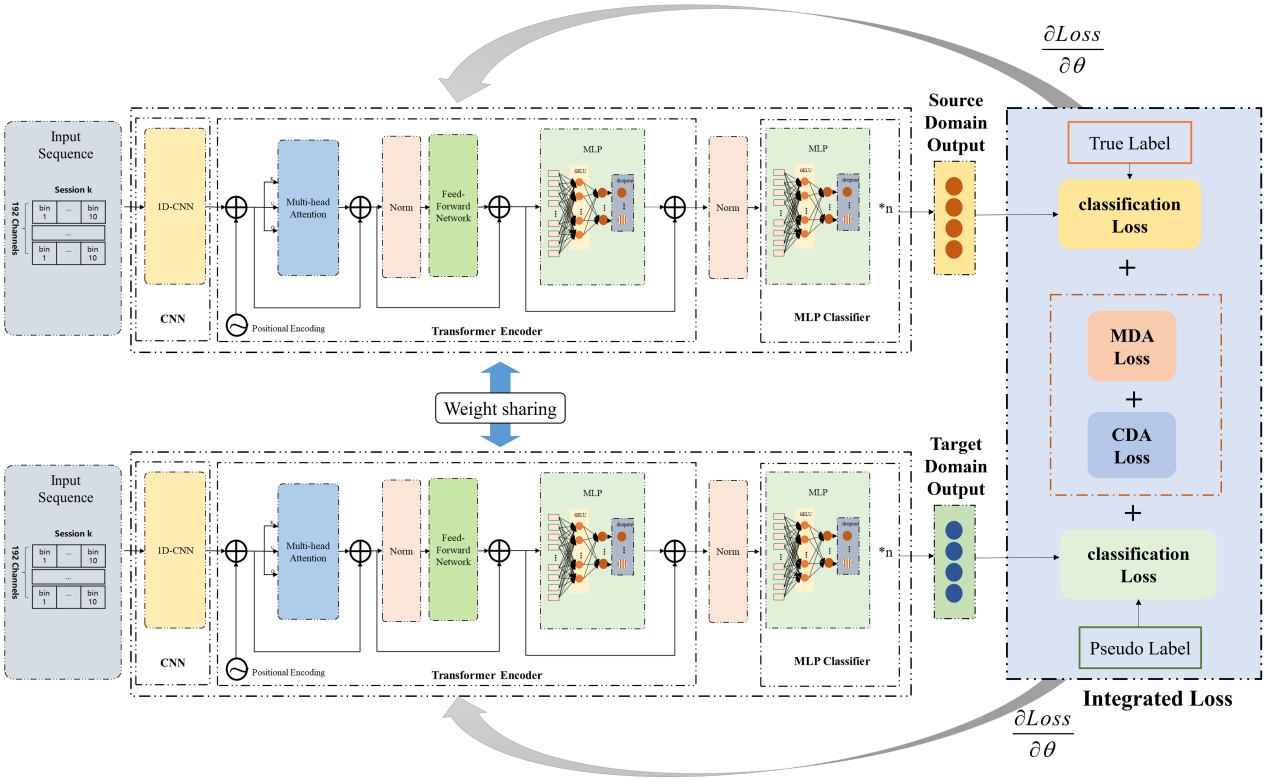


Figure 5. Schematic diagram of the CNN-Trans-DA model architecture.

By integrating CNN-based local enhancement with statistical transfer learning, the model achieves strong decoding accuracy under single-condition training and exhibits significant generalization stability during cross-condition transfer.

For completeness and reproducibility, Algorithm 1 summarizes the training procedure of the proposed CNN-Trans-DA framework. The model first extracts source- and target-session neural features through the CNN-Enhanced Transformer feature extractor. The movement-direction classifier is optimized using the available class labels, while the MMD and CORAL losses are jointly introduced to reduce cross-session feature-distribution discrepancies. Through this joint optimization process, the model learns discriminative and session-invariant neural representations for robust motor intention decoding.

Algorithm 1 Training Procedure of CNN-Trans-DA for Cross-Session Neural Decoding

Require:

Source-session dataset $Ds = (x_s, y_s)$
Target-session dataset $Dt = (x_t, y_t)$ or unlabeled target samples x_t
CNN-Enhanced Transformer feature extractor $F(\cdot)$
Movement-direction classifier $C(\cdot)$
Number of training epochs E
Learning rate η
Loss weights λ_1, λ_2, λ_3, and λ_4

Ensure:

Trained CNN-Trans-DA model for cross-session motor neural decoding
Initialize the parameters of $F(\cdot)$ and $C(\cdot)$
for epoch = 1 to E do
Sample mini-batches from the source session and target session
Extract source and target features:
    $z_s = F(x_s), z_t = F(x_t)$
Predict movement-direction probabilities:
    $p_s = C(z_s), p_t = C(z_t)$
Compute the source classification loss Ls using source labels y_s
if target labels are available then
    Compute the target classification loss Lt using target labels y_t
else
    Set Lt = 0 or use pseudo-labels for target samples
end if
Compute the MMD loss L_{MMD} between z_s and z_t
Compute the CORAL loss L_{CORAL} between z_s and z_t
Calculate the total loss:
    $L = \lambda_1 L_s + \lambda_2 L_t + \lambda_3 L_{MMD} + \lambda_4 L_{CORAL}$
Update the parameters of $F(\cdot)$ and $C(\cdot)$ by backpropagation
end for
return Trained CNN-Trans-DA model

4. Experiments and Results

To validate the effectiveness of the proposed CNN-enhanced Transformer with Statistical Transfer Learning for cross-condition decoding of monkey motor neural signals, this section provides a systematic analysis from three perspectives: data preprocessing parameters, baseline model comparisons, and transfer-decoding performance.

4.1. Data Preprocessing and Session Parameter Configuration

Prior to model training, the raw spike-train neural data underwent rigorous preprocessing and parameter configuration to ensure consistency in temporal resolution and statistical distribution across samples.

The original signals were segmented using a delay time of 0.1 s and divided into 20 temporal windows (num_windows), each with a duration of 0.04 s, resulting in a total analysis span of 0.9 s. Low-quality trials and abnormal segments were removed according to the preprocessing criteria described in Section 2.3, including abnormal movement duration, irregular trajectories, and excessive missing-channel ratios. After preprocessing, a total of 1893 valid samples were retained for training and validation. The detailed preprocessing parameters are summarized in Table 1.

Table 1. Preprocessing Parameters for Raw Neural Signals.

Parameter	Data
delay_time	0.1 s
num_windows	20
window_length	0.04 s
total_time	0.9 s
valid samples	1893

4.2. Decoding Performance on Monkey Motor Neural Signals

Under fixed experimental conditions, we evaluated four neural network architectures—CNN-Enhanced Transformer, Transformer, CNN, and BiLSTM—for the task of classifying hand-movement-related neural activity, in order to compare their spatiotemporal modeling capabilities. Quantitative results (Table 2) show that the proposed CNN-Enhanced Transformer achieves the best balance between local feature extraction and global dependency modeling, attaining an average accuracy of 84.64%. This significantly outperforms CNN (80.78%) and BiLSTM (79.29%), and provides a 3.42-percentage-point improvement over the standard Transformer (81.22%).

Table 2. Comparison of Neural Signal Decoding Accuracies.

Model	CNN-Enhanced Transformer	Transformer	CNN	BiLSTM
Mean Accuracy	84.64%	81.22%	80.78%	79.29%

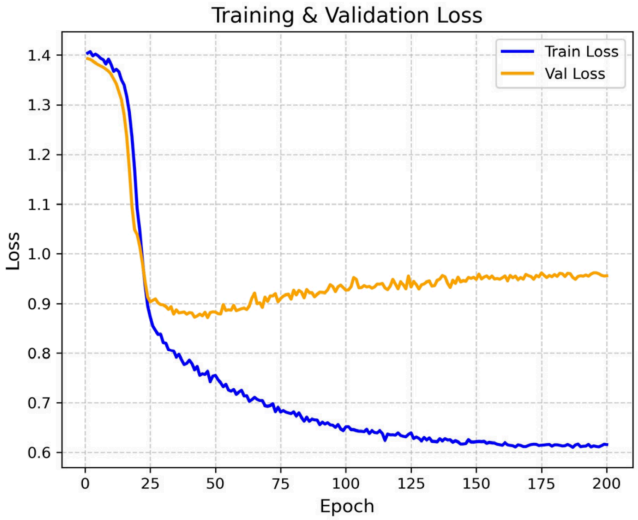
Training dynamics further confirm its robustness. As shown in Figure 6, both the training and validation losses of the CNN-Enhanced Transformer decrease smoothly, exhibiting fast and stable convergence. Its validation accuracy curve (Figure 7) rises rapidly in the early epochs and stabilizes around 85% after approximately 50 epochs, indicating strong generalization and reliable training behavior. In contrast, the Transformer baseline shows larger oscillations in its loss curves, suggesting greater sensitivity to hyperparameter choices, while CNN and BiLSTM converge stably but yield inferior final accuracy.

Hyperparameter analysis summarized in Table 3 indicates that the configuration with learning rate 5×10^{-5}, dropout rate 0.4, and weight decay 5×10^{-3} (Config 1) best balances fitting and generalization, achieving $82.3 \pm 1.2\%$ accuracy. Both excessively high/low learning rates and inappropriate dropout settings slow convergence or introduce overfitting, highlighting the importance of proper regularization for CNN-Enhanced Transformer architectures.

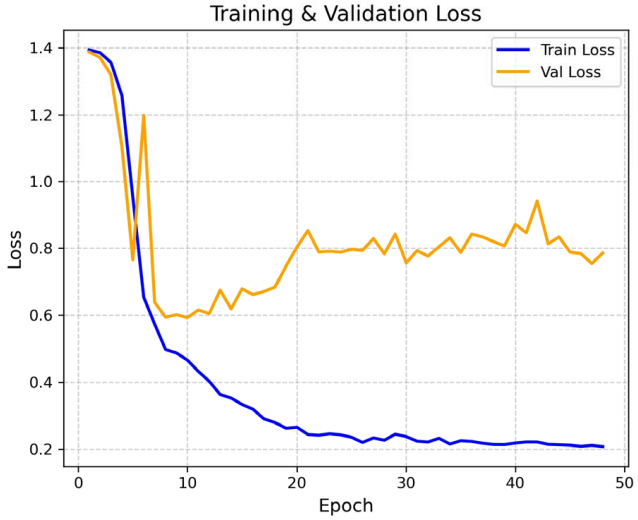
Table 3. Hyperparameter Configuration and Accuracy Comparison.

Setting	LR *	DR *	WD *	Acc * (%)	Interpretation
1	5×10^{-5}	0.4	5×10^{-3}	82.3 ± 1.2	Optimal
2	1×10^{-4}	0.4	5×10^{-3}	80.7 ± 0.9	Large LR *
3	1×10^{-5}	0.4	5×10^{-3}	78.1 ± 1.5	Slow convergence
4	5×10^{-5}	0.2	5×10^{-3}	76.9 ± 2.1	Overfitting
5	5×10^{-5}	0.6	5×10^{-3}	81.5 ± 1.1	Underfitting
6	5×10^{-5}	0.4	1×10^{-3}	79.2 ± 1.8	Weak regularization

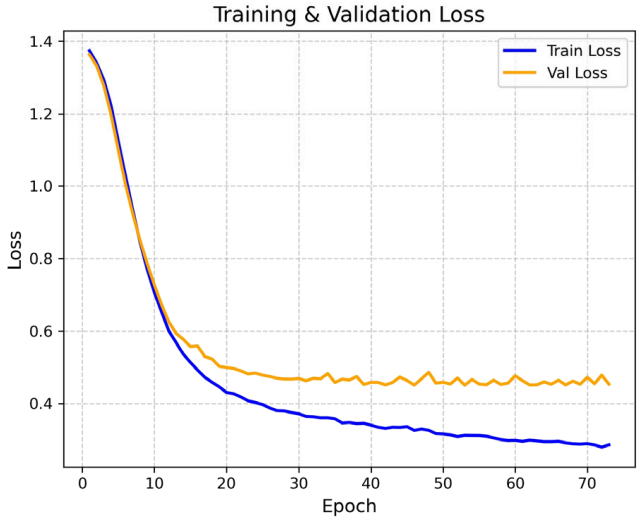
* LR: Learning Rate; DR: Dropout Rate; WD: Weight Decay; Acc Accuracy.



(**a**) CNN-Enhanced Transformer.

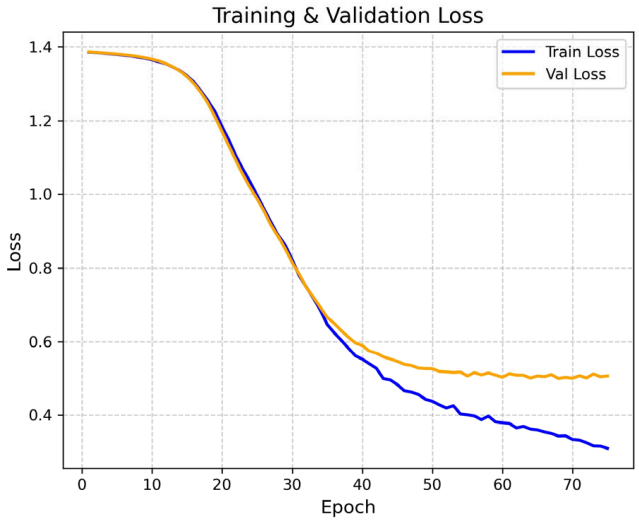


(**b**) Transformer.



(**c**) CNN.

Figure 6. *Cont.*



(**d**) BiLSTM.

Figure 6. Training Loss Curves of Different Models.

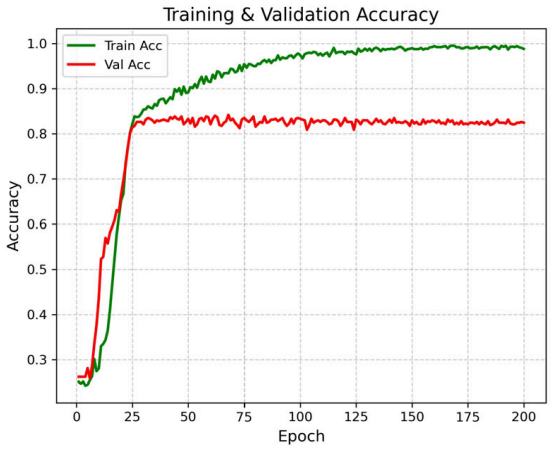


Figure 7. Accuracy Curve of the CNN-Enhanced Transformer Model.

Overall, these results demonstrate that augmenting the Transformer with convolutional front-end modules enhances its ability to capture local firing dynamics and suppress noise, thereby improving decoding accuracy and stability.

4.3. Cross-Condition Transfer Decoding Results

To evaluate the model's robustness to experimental variability, we further conducted cross-session transfer experiments. Neural recordings from different days were treated as source and target domains, and statistical transfer-learning strategies were applied to align feature distributions. All four model architectures (CNN-Enhanced Transformer, Transformer, CNN and BiLSTM) were trained under identical transfer settings to ensure fair comparison.

Results (Table 4) show that the proposed CNN-Trans-DA model achieves the highest cross-condition accuracy of 82.42%, outperforming the Transformer baseline by 1.92%. As shown in Figure 8, CNN-Trans-DA exhibits smoother loss convergence and consistently improving validation accuracy, indicating that distribution-alignment constraints effectively mitigate performance degradation caused by session-to-session drift.

Table 4. Comparison of Cross-Session Transfer-Decoding Results.

Base Model Architecture	Mean Accuracy
CNN-Trans-DA (CNN-Enhanced Transformer + DA)	82.42%
Transformer	80.50%
CNN	81.21%
LSTM	76.97%

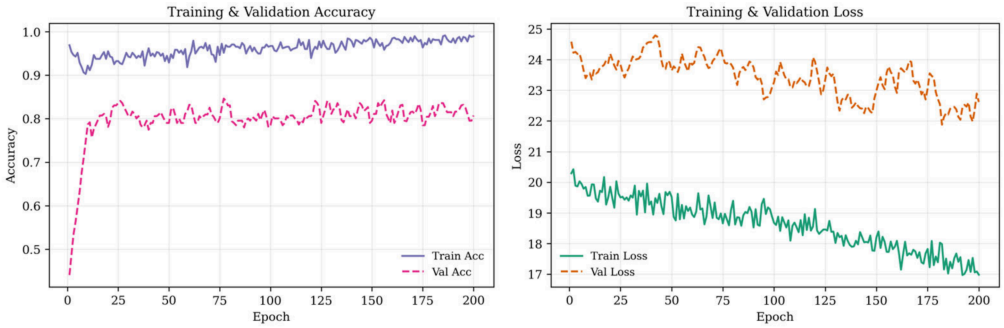


Figure 8. Accuracy and Loss Curves of the CNN-Trans-DA Model.

The confusion matrix (Figure 9) reveals balanced recognition across movement classes. Misclassifications primarily occur between physiologically adjacent motion categories, consistent with natural overlaps in neural representations. Hyperparameter analysis (Table 5) indicates that the weighting scheme $\omega_1 = 0.6, \omega_2 = 0.3, \omega_3 = 0.1, \omega_4 = 0.05$ provides an effective trade-off between classification loss and alignment loss, producing the most stable transfer performance.

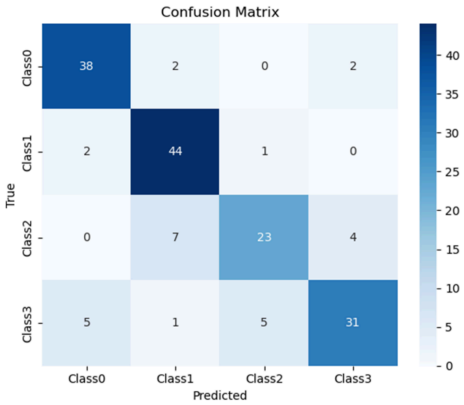


Figure 9. Cross-Condition Confusion Matrix of the CNN-Trans-DA Model.

Table 5. Weight configuration and accuracy comparison for transfer-learning loss terms.

Weight Configuration (ω_1,ω_2,ω_3,ω_4)	Mean Accuracy
(0.6, 0.3, 0.1, 0.05)	82.42%
(0.5, 0.4, 0.1, 0.05)	81.19%
(0.7, 0.2, 0.1, 0.05)	80.11%

Taken together, the CNN-Trans-DA model demonstrates strong adaptability and robustness in cross-session decoding, validating the effectiveness of the embedded statistical alignment mechanism for mitigating distribution shifts in neural datasets.

4.4. Discussion

The combined analyses indicate that the CNN-enhanced Transformer architecture effectively captures both local and global temporal structures in motor-related neural activity. Under single-condition settings, it achieves 84.64% accuracy, outperforming conventional CNN (80.78%) and Transformer (81.22%) models, thus validating its advantage in spatiotemporal feature integration.

In cross-condition experiments, the CNN-Trans-DA variant further demonstrates structural robustness. By introducing statistical distribution-alignment constraints, it maintains 82.42% accuracy across sessions—a 1.92% improvement over the baseline Transformer. Training dynamics (Figure 8) show more stable convergence, and the confusion matrix (Figure 9) reveals physiologically plausible misclassification patterns concentrated between neighboring motions.

These findings confirm that the convolutional front-end enhances local feature sensitivity and noise robustness, while the statistical transfer mechanism effectively counteracts distribution drift across recording sessions. As a result, the CNN-Enhanced Transformer architecture achieves a strong balance between decoding accuracy and generalization.

The proposed framework provides an effective solution for motor-intent decoding, and its hybrid architectural design can inform broader applications in biological signal processing. From the perspective of computational efficiency, the proposed adaptation strategy is relatively lightweight. The MMD and CORAL losses are only used during the training or adaptation stage to constrain the feature distributions between source and target sessions. During online inference, the model only requires a standard forward pass through the CNN front-end, Transformer encoder, and MLP classifier, without computing cross-domain alignment losses. Therefore, the additional computational burden introduced by the distribution alignment module does not affect real-time decoding after deployment.

Although the present study focuses on four-direction monkey fingertip movement decoding, the proposed CNN-Trans-DA framework is not restricted to this specific task. The convolutional front-end and Transformer encoder can be adapted to neural signals with different channel numbers, sampling rates, and temporal resolutions, while the output classifier can be modified according to the number or type of motor labels. Therefore, the framework has potential applicability to more complex motor tasks, such as multi-directional reaching, continuous trajectory decoding, and grasp-related movement decoding. For human neural datasets, including ECoG, EEG, or intracortical recordings, the same framework may also be applied after appropriate adjustment of the input representation and preprocessing pipeline. Nevertheless, further validation on larger cross-subject and cross-task datasets is required before drawing general conclusions about its applicability to clinical BMI systems.

5. Conclusions

To address the challenges posed by distribution shifts and insufficient model generalization in cross-session motor neural decoding, this study proposes a hybrid architecture (CNN-Trans-DA), combining convolutional neural networks with Transformer mechanisms enhanced through statistical transfer learning for improved domain adaptation.

The convolutional modules effectively capture local spatiotemporal firing patterns, while the self-attention mechanism models long-range temporal dependencies. Furthermore, by jointly aligning the marginal and conditional distributions of source and target domains, the model significantly reduces cross-session variability.

Experimental results demonstrate that the proposed approach substantially outperforms CNN, BiLSTM, and standard Transformer baselines in monkey motor-intent decoding tasks. Under cross-session transfer settings, CNN-Trans-DA achieves superior

classification accuracy, stronger training stability, and more favorable convergence behavior, confirming its robustness against real-world neural signal variability.

Looking ahead, the proposed framework may be further enhanced in the following directions:

1. Explainability: Incorporating methods such as attention-weight visualization and attribution analysis to reveal key neural features and their roles in motor encoding.
2. Advanced transfer-learning strategies: Exploring adversarial training, meta-learning, or self-supervised pretraining to enhance adaptability across individuals and task conditions.
3. Multimodal integration: Combining cortical signals with EMG, behavioral trajectories, and other physiological modalities to develop more biologically grounded and practical decoding systems.

This study provides new methodological insights for developing high-generalization BCI decoding models and supports the transition of brain–machine interface systems from controlled laboratory environments toward real-world deployment.

Author Contributions: Conceptualization, M.W., T.G. and Y.D.; methodology, M.W., T.G. and Y.D.; software, T.G. and Y.D.; validation, M.W., T.G. and Y.D.; formal analysis, M.W. and Y.R.; investigation, M.W., Y.R. and S.H.; resources, M.W. and J.Z.; data curation, Y.R. and S.H.; writing—original draft preparation, M.W. and Y.R.; writing—review and editing, M.W., Y.R. and J.Z.; visualization, T.G. and Y.D.; supervision, J.Z.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Major Program (JD) of Hubei Province (2023BAA005).

Institutional Review Board Statement: No new animal experiments were conducted in this study. The neural recording data analyzed in this work were obtained from a previously published/publicly available dataset. The original animal experimental procedures and ethical approvals were reported in the source study. This study only performed secondary analysis of the existing neural data.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Ahmadi, N.; Constandinou, T.G.; Bouganis, C.-S. Robust and accurate decoding of hand kinematics from entire spiking activity using deep learning. *J. Neural Eng.* **2021**, *18*, 026011. [CrossRef]
2. Sachdeva, P.S.; Livezey, J.A.; Dougherty, M.E.; Gu, B.-M.; Berke, J.D.; Bouchard, K.E. Improved inference in coupling, encoding, and decoding models and its consequence for neuroscientific interpretation. *J. Neurosci. Methods* **2021**, *358*, 109195. [CrossRef] [PubMed]
3. Jiang, Y.; Gao, T.; Dai, Y.; Si, R.; Hao, J.; Zhang, J.; Gao, D.W. Very short-term residential load forecasting based on deep-autoformer. *Appl. Energy* **2022**, *328*, 120120. [CrossRef]
4. Kapoor, J.; Schulz, A.; Vetter, J.; Pei, F.; Gao, R.; Macke, J.H. Latent diffusion for neural spiking data. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): San Diego, CA, USA, 2024; Volume 37, pp. 118119–118154.
5. Keshtkaran, M.R.; Sedler, A.R.; Chowdhury, R.H.; Tandon, R.; Basrai, D.; Nguyen, S.L.; Sohn, H.; Jazayeri, M.; Miller, L.E.; Pandarinath, C. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nat. Methods* **2022**, *19*, 1572–1577. [PubMed]
6. Luo, J.; Li, J.; Mao, Q.; Liu, Y.; Yan, W.; Xue, Y.; Shi, Z.; Hei, X. MCDDT: Mirror Center Loss-Based Dual-Scale Dual-Softmax Transformer for Multisource Subjects Transfer Learning in Motor Imagery Recognition. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2543314.
7. Makin, G.; O'Doherty, J.E.; Cardoso, M.M.B.; Sabes, P.N. Superior arm-movement decoding from cortex with a new, unsupervised learning algorithm. *J. Neural Eng.* **2018**, *15*, 026010. [CrossRef] [PubMed]

8. Qi, Y.; Zhu, X.; Xu, K.; Ren, F.; Jiang, H.; Zhu, J.; Zhang, J.; Pan, G.; Wang, Y. Dynamic ensemble Bayesian filter for robust control o a human brain-machine interface. *IEEE Trans. Biomed. Eng.* **2022**, *69*, 3825–3835. [PubMed]
9. Savolainen, O.W. The significance of neural inter-frequency power correlations. *Sci. Rep.* **2021**, *11*, 23190. [CrossRef] [PubMed]
10. Song, Y.; Wang, Y.; He, H.; Gao, X. Recognizing Natural Images From EEG With Language-Guided Contrastive Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 15896–15910. [CrossRef] [PubMed]
11. Valencia, D.; Mercier, P.P.; Alimohammad, A. In vivo neural spike detection with adaptive noise estimation. *J. Neural Eng.* **2022**, *19*, 046018. [CrossRef]
12. Ye, J.; Collinger, J.; Wehbe, L.; Gaunt, R. Neural data transformer 2: Multi-context pretraining for neural spiking activity. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): San Diego, CA, USA, 2023; Volume 36, pp. 80352–80374.
13. Zhu, X.; Qi, Y.; Pan, G.; Wang, Y. Tracking functional changes in nonstationary signals with evolutionary ensemble bayesian model for robust neural decoding. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022.
14. Wang, J.; Bi, L.; Fei, W.; Guan, C. Decoding single-hand and both-hand movement directions from noninvasive neural signals. *IEEE Trans. Biomed. Eng.* **2020**, *68*, 1932–1940.
15. Livezey, J.A.; Glaser, J.I. Deep learning approaches for neural decoding across architectures and recording modalities. *Brief. Bioinform.* **2021**, *22*, 1577–1591. [PubMed]
16. Wang, Y.; Liu, H.; Wang, Y.; Xuan, C.; Hou, Y.; Feng, S.; Liu, H.; Liao, Y.; Wang, Y. Progress, challenges and future of linguistic neural decoding with deep learning. *Commun. Biol.* **2025**, *8*, 1350. [CrossRef] [PubMed]
17. Aoe, J.; Fukuma, R.; Yanagisawa, T.; Harada, T.; Tanaka, M.; Kobayashi, M.; Inoue, Y.; Yamamoto, S.; Ohnishi, Y.; Kishima, H. Automatic diagnosis of neurological diseases using meg signals with a deep neural network. *Sci. Rep.* **2019**, *9*, 5057. [CrossRef] [PubMed]
18. Zhang, R.; Zong, Q.; Dou, L.; Zhao, X.; Tang, Y.; Li, Z. Hybrid deep neural network using transfer learning for EEG motor imagery decoding. *Biomed. Signal Process. Control* **2021**, *63*, 102144. [CrossRef]
19. Tan, Y.; Zhang, E.; Li, Y.; Huang, S.-L.; Zhang, X.-P. Transferability guided cross-domain cross-task transfer learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 2423–2436. [PubMed]
20. Wu, D. Revisiting Euclidean alignment for transfer learning in EEG-based brain-computer interfaces. *J. Neural Eng.* **2025**, *22*, 031005.
21. She, Q.; Cai, Y.; Du, S.; Chen, Y. Multi-source manifold feature transfer learning with domain selection for brain-computer interfaces. *Neurocomputing* **2022**, *514*, 313–327.
22. Varone, G.; Boulila, W.; Driss, M.; Kumari, S.; Khan, M.K.; Gadekallu, T.R.; Hussain, A. Finger pinching and imagination classification: A fusion of CNN architectures for IoMT-enabled BCI applications. *Inf. Fusion.* **2024**, *101*, 102006.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): San Diego, CA, USA, 2017; Volume 30.
24. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
25. Wen, J.; Antoniades, M.; Yang, Z.; Hwang, G.; Skampardoni, I.; Wang, R.; Davatzikos, C. Dimensional neuroimaging endophenotypes: Neurobiological representations of disease heterogeneity through machine learning. *Biol. Psychiatry* **2024**, *96*, 564–584. [CrossRef] [PubMed]
26. Li, A.; Wang, Z.; Zhao, X.; Xu, T.; Zhou, T.; Hu, H. MDTL: A novel and model-agnostic transfer learning strategy for cross-subject motor imagery BCI. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2023**, *31*, 1743–1753. [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.