STABILIZED NEURAL DYNAMICS FOR BEHAVIORAL DECODING VIA HIERARCHICAL DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Brain-Computer Interfaces (BCI) have demonstrated significant potential in neural rehabilitation. However, the variability of non-stationary neural signals often leads to instabilities of behavioral decoding, posing critical obstacles to chronic applications. Domain adaptation technique offers a promising solution. Nonetheless, the existing direct adaptation within latent spaces could result in feature deviations. Therefore, developing a stable and efficient alignment framework is crucial for neural decoders. In this work, we find that dynamical latent features can be extracted from neural dynamics utilizing causal architectures. We also demonstrate that the process of self-consistent alignment can generate more stable latent features. Based on these insights, we propose a novel hierarchical domain adaptation (HDA) method for the alignment of dynamical latent features. Using Lyapunov theory, we further analytically validate the stability of dynamical features, which experimentally exhibit significant enhancements across various datasets. Our HDA approach effectively addresses the challenge of non-stationary neural signals, thereby potentially improving the reliability of BCIs.

1 INTRODUCTION

Brain-Computer Interfaces (BCI) offer a direct pathway for connecting the brain with external devices, demonstrating great potential in neural rehabilitation for people with paralysis (Collinger et al., 2018; Chaudhary et al., 2016; Willett et al., 2021; Metzger et al., 2023; Willett et al., 2023). Despite recent advances, one key challenge for BCIs is how to maintain stable performance, considering that the non-stationary neural recordings could vary across days (Perge et al., 2013; Wimalasena et al., 2020). The variability in neural signals could stem from various factors, such as environmental conditions (Santhanam et al., 2007), device degradation (Woeppel et al., 2021), physiological changes (Athalye et al., 2017) to foreign materials, and behavioral changes (Truccolo et al., 2008). Consequently, frequent recalibration of a BCI system is necessary to maintain its performance, leading to a critical barrier to chronic applications (Pandarinath & Bensmaia, 2022).

To alleviate the burden of recalibration, some studies aimed to develop automatic decoder adjustment approaches to cope with variability in neural signals(Wimalasena et al., 2020; Degenhart et al., 2020). One strategy is to align the neural signals across multiple days. These approaches allow neural decoders trained on one day to apply to another day directly. To achieve this, unsupervised domain adaptation (UDA) techniques have been employed to align the distributions of neural signals across different recording sessions. Existing UDA approaches for BCIs can be categorized into two types. The first type performs the distribution alignment in raw neural signal spaces (Farshchian et al., 2018; Ma et al., 2023). The second type aligns on the latent feature spaces and seeks for the spatio-temporal relationships in neural signals (Degenhart et al., 2020; Ju & Guan, 2022; Kobler et al., 2022; Cho et al., 2023; Jude et al., 2022).

Unfortunately, unlike conventional data such as images and videos, aligning neural signals is more challenging due to the inherently non-stationary nature of neural activities (Gallego et al., 2020). Directly aligning raw neural signals (Farshchian et al., 2018; Ma et al., 2023) or latent features (Karpowicz et al., 2022; Wang et al., 2023) may result in unstable features for decoding. Therefore, it is crucial to develop a stable and efficient alignment framework, thereby achieving a stable feature space for robust neural decoders.

Existing researches have shown that the brain executes various functions by converging towards attractors (Khona & Fiete, 2022), which are linked to dynamical stability in response to neural perturbations. Inspired by these observations, we propose that dynamical latent features can be extracted from neural dynamics utilizing causal architectures (Chen et al., 2024). Furthermore, we show that the process of self-consistent alignment within neural systems promotes the generation of more stable dynamical latent features. Building on these findings, we introduce a novel framework of hierarchical domain adaptation (HDA) that efficiently aligns dynamical latent features. Through validation grounded on Lyapunov theory (Angeli, 2002; Jiang & Wang, 2001), we analytically demonstrate that HDA enhances the dynamical stability of latent features, achieving stable neural decoders over extended periods. Experimental validation of HDA reveals significant improvements across various datasets. Our HDA approach effectively tackles the challenge of non-stationary neural signals, potentially improving the reliability of BCIs.

The main contributions of this paper are summarized as follows:

- **Causal Architectures**: Unlike existing UDA studies for BCI decoding, our research utilizes causal architectures (Chen et al., 2024) based on neural dynamics to extract latent features. Consequently, the cumulative final latent features (Gros, 2010) can be used for stable neural decoding. In addition, these dynamical features, derived from short-time windows, have the potential to meet the real-time operational requirements of BCIs.
- **Hierarchical Domain Adaptation**: We propose a novel framework for hierarchical domain adaptation (HDA) that enhances the dynamical stability of latent features, based on causal architectures. Our findings also indicate that a pre-controlled upper bound on latent feature deviations contributes to the dynamical stability using Lyapunov theory. A theoretical verification is provided in Section 3.4.
- **Experimental Validation**: We conduct extensive experiments on motor cortex datasets (Ma et al., 2023) to validate the superior performance of HDA compared with existing methods. Employing Lyapunov exponents, we have numerically verified that HDA enhances feature stability in non-stationary signals and effectively stabilizes behavioral decoding.

2 Related Work

Unsupervised Domain Adaptation Unsupervised Domain Adaptation (UDA) aims to bridge the gap between labeled source domain(s) and unlabeled target domains by matching their distributions. Some studies have achieved by minimizing discrepancies based on specific metrics (Peng et al., 2019a; Sun et al., 2016; Sun & Saenko, 2016), such as the maximum mean discrepancy (Long et al., 2015; 2017a). Inspired by Generative Adversarial Networks, another line of research utilizes domain adversarial training to obtain domain-invariant features (Saito et al., 2018; Sankaranarayanan et al., 2018; Chen et al., 2020; Long et al., 2018). For instance, the widely-used Domain Adversarial Neural Network (DANN) (Ganin & Lempitsky, 2015; Ganin et al., 2016) optimizes feature extractors to generate domain-invariant features that confuse the trained domain classifier.

As mentioned in Section 1, for stabilizing BCI decoding over time, UDA-based alignment approaches have been utilized for unsupervised recalibrations within raw signal and latent feature spaces. Recently, consistent low-dimensional latent dynamics have been leveraged as the latent features for alignment (Karpowicz et al., 2022; Wang et al., 2023; Vermani et al., 2023; Pandarinath et al., 2018; Fang et al., 2023; Safaie et al., 2023). These latent dynamics, situated within the neural manifold (Degenhart et al., 2020; Gallego et al., 2017; Mitchell-Heggs et al., 2023), are assumed to provide a stable underlying representation of behavior.

Nevertheless, some intrinsic features, including low signal-to-noise ratios (Hu et al., 2010), frequently result in instabilities when attempting to align the high-dimensional raw neural signals (Wang et al., 2023). Meanwhile, alignment in latent spaces typically assumes its stability ensured by neural manifolds (Gallego et al., 2017; Mitchell-Heggs et al., 2023), lacking further consideration for the dynamical stability of latent features. For instance, NoMAD (Karpowicz et al., 2022) and the source-free and unsupervised alignment (Vermani et al., 2023) directly match latent dynamics, which may overlook the potential instability of the source domain's extracted latent features. s In contrast, our method proposes a novel hierarchical alignment based on causal architectures in neural dynamics. We demonstrate that the iterative process of self-consistent alignment can generate more stable latent

features. This optimization enhances the dynamical stability of latent features, thereby stabilizing the neural manifolds against stochastic perturbations.

Representation Disentanglement Representation Disentanglement has been used in UDA to learn domain-invariant features. In the field of computer vision, researchers have successfully applied this technique to disentangle semantic latent features for tasks such as cross-domain image classification (Cai et al., 2019; Lee et al., 2021; Lv et al., 2022) and video action recognition (Wei et al., 2023). Various methods have been explored, including reweighting source features for meta knowledge transfer (Wei et al., 2021) and utilizing deep adversarial autoencoders (Peng et al., 2019b). In the realm of time series analysis, researchers have disentangled semantically meaningful factors to control the shape of ECG signals (Li et al., 2022).

In neural data analysis, researchers have focused on building robust and generalizable representations using advanced networks such as transformers (Ye & Pandarinath, 2021; Liu et al., 2022; Le & Shlizerman, 2022). Recently, unified frameworks have been proposed to enable scalable representations across sessions and subjects (Azabou et al., 2023). To achieve this, representation disentanglement has been employed to understand how different neural populations encode diverse external stimuli and their intrinsic connections with observable behavioral variables. Supervised learning techniques, supported by auxiliary variables (Zhou & Wei, 2020), along with self-supervised learning approaches, such as contrastive learning (Cheng et al., 2020) or swapping (Liu et al., 2021), are used to identify latent variables that are directly related to observable variables.

Existing disentanglement approaches for neural data analysis primarily focus on learning domain-invariant representations directly, without performing distribution alignment. Such methods may work well when source domains contain sufficient samples of various sessions and tasks (Wang et al., 2020; Parnami & Lee, 2022). Considering that UDA often requires little data (Ma et al., 2023), they are more practical when less source data is available. In this study, we propose HDA, and leverage disentanglement techniques as a tool to decompose latent spaces for better distribution alignment. The use of UDA with disentanglement techniques to stabilize BCI decoding performance has not been explored.

3 Methodology

3.1 Problem Formulation

We view the unsupervised recalibration of BCIs over time as a UDA problem (Long et al., 2017b). First, we define the domain $\mathcal{D}$ as follows: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i(t)$ (for $t = 0, \ldots, w-1$) represents a raw neural signal sample of window length w (with w being significantly shorter than the length of the entire trial) from one session, and $x_i(t) \in \mathbb{R}^m$. The behavioral label y_i corresponds to the $(w-1)$-th time step, and $y_i \in \mathbb{R}^d$. Moreover, we define $\mathbf{X}$, $\mathbf{Y}$ as the random variable representing neural signals x_i and the corresponding y_i from $\mathcal{D}$. Based on $\mathcal{D}$, we further define a labeled source domain $\mathcal{D}_S$ encompassing neural signals and labels from a single session: $\mathcal{D}_S = \{(x_1^S, y_1^S), \ldots, (x_{n_S}^S, y_{n_S}^S)\}$. Concurrently, the unlabeled target domain $\mathcal{D}_T$ includes signals from a separate session: $\mathcal{D}_T = \{x_1^T, \ldots, x_{n_T}^T\}$. For convenience, we define $\mathbf{X}^S$, $\mathbf{Y}^S$ as the random variable representing neural signals x_i^S and the corresponding y_i^S from domain $\mathcal{D}_S$, respectively. $\mathbf{X}^T$ denotes the random variable of original signals x_i^T from domain $\mathcal{D}_T$. Due to various factors, the distribution mismatch between $\mathbf{X}^S$ and $\mathbf{X}^T$ prevents the direct application of a decoder trained on $\mathcal{D}_S$ to $\mathcal{D}_T$. Our goal is to maintain decoding performance in $\mathcal{D}_T$ by stabilizing the extracted latent features via HDA and ensuring a consistent mapping to the label space.

3.2 The Framework of Hierarchical Domain Adaptation

To stabilize extracted latent features, we propose HDA based on the dynamical latent states, as shown in Fig. 1(a). Initially, we employ an unsupervised alignment strategy to align the raw signal distribution of $\mathcal{D}_T$ with that of $\mathcal{D}_S$. As outlined in Section 3.4, we found that this step effectively maintains an upper bound on latent feature deviations under control. The aligned neural signals are then provided as external inputs to a dynamical system for extracting dynamical latent dynamics.

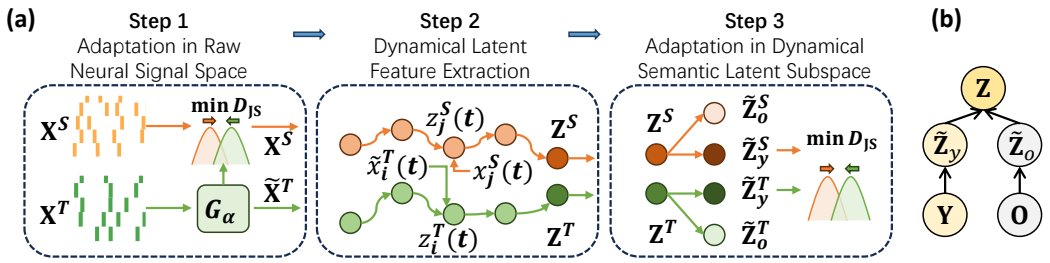

Figure 1: (a) The overall framework of HDA. (b) The model of generating $\mathbf{Z}$, which is controlled by the label variable $\mathbf{Y}$ and domain variable $\mathbf{O}$.

Subsequently, drawing inspiration from existing researches on learning interpretable and generalizable latent variables within neural signals (Zhou & Wei, 2020; Liu et al., 2021), we extract latent variables that are directly related to behavioral labels. These variables are then identified as the latent semantic features used for decoding. We further provide the dynamical systems with self-consistent alignment of these semantic features as feedback. To verify the stability of latent features, we employ Lyapunov stability (Angeli, 2002; Jiang & Wang, 2001) for a quantitative analysis of the system's stability. We have found that by optimizing parameters of the nonlinear dynamical system through HDA, the dynamical stability of latent features can be enhanced, thereby stabilizing neural decoders.

3.2.1 Adaptation in Raw Neural Signal Space

We begin by aligning the distribution of target signals with that of the source signals. The characteristic of spike signals is their capability to capture neural activities at the neuron level (Buzsáki, 2004), which ensures the sparse dependencies among different channels (Chen et al., 2010; Bighamian et al., 2019). In comparison to latent feature spaces, where spatio-temporal dependencies are more intricate, the raw neural signal spaces may offer a more direct causal relationship that facilitates the identification of units affected by drifts. This identification may help to align the distribution of the raw signals and forms the basis of our approach to enhance feature stability. Additionally, aligning input raw neural signals prior to optimizing system parameters enables a more statistically consistent input representation for our dynamical system. This approach contrasts with those that apply shared-parameter encoders to derive latent features directly from the original signals across source and target domains (Wei et al., 2023; 2021).

Considering the unique properties of biological systems, we found that probability densities based on individual samples may be more accurate to measure distribution discrepancies. This is because sufficient statistics, such as the mean, typically characterize the collective properties of random variables. However, the characteristics of individual samples are often more critical due to the common presence of outliers in biological systems (Gomez-Ramirez & Sanz, 2013). Therefore, we chose the f-divergence, which is based on probability density functions, to measure the discrepancy between distributions. However, since f-divergence is difficult to compute directly, we employed GANs to implement alignment based on f-divergences in an indirect manner. Given that naive GANs often suffer from training instability, we used LSGANs (Mao et al., 2017) based on the χ^2 divergence, which is a specific case of f-divergence. The benefits of alignment based on f-divergences are demonstrated in Fig. 7(b).

Specifically, our optimization objective is to identify a nonlinear transformation based on G that minimizes the χ^2 divergence between the distribution of $\mathbf{X}^S$ and $G(\mathbf{X}^T)$, denoted as $D_{\chi^2}\left(p(\mathbf{X}^S)||p(G(\mathbf{X}^T))\right)$. Here, this nonlinear transformation is implemented via a generator G_α from LSGAN, which is more stable than vanilla GANs, with parameters α. As mentioned in (Mao et al., 2017), it is implemented by alternately training the generator G_α, and a discriminator D_β (with parameters β) via a min-min optimization based on the least-square loss functions $\mathcal{L}_{\text{lsd}}(\beta)$ and $\mathcal{L}_{\text{lsg}}(\alpha)$, respectively:

$$\begin{aligned} \min_\beta \left\{ \mathbb{E}_{\mathbf{X}^S \sim \mathcal{D}_S}\left[(D_\beta(\mathbf{X}^S)-1)^2\right] + \mathbb{E}_{\mathbf{X}^T \sim \mathcal{D}_T}\left[\left(D_\beta(G_\alpha(\mathbf{X}^T))\right)^2\right] \right\} &= \{\mathcal{L}_{\text{lsd}}(\beta)\}, \\ \min_\alpha \mathbb{E}_{\mathbf{X}^T \sim \mathcal{D}_T}\left[\left(D_\beta(G_\alpha(\mathbf{X}^T))-1\right)^2\right] &= \{\mathcal{L}_{\text{lsg}}(\alpha)\}. \end{aligned} \tag{1}$$

After the adaptation within the original data space, we proceed with the extraction and alignment of dynamical latent features using $\mathbf{X}^S$ and $G_\alpha(\mathbf{X}^T)$.

3.2.2 Dynamical Latent Feature Extraction

To achieve real-time extraction of latent features for decoding, we employ a causal architecture (Chen et al., 2024) based on nonlinear dynamical systems to extract latent dynamics from the raw neural signals. The initial latent state of this system is based on signals that were recorded at the onset of time windows, rather than a posterior distribution of the entire trial (Karpowicz et al., 2022; Wang et al., 2023; Vermani et al., 2023). The latent state evolution is jointly driven by the current time step's externally input signal and the latent state from the previous step through a nonlinear transformation.

We utilize $x_i(t)$ as the external input to the dynamical system at time t, and the corresponding low-dimensional latent state is denoted as $z_i(t) \in \mathbb{R}^k$ (where $k < m$). The initial latent state is determined by the function g: $\mathbb{R}^m \to \mathbb{R}^k$, and the evolution of the latent state is determined by the nonlinear function f: $(\mathbb{R}^k \times \mathbb{R}^m) \to \mathbb{R}^k$. Thus, the initial state and the subsequent evolution of the nonlinear dynamical system are characterized by the following equations:

$$z_i(0) = g(x_i(0)), \quad z_i(t) = f(z_i(t-1), x_i(t))\,(t = 1, \ldots, w-1). \tag{2}$$

Considering the cumulative effect (Gros, 2010), we utilize the latent state $z_i(w-1)$ at the final time step to represent the dynamical latent feature corresponding to x_i, which is further transformed for decoding y_i. Specifically, we construct this nonlinear dynamical system using an LSTM-based (Hochreiter & Schmidhuber, 1997; D'Amico et al., 2023) network E_γ (with parameters γ). The input of E_γ is x_i, and the cell state is regarded as the latent state z_i. The cell state at the final time step, denoted as $z_i(w-1)$, is viewed as the output dynamical latent feature of E_γ: $z_i(w-1) = E_\gamma(x_i)$.

3.2.3 Adaptation in Dynamical Semantic Latent Subspace

When performing a specific task, the brain processes a wide range of information, including perception, decision-making, environmental cues, feedback, and more. For instance, task-irrelevant perceptual information and environmental feedback are also encoded within the latent dynamics. By decomposing these latent spaces to remove irrelevant components, we aim to reduce variability within the latent dynamics, thereby improving alignment of latent spaces. Specifically, inspired by previous studies mentioned in Section 2 and the high parallelism (Wässle, 2004) of brains, we hypothesize that the drifts of dynamical latent features in the target domain primarily stem from latent variables that are loosely connected to observable behavioral variables. Based on this hypothesis, we believe that constructing a semantic subspace, by extracting components of the latent space that are directly related to behavioral variables, can effectively reduce the drift of neural population dynamics. Furthermore, we have found that, compared to directly aligning latent features, performing alignment only within the semantic subspace can provide the dynamical system with a more efficient self-consistent feedback, guiding it to obtain more stable latent dynamics.

Decomposition of the Dynamical Latent Space Based on the above hypothesis, the generation of dynamical latent features $z_i(w-1)$ extracted by E_γ, denoted by the random variable $\mathbf{Z}$, is assumed to be governed by two independent variables: the domain variable $\mathbf{O}$ and the observed behavioral variable $\mathbf{Y}$. These two variables form two independent subspaces. As depicted in Fig. 1(b), we decompose the dynamical latent features $\mathbf{Z}$ into two independent components based on these two variables: one part directly encodes the semantic information, and the other part directly encodes the domain information.

Existing studies (Zhou & Wei, 2020; Liu et al., 2021; Cai et al., 2019; Wei et al., 2023) have shown that Variational Autoencoders (VAE) can solve for latent feature subspaces determined by different variables. Here, we first use the VAE's encoder to transform original dynamical features $\mathbf{Z}$ into latent features $\tilde{\mathbf{Z}}$ ($\tilde{\mathbf{Z}} \in \mathbb{R}^{\tilde{k}}$) that follow a pre-defined Gaussian distribution. Then, we divide $\tilde{\mathbf{Z}}$ into two independent parts. The first $\tilde{k}_y$ dimensions represent components directly governed by $\mathbf{Y}$, denoted as the semantic latent features $\tilde{\mathbf{Z}}_y$ ($\tilde{\mathbf{Z}}_y \in \mathbb{R}^{\tilde{k}_y}$). The remaining $\tilde{k}_o$ ($\tilde{k}_o = \tilde{k} - \tilde{k}_y$) dimensions represent components directly governed by $\mathbf{O}$, denoted as $\tilde{\mathbf{Z}}_o$ ($\tilde{\mathbf{Z}}_o \in \mathbb{R}^{\tilde{k}_o}$): $\tilde{\mathbf{Z}} = [\tilde{\mathbf{Z}}_y, \tilde{\mathbf{Z}}_o]$. Finally, $\mathbf{Z}$ is reconstructed by the VAE's decoder using both $\tilde{\mathbf{Z}}_y$ and $\tilde{\mathbf{Z}}_o$. The Evidence Lower Bound (ELBO) loss function (Kingma & Welling, 2013) is further utilized to enforce independence (Higgins et al., 2017;

Burgess et al., 2018; Higgins et al., 2018) and reconstruction constraints on the decomposed latent subspaces after transformation. The VAE's encoder, denoted as Q_ϕ (with parameters ϕ), estimates the posterior distribution $q_\phi(\tilde{\mathbf{Z}}|\mathbf{Z})$. The prior distribution of the transformed latent features $\tilde{\mathbf{Z}}$, denoted as $p_z(\tilde{\mathbf{Z}})$ is set to a multivariate Gaussian distribution by convention: $p_z(\tilde{\mathbf{Z}}) \sim \mathcal{N}(0, \mathbf{I})$. The VAE's decoder, denoted as R_θ (with parameters θ), is used to estimate $p_\theta(\mathbf{Z}|\tilde{\mathbf{Z}})$. The ELBO can then be expressed as follows:

$$\log p(\mathbf{Z}) \geqslant \mathbb{E}_{\tilde{\mathbf{Z}}\sim q_\phi(\tilde{\mathbf{Z}}|\mathbf{Z})}[p_\theta(\mathbf{Z}|\tilde{\mathbf{Z}})] - D_{\mathrm{KL}}(q_\phi(\tilde{\mathbf{Z}}|\mathbf{Z})||p_z(\tilde{\mathbf{Z}})) = -\mathcal{L}_{vae}(\theta, \phi, \gamma), \mathbf{Z} = E_\gamma(\mathbf{X}). \quad (3)$$

Here, D_{KL} represents the Kullback-Leibler (KL) divergence, for which a closed-form solution can be directly provided for Gaussian distributions. Minimizing the divergence based on a Gaussian distribution with the zero covariance enforces the independence of decomposed subspaces after transformation, consistent with the previous hypothesis. We maximize $\mathbb{E}_{\tilde{\mathbf{Z}}\sim q_\phi(\tilde{\mathbf{Z}}|\mathbf{Z})}[p_\theta(\mathbf{Z}|\tilde{\mathbf{Z}})]$ by minimizing the reconstructed Mean Squared Error (MSE) loss. Finally, we achieve the maximization of ELBO by minimizing $\mathcal{L}_{vae}$.

Further Constraints on Dynamical Semantic Latent Subspace $\mathcal{L}_{vae}$ is not sufficient to ensure the encoding information of $\tilde{\mathbf{Z}}_y$ and $\tilde{\mathbf{Z}}_o$ as hypothesized. Therefore, drawing on related work (Cai et al., 2019; Wei et al., 2023), we introduce additional terms to constrain the information encoded within latent subspaces. Let $\mathbf{Z}^S$ denote the random variable representing dynamical latent features from $\mathcal{D}_S$. The corresponding latent features transformed by Q_ϕ are $\tilde{\mathbf{Z}}^S$, which are further decomposed into semantic components $\tilde{\mathbf{Z}}^S_y$ via corresponding parameters ϕ_y of Q_ϕ, and domain-related components $\tilde{\mathbf{Z}}^S_d$ with corresponding parameters ϕ_d. The semantic latent features $\tilde{\mathbf{Z}}^S_y$ are used to decode the behavioral labels $\mathbf{Y}^S$. Similarly, for $\mathcal{D}_T$, the dynamical latent features extracted from the aligned neural signals $G_\alpha(\mathbf{X}^T)$ are $\mathbf{Z}^T$, which are decomposed into semantic components $\tilde{\mathbf{Z}}^T_y$ and domain-related components $\tilde{\mathbf{Z}}^T_o$.

First, to ensure that $\tilde{\mathbf{Z}}^S_y$ and $\tilde{\mathbf{Z}}^T_y$ directly encode semantic information without the effect from domain variables, we optimize the decoding performance of semantic features and minimize distribution discrepancies between $\tilde{\mathbf{Z}}^S_y$ and $\tilde{\mathbf{Z}}^T_y$. Specifically, we use $\mathbf{Y}^S$ and $\tilde{\mathbf{Z}}^S_y$ for supervised training of a linear decoder C_η (with parameters η), and measure the decoding performance of $\tilde{\mathbf{Z}}^S_y$ using the loss function $\mathcal{L}_y$: $\mathcal{L}_y(\gamma, \phi_y, \eta) = \|\mathbf{Y}^S - C_\eta(\tilde{\mathbf{Z}}^S_y)\|_2$, where $\tilde{\mathbf{Z}}^S_y = Q_{\phi_y}(\mathbf{Z}^S) = Q_{\phi_y}(E_\gamma(\mathbf{X}^S))$. Meanwhile, we minimize the conditional distribution discrepancy between $\tilde{\mathbf{Z}}^S_y$ and $\tilde{\mathbf{Z}}^T_y$ using the χ^2 divergence $D_{\chi^2}\left(p(\tilde{\mathbf{Z}}^S_y|\mathbf{X}^S) \,\|\, p(\tilde{\mathbf{Z}}^T_y|G_\alpha(\mathbf{X}^T))\right)$. Similar to the raw signal alignment, we alternately optimize γ, ϕ_y, and the discriminator's ($D^y_{\beta_y}$) parameters β_y based on the loss function $\mathcal{L}_{bd}(\beta_y)$, and $\mathcal{L}_{bg}(\gamma, \phi_y)$ formulated as follows:

$$\begin{aligned}
&\min_{\beta_y}\left\{\mathbb{E}_{\mathbf{X}^S\sim\mathcal{D}_S}\left[(D^y_{\beta_y}(Q_{\phi_y}(E_\gamma(\mathbf{X}^S))) - 1)^2\right] + \mathbb{E}_{\mathbf{X}^T\sim\mathcal{D}_T}\left[\left(D^y_{\beta_y}(Q_{\phi_y}(E_\gamma(G_\alpha(\mathbf{X}^T))))\right)^2\right]\right\} \\
&\min_{\gamma,\phi_y}\left\{\mathbb{E}_{\mathbf{X}^S\sim\mathcal{D}_S}\left[(D^y_{\beta_y}(Q_{\phi_y}(E_\gamma(\mathbf{X}^S))) - 1)^2\right] + \mathbb{E}_{\mathbf{X}^T\sim\mathcal{D}_T}\left[\left(D^y_{\beta_y}(Q_{\phi_y}(E_\gamma(G_\alpha(\mathbf{X}^T)))) - 1\right)^2\right]\right\}
\end{aligned} \quad (4)$$

Secondly, considering the linearity of C_η and the independence constraint between the decomposed subspaces, C_η could not work well with $\tilde{\mathbf{Z}}^S_o$ and $\tilde{\mathbf{Z}}^T_o$. Therefore, to ensure that $\tilde{\mathbf{Z}}^S_o$ and $\tilde{\mathbf{Z}}^T_o$ directly encode the domain information, we only constrain the domain relevance of $\tilde{\mathbf{Z}}^S_o$ and $\tilde{\mathbf{Z}}^T_o$. Here, we maximize the χ^2 divergence between the conditional distribution of $\tilde{\mathbf{Z}}^S_o$ and $\tilde{\mathbf{Z}}^T_o$, represented as $D_{\chi^2}\left(p(\tilde{\mathbf{Z}}^S_o|\mathbf{X}^S) \,\|\, p(\tilde{\mathbf{Z}}^T_o|G_\alpha(\mathbf{X}^T))\right)$. Similarly, we alternately optimize the parameters γ, ϕ_o, and the discriminator's ($D^o_{\beta_o}$) parameters β_o based on minimizing the loss function $\mathcal{L}_{od}(\beta_o)$, maximizing $\mathcal{L}_{og}(\gamma, \phi_o)$ as defined in Eq. (4).

3.3 Overall Learning Algorithm

During the training phase, we initially optimize G_α and D_β alternately based on $\mathcal{L}_{lsd}$ and $\mathcal{L}_{lsg}$. This step yields the aligned target neural signals, denoted as $G_\alpha(\mathbf{X}^T)$. For alignment within semantic

Algorithm 1 Hierarchical Domain Adaptation

Input: source domain $\mathcal{D}_S$; target domain $\mathcal{D}_T$;
Output: signal aligner G_α; latent dynamic extractor E_γ; VAE's encoder Q_ϕ; linear decoder C_η
Initialize G_α, D_β, E_γ, Q_ϕ, R_θ, $D^y_{\beta_y}$, $D^o_{\beta_o}$
Adaptation in Raw Neural Signal Space:
Optimize G_α and D_β alternately based on $\mathcal{L}_{\text{lsg}}(\alpha)$ and $\mathcal{L}_{\text{lsd}}(\beta)$;
Adaptation in Dynamical Semantic Latent Subspace:
for $iter = 1$ **to** n_{iter} **do**
  Sample mini-batch from $\mathcal{D}_S$ and $\mathcal{D}_T$;
  Update $D^y_{\beta_y}$ by $\mathcal{L}_{bd}(\beta_y)$; Update $D^o_{\beta_o}$ by $\mathcal{L}_{od}(\beta_o)$;
  Update E_γ, Q_ϕ, R_θ, C_η by $\mathcal{L}_{total}(\gamma, \phi, \theta, \eta)$
  $(\mathcal{L}_{total}(\gamma, \phi, \theta, \eta) = \mathcal{L}_{vae}(\gamma, \phi, \theta) + \lambda_y \mathcal{L}_y(\gamma, \phi_y, \eta) + \lambda_b \mathcal{L}_{bg}(\gamma, \phi_y) - \lambda_o \mathcal{L}_{og}(\gamma, \phi_o))$;
end for
return G_α,E_γ,Q_ϕ,C_η.

subspaces, we proceed to train the feature extractor E_γ based on dynamical systems, the VAE's encoder Q_ϕ and decoder R_θ, and the linear decoder C_η. The training is guided by a combined loss function $\mathcal{L}_{total}$: $\mathcal{L}_{total} = \mathcal{L}_{vae} + \lambda_y \mathcal{L}_y + \lambda_b \mathcal{L}_{bg} - \lambda_o \mathcal{L}_{og}$, where λ_y, λ_b, and λ_o serve as weighting factors for the respective losses. Concurrently, the discriminators $D^y_{\beta_y}$ and $D^o_{\beta_o}$ are trained based on $\mathcal{L}_{bd}$ and $\mathcal{L}_{od}$, respectively. A detailed description of the training procedure is presented in Algorithm 1.

3.4 Verification of Dynamical Feature Stability

Here, we propose a novel way to measure feature stability grounded in Lyapunov theory. First of all, for any two hidden states $z_i(t)$ and $z_j(t)$, the system is stable (Agrachev et al., 2008) if there exist functions $\beta(\|z\|, t)$ and $\gamma(\|x\|)$. For $t \geqslant 1$, the following inequality holds:

$$\|z_i(t) - z_j(t)\| = \|z_i(t, z_i(0), x_i(t)) - z_j(t, z_j(0), x_j(t))\| \leqslant \beta(\|z_i(0) - z_j(0)\|, t) + \gamma(\|x_i(t) - x_j(t)\|_\infty). \tag{5}$$

Furthermore, the stability defined above can be determined using a Lyapunov function $V(z)$. Given an equilibrium point z^* of the system, the following equations are satisfied: (1) $V(z^*) = 0$, (2) $\dot{V}(z^*) = 0$, (3) $V(z) > 0$ for all $z \neq z^*$, (4) $\dot{V}(z) < 0$ for all $z \neq z^*$. It is known that $V(z) = \frac{1}{2} z^T z$ is one of the functions that meet the conditions. However, directly calculating complex $V(z)$ can be difficult. Therefore, we used the method based on (Wolf et al., 1985) to estimate the stability of $z(t)$ based on the maximum Lyapunov exponent (MLE). The maximum Lyapunov exponent λ can be defined based on the latent state $z_i(t)$ as follows: $\lambda = \lim_{t \to \infty} \lim_{|\delta z_i(0)| \to 0} \frac{1}{t} \ln \frac{|\delta z_i(t)|}{|\delta z_i(0)|}$. A non-positive MLE often indicates the stability of dynamical systems, achieving stable dynamical latent features (Wolf et al., 1985). Here, the MLE λ of z_i is estimated to evaluate the stability of dynamical latent features extracted from $\mathcal{D}_S$ and $\mathcal{D}_T$ after adaptation. The detailed calculation of λ and the theoretical explanation of how pre-alignment enhances stability are provided in Appendix A.2.3.

4 Experiments and Results

4.1 Experimental Setup

Datasets We utilized three distinct datasets of extracellular neural recordings obtained from the primary motor cortex (M1) of non-human primates (Ma et al., 2023), as outlined below. More detailed dataset descriptions can be found in Appendix B.1.
Random-Target (RT-M). Monkey M was trained to move the cursor into a sequence of three randomly located targets on the screen within 2.0s after viewing.
Center-Out Reaching (CO-C&CO-M). Monkeys C and M were trained to use an upright handle to aim for one of eight randomized targets upon receiving an auditory cue.
Data Preprocess and Split For all datasets, we extracted trials from 'gocue time' to 'trial end'. The data was then timestamped and smoothed with a Gaussian kernel to estimate firing rate over 50 ms

bins. We utilized the session labeled with 2D cursor velocity recorded on the Day 0 as $\mathcal{D}_S$ for training. As for training $\mathcal{D}_T$, we used 80% trials of the unlabeled session collected on another day. As for tests, we employed the remaining 20% trials from this session.

Evaluation Metric The deviation between decoded and actual cursor velocity is quantified using the R^2 score. All results presented below are averaged on five distinct random seeds. Further experimental details and settings are elaborated in Appendix B.

Table 1: Comparison of average R^2 scores (%) in cross-session velocity decoding

Data	Session	LSTM	Cebra	DAF	ERDiff	NoMAD	Cycle-GAN	**HDA**	retrain
CO-M	Day 0	$74.18_{\pm 4.90}$	**79.24**$_{\pm 4.90}$	$75.24_{\pm 2.22}$	$76.31_{\pm 3.62}$	$58.29_{\pm 3.38}$	$64.99_{\pm 1.17}$	$70.86_{\pm 6.13}$	$78.38_{\pm 1.37}$
	Day 8	$-118.53_{\pm 98.70}$	$-51.92_{\pm 98.70}$	$-0.01_{\pm 0.03}$	$-75.33_{\pm 83.37}$	$57.86_{\pm 2.25}$	$71.30_{\pm 1.46}$	**76.84**$_{\pm 1.19}$	$85.92_{\pm 1.27}$
	Day 14	$-63.85_{\pm 19.96}$	$-1.77_{\pm 19.96}$	$-0.08_{\pm 0.05}$	$-102.82_{\pm 34.63}$	$63.45_{\pm 1.41}$	$67.05_{\pm 1.36}$	**71.70**$_{\pm 1.50}$	$82.92_{\pm 0.65}$
	Day 15	$-712.91_{\pm 316.04}$	$-83.24_{\pm 316.04}$	$0.03_{\pm 0.03}$	$-66.76_{\pm 86.32}$	$57.92_{\pm 0.69}$	$64.83_{\pm 1.87}$	**75.53**$_{\pm 0.91}$	$82.20_{\pm 0.99}$
	Day 22	$-88.57_{\pm 58.85}$	$-21.10_{\pm 58.85}$	$-0.08_{\pm 0.04}$	$-74.60_{\pm 60.20}$	**55.49**$_{\pm 3.82}$	$52.88_{\pm 9.75}$	$55.47_{\pm 8.34}$	$80.43_{\pm 1.26}$
	Day 24	$-39.52_{\pm 86.25}$	$-10.28_{\pm 86.25}$	$0.04_{\pm 0.02}$	$-14.52_{\pm 76.57}$	$62.52_{\pm 2.39}$	$70.13_{\pm 2.49}$	**71.32**$_{\pm 1.75}$	$86.54_{\pm 0.97}$
	Day 25	$-253.83_{\pm 270.30}$	$-64.67_{\pm 270.30}$	$0.10_{\pm 0.03}$	$-60.00_{\pm 37.44}$	$62.24_{\pm 4.23}$	$61.73_{\pm 2.34}$	**66.64**$_{\pm 3.08}$	$85.80_{\pm 1.08}$
	Day 28	$-107.64_{\pm 124.47}$	$-35.95_{\pm 124.47}$	$0.03_{\pm 0.02}$	$-46.10_{\pm 74.64}$	$48.82_{\pm 18.62}$	$66.01_{\pm 2.87}$	**71.38**$_{\pm 2.27}$	$88.33_{\pm 0.44}$
	Day 29	$-206.99_{\pm 117.46}$	$-64.32_{\pm 117.46}$	$-0.00_{\pm 0.03}$	$-42.48_{\pm 106.10}$	$61.51_{\pm 1.61}$	$60.82_{\pm 1.80}$	**64.21**$_{\pm 1.06}$	$80.92_{\pm 1.67}$
	Day 31	$-63.01_{\pm 40.94}$	$-81.41_{\pm 40.94}$	$0.04_{\pm 0.05}$	$-77.22_{\pm 91.34}$	$62.17_{\pm 2.68}$	$61.77_{\pm 0.94}$	**68.23**$_{\pm 2.17}$	$81.64_{\pm 1.08}$
	Day 32	$-417.39_{\pm 295.63}$	$-40.10_{\pm 295.63}$	$-0.06_{\pm 0.04}$	$-78.23_{\pm 59.91}$	$55.30_{\pm 4.43}$	$58.97_{\pm 2.84}$	**69.17**$_{\pm 3.39}$	$83.36_{\pm 1.36}$
RT-M	Day 0	$77.91_{\pm 1.14}$	$74.86_{\pm 1.03}$	$76.35_{\pm 2.36}$	$75.28_{\pm 1.96}$	$59.26_{\pm 3.14}$	$70.73_{\pm 3.58}$	**80.24**$_{\pm 1.97}$	$79.69_{\pm 2.68}$
	Day 1	$58.51_{\pm 4.42}$	$65.97_{\pm 2.38}$	$0.05_{\pm 0.01}$	$-130.22_{\pm 20.98}$	$57.83_{\pm 3.07}$	$66.04_{\pm 3.67}$	**69.54**$_{\pm 2.55}$	$76.27_{\pm 1.30}$
	Day 38	$-17.93_{\pm 17.68}$	$21.34_{\pm 6.71}$	$-0.31_{\pm 0.11}$	$-54.49_{\pm 32.98}$	$59.14_{\pm 1.82}$	**64.14**$_{\pm 1.77}$	$61.91_{\pm 1.04}$	$68.68_{\pm 1.97}$
	Day 39	$-104.81_{\pm 99.91}$	$-36.86_{\pm 25.62}$	$-0.14_{\pm 0.12}$	$-38.28_{\pm 43.94}$	$58.13_{\pm 1.58}$	**69.86**$_{\pm 3.89}$	$68.78_{\pm 1.68}$	$78.03_{\pm 1.28}$
	Day 40	$-14.81_{\pm 47.15}$	$2.63_{\pm 20.16}$	$0.06_{\pm 0.04}$	$-31.41_{\pm 39.80}$	$61.27_{\pm 1.51}$	$66.01_{\pm 3.75}$	**68.77**$_{\pm 3.52}$	$83.55_{\pm 1.52}$
	Day 52	$6.10_{\pm 18.37}$	$30.50_{\pm 6.94}$	$-0.16_{\pm 0.05}$	$-110.11_{\pm 46.32}$	$53.41_{\pm 4.55}$	$47.74_{\pm 7.58}$	**56.31**$_{\pm 2.23}$	$61.36_{\pm 3.15}$
	Day 53	$-47.05_{\pm 63.72}$	$42.33_{\pm 4.84}$	$-0.36_{\pm 0.05}$	$-112.86_{\pm 31.80}$	$53.65_{\pm 2.69}$	$61.96_{\pm 6.85}$	$68.49_{\pm 1.15}$	$76.92_{\pm 1.96}$
	Day 67	$-158.42_{\pm 104.75}$	$25.09_{\pm 13.79}$	$-0.30_{\pm 0.07}$	$-81.18_{\pm 77.17}$	$58.12_{\pm 1.73}$	$43.76_{\pm 7.69}$	**64.79**$_{\pm 1.54}$	$74.83_{\pm 0.97}$
	Day 69	$-101.08_{\pm 32.58}$	$-38.82_{\pm 29.41}$	$-0.22_{\pm 0.04}$	$-168.49_{\pm 35.82}$	$50.17_{\pm 4.60}$	$34.54_{\pm 6.80}$	**53.94**$_{\pm 2.31}$	$66.07_{\pm 2.27}$
	Day 77	$-280.39_{\pm 104.05}$	$-53.79_{\pm 21.04}$	$0.01_{\pm 0.00}$	$-63.76_{\pm 54.85}$	$52.39_{\pm 2.32}$	$33.38_{\pm 7.72}$	**56.90**$_{\pm 2.28}$	$60.33_{\pm 1.62}$
	Day 79	$-184.05_{\pm 76.77}$	$-47.01_{\pm 13.77}$	$-0.13_{\pm 0.05}$	$-46.66_{\pm 49.72}$	$55.01_{\pm 3.05}$	$36.83_{\pm 12.53}$	**58.35**$_{\pm 2.40}$	$76.56_{\pm 0.92}$

4.2 Comparative Study

Baselines We used the following methods as baselines, more details are shown in Appendix B.2:
LSTM(Hochreiter & Schmidhuber, 1997): We employed an unaligned LSTM as the decoder to evaluate the efficacy of alignment.
CEBRA(Schneider et al., 2023): CEBRA is a machine learning method that compresses time series to uncover hidden structures, demonstrating broad generalizability across various datasets and conditions.
DAF(Jin et al., 2022) DAF leverages an attention mechanism to extract domain-invariant features while retaining domain-specific details through a shared module, domain discriminator, and private modules.
ERDiff(Wang et al., 2023) ERDiff utilizes diffusion models to meticulously reconstruct spatio-temporal structures and seamlessly aligns them with the latent dynamics extracted from VAEs.
NoMAD(Karpowicz et al., 2022): NoMAD achieves signal alignment in neural manifolds by capturing the latent dynamics of neural population activities via LFADS (Pandarinath et al., 2018).
Cycle-GAN (Ma et al., 2023): This research employs Cycle-GAN to align the distributions of full-dimensional neural recordings at each time step.

Results We conducted quantitative comparisons using average R^2 of target domains. Day 0 corresponds to $\mathcal{D}_S$, while the other sessions represent $\mathcal{D}_T$. Results for CO-M and RT-M are presented in Table 1, with results for the CO-C dataset shown in Table 8. Our method consistently outperforms others across most sessions of the selected datasets. The unaligned LSTM and CEBRA designed for generalizable representations frequently fail over extended periods, demonstrating the need for distribution alignment to stabilize decoding performance. Regarding existing UDA-based alignment methods, HDA achieves over 6.00% and 10.00% higher R^2 on average compared to Cycle-GAN for CO-M and RT-M, respectively. Compared to NoMAD, we achieve over 10.00% and 8.00% higher on average. It can also be seen that HDA achieves much better performance than ERDiff and DAF. In addition, we visualized reach trajectories of CO-M integrated from the decoded cursor velocity. As shown in Appendix C.2, we observe that HDA yields more precise trajectories.

In addition, HDA consistently demonstrates effective performance across various source days. To validate this, we conducted additional experiments across all sessions, with the overall performance

illustrated in Fig. 6(a). We also conducted experiments using Day 0 as the source session, with source and target training ratios set at $0.1, 0.2, 0.4, 0.6$. As illustrated in Fig. 6(b), HDA shows effective performance even with a relatively small number of trials.

As for feature stability, we compared our MLE with those from ERDiff and NoMAD, as MLE can only be derived from sequential models. Fig. 7(d) shows the average MLE for each target session and their overall averages. A non-positive MLE typically indicates the stability of dynamical systems. We observed that HDA generally achieves the most stable latent space. Additionally, ERDiff exhibits greater instability compared to NoMAD, consistent with the R^2 score performance shown in Table 1. We further visualized latent trajectories of latent features $z(t)$ on Day22, Day24 from CO-M. Presented Fig. 3(c), we found that the divergent curves evolve over time to gradually approach each other. This indicates that the trajectories exhibit characteristics of Lyapunov stability.

Computational Efficiency We compared our efficiency with that of the baselines. Presented in Table 2, we found that HDA exhibits a similar parameter count to ERDiff and NoMAD, with greater time efficiency.

Table 2: List of computational efficiency with different methods

Method	DAF	ERDiff	Cycle-GAN	NoMAD	**HDA**
Parameter Number (M)	0.06	0.04	0.03	0.05	**0.04**
Training Time per Epoch (s)	0.15	0.28	0.02	3.77	**0.06**

4.3 Ablation Study

We conducted an ablation study to confirm the effectiveness of HDA. Performance was evaluated based on the cross-session decoding and the stability of extracted dynamical latent features.

Evaluation of Main Components We specifically compared our full method against variations lacking raw neural signal adaptation (HDA-r), latent space decomposition (HDA-d), and semantic subspace adaptation (HDA-s). The cross-session decoding performance was validated on CO-C, CO-M, and RT-M datasets, with the results presented in Fig. 2(a). HDA performs the best, demonstrating the effectiveness of our main modules for stabilizing latent features. We observe that R^2 of HDA-r decreases the most, indicating that this step forms the foundation for better latent space alignment. Furthermore, HDA-d yields lower R^2, highlighting the advantages of latent space decomposition for more stable semantic features. Without the semantic subspace alignment, HDA-s performs second best on average, underscoring the necessity of further alignment within the decomposed subspace.

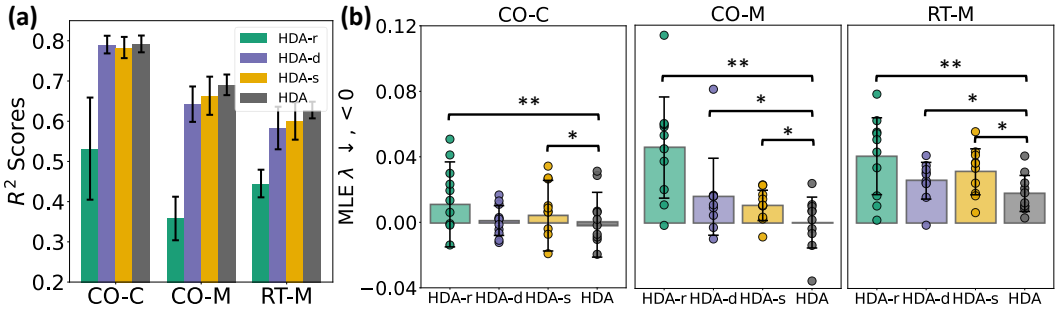

Figure 2: (a) R^2 scores for cross-session decoding, achieved by the variants HDA-r, HDA-d, HDA-s, and HDA across CO-C, CO-M, and RT-M datasets. (b) Comparison of the maximum Lyapunov exponent λ with different methods on CO-C, CO-M, and RT-M datasets. Dots in various colors represent average MLE from an individual session. The symbols '*' and '**' denote significant p-values from paired t-tests, indicating $p < 5 \times 10^{-2}$ and $p < 1 \times 10^{-2}$, respectively.

Evaluation of Each Loss Term We further conducted a ablation study on each loss term. Specifically, we evaluated on $\mathcal{L}_y$, $\mathcal{L}_b$, and $\mathcal{L}_o$ with different weights λ_y, λ_b, and λ_o. The average R^2 scores (%) are listed below. As shown in Table 3, it can be seen that all these loss terms are necessary for optimizing the model performance.

Stability Validation of Dynamical Latent Features To validate the stability of dynamical latent features after our adaptation, we evaluated the dynamical system's stability based on the maximum Lyapunov exponent λ as mentioned in Section 3.4. A non-positive MLE often indicates the stability of dynamical systems, achieving stable dynamical latent features (Wolf et al., 1985). More background information is shown in Appendix A.2. As depicted in Fig. 2(b), consistent with previous findings on the decoding stability, HDA-r emerges as the most unstable. This underscores the stabilizing effect of consistent input neural signals on latent features. Furthermore, semantic subspace alignment effectively enhances the dynamical stability of latent features, compared to HDA-d and HDA-s.

Furthermore, empirical results of pre-alignment are shown in Fig. 3(a) and (b). We found that R^2 and MLE demonstrate an upward trend with an increasing number of pre-alignment epochs. In addition, we also observed that the latent space alignment can enhance its dynamical stability. As depicted in Fig. 3(e), MLE converges to a non-positive value with an increasing number of alignment epochs. We conducted additional experiments on R^2 of test signals from the target session during latent space alignment as well. As depicted in Fig. 3(d), the curves on CO-M, and RT-M show successful convergences, indicating the training stability of HDA.

Table 3: Full ablation studies on different loss terms. Each weight is adjusted while keeping the other two fixed at 0.1.

Data	λ_o				λ_b				λ_y			
	0	0.1	1	2	0	0.1	1	2	0	0.1	1	2
CO-C	$79.24_{\pm 1.90}$	$79.60_{\pm 1.68}$	$\mathbf{79.92}_{\pm 1.94}$	$79.84_{\pm 1.54}$	$79.02_{\pm 2.42}$	$\mathbf{81.17}_{\pm 1.98}$	$79.53_{\pm 3.04}$	$78.99_{\pm 3.12}$	$-1.46_{\pm 11.23}$	$\mathbf{80.39}_{\pm 2.20}$	$79.53_{\pm 2.90}$	$78.57_{\pm 3.00}$
CO-M	$67.68_{\pm 4.31}$	$67.93_{\pm 3.50}$	$\mathbf{69.21}_{\pm 2.89}$	$68.38_{\pm 4.24}$	$66.75_{\pm 4.25}$	$69.49_{\pm 3.81}$	$69.10_{\pm 3.92}$	$\mathbf{69.60}_{\pm 3.21}$	$8.85_{\pm 2.12}$	$\mathbf{67.50}_{\pm 4.35}$	$67.38_{\pm 4.14}$	$66.36_{\pm 4.86}$
RT-M	$65.02_{\pm 4.92}$	$65.12_{\pm 3.50}$	$64.37_{\pm 2.06}$	$\mathbf{65.63}_{\pm 5.29}$	$60.06_{\pm 2.07}$	$\mathbf{63.86}_{\pm 4.28}$	$63.61_{\pm 4.49}$	$61.90_{\pm 4.85}$	$-1.02_{\pm 0.72}$	$\mathbf{62.16}_{\pm 4.14}$	$61.55_{\pm 3.97}$	$61.54_{\pm 4.49}$

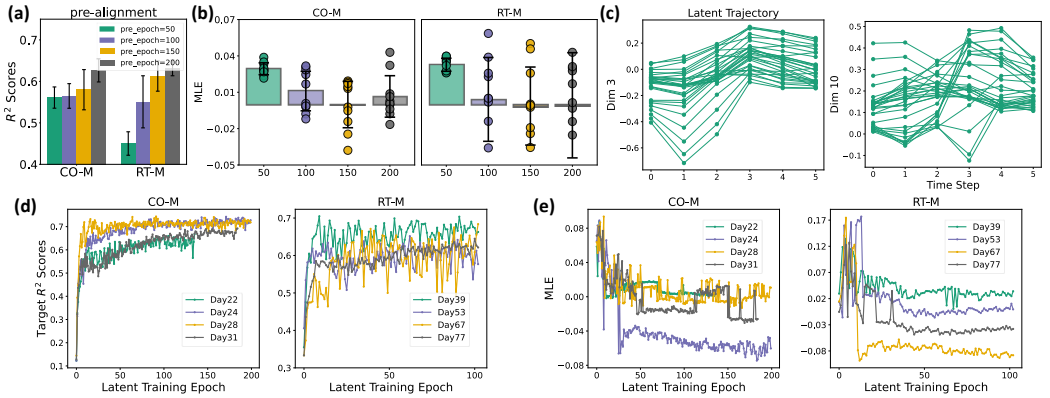

Figure 3: R^2 scores (a) and Maximum Lyapunov Exponent (MLE) (b) with varying pre-alignment training epochs (50, 100, 150, and 200) before the optimization. (c) Latent state trajectory visualizations ($z(t)$ from dimension 3 and 10) from CO-M on Day22 and 24. R^2 scores (d) and MLE (e) of test target trials during the latent space alignment on CO-M, and RT-M datasets, respectively.

5 Conclusions and Limitations

In this study, we addressed the challenge of the decoding instability in BCIs caused by the variability of neural signals over time. We present a novel hierarchical domain adaptation (HDA) that focuses on neural dynamics. This framework utilizes causal architecture to extract dynamical latent features, and improves the feature stability based on the self-consistent alignment. The experimental analysis, supported by Lyapunov stability theory, demonstrate that our HDA can effectively improve the stability of dynamical systems, allowing for high-performance behavioral decoding for non-stationary neural signals. Our work successfully addressed the challenge of non-stationary neural signals, thereby potentially advancing the reliability of BCIs in chronic applications.

The limitations of this study are as follows. For scenarios with abundant data, HDA requires further verification to extract more generalizable features, enabling zero-shot inference. Additionally, the extension of HDA to other datasets involving humans needs validation.

REFERENCES

Andrei A Agrachev, A Stephen Morse, Eduardo D Sontag, Héctor J Sussmann, Vadim I Utkin, and Eduardo D Sontag. Input to state stability: Basic concepts and results. *Nonlinear and optimal control theory: lectures given at the CIME summer school held in Cetraro, Italy June 19–29, 2004*, pp. 163–220, 2008.

David Angeli. A lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

Vivek R Athalye, Karunesh Ganguly, Rui M Costa, and Jose M Carmena. Emergence of coordinated neural dynamics underlies neuroprosthetic learning and skillful control. *Neuron*, 93(4):955–970, 2017.

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael Mendelson, Blake Richards, Matthew Perich, Guillaume Lajoie, and Eva Dyer. A unified, scalable framework for neural population decoding. *Advances in Neural Information Processing Systems*, 36, 2023.

Randall D Beer. Dynamical approaches to cognitive science. *Trends in cognitive sciences*, 4(3): 91–99, 2000.

Ramin Bighamian, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Sparse model-based estimation of functional dependence in high-dimensional field and spike multiscale networks. *Journal of neural engineering*, 16(5):056022, 2019.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

György Buzsáki. Large-scale recording of neuronal ensembles. *Nature neuroscience*, 7(5):446–451, 2004.

Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. In *IJCAI: proceedings of the conference*, volume 2019, pp. 2060. NIH Public Access, 2019.

Ujwal Chaudhary, Niels Birbaumer, and Ander Ramos-Murguialday. Brain–computer interfaces for communication and rehabilitation. *Nature Reviews Neurology*, 12(9):513–525, 2016.

Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3521–3528, 2020.

Xupeng Chen, Ran Wang, Amirhossein Khalilian-Gourtani, Leyao Yu, Patricia Dugan, Daniel Friedman, Werner Doyle, Orrin Devinsky, Yao Wang, and Adeen Flinker. A neural speech decoding framework leveraging deep learning and speech synthesis. *Nature Machine Intelligence*, pp. 1–14, 2024.

Zhe Chen, David F Putrino, Soumya Ghosh, Riccardo Barbieri, and Emery N Brown. Statistical inference for assessing functional connectivity of neuronal ensembles with sparse spiking data. *IEEE transactions on neural systems and rehabilitation engineering*, 19(2):121–135, 2010.

Joseph Y Cheng, Hanlin Goh, Kaan Dogrusoz, Oncel Tuzel, and Erdrin Azemi. Subject-aware contrastive learning for biosignals. *arXiv preprint arXiv:2007.04871*, 2020.

Cheol Jun Cho, Edward Chang, and Gopala Anumanchipalli. Neural latent aligner: cross-trial alignment for learning representations of complex, naturalistic neural data. In *International Conference on Machine Learning*, pp. 5661–5676. PMLR, 2023.

Jennifer L Collinger, Robert A Gaunt, and Andrew B Schwartz. Progress towards restoring upper limb movement and sensation through intracortical brain-computer interfaces. *Current Opinion in Biomedical Engineering*, 8:84–92, 2018.

William D'Amico, Alessio La Bella, and Marcello Farina. An incremental input-to-state stability condition for a class of recurrent neural networks. *IEEE Transactions on Automatic Control*, 2023.

Alan D Degenhart, William E Bishop, Emily R Oby, Elizabeth C Tyler-Kabara, Steven M Chase, Aaron P Batista, and Byron M Yu. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature biomedical engineering*, 4(7):672–685, 2020.

Rainer Engelken, Fred Wolf, and Larry F Abbott. Lyapunov spectra of chaotic recurrent neural networks. *Physical Review Research*, 5(4):043044, 2023.

Tao Fang, Qian Zheng, Yu Qi, and Gang Pan. Extracting semantic-dynamic features for long-term stable brain computer interface. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5965–5973, 2023.

Ali Farshchian, Juan A Gallego, Joseph P Cohen, Yoshua Bengio, Lee E Miller, and Sara A Solla. Adversarial domain adaptation for stable brain-machine interfaces. In *International Conference on Learning Representations*, 2018.

Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

Juan A Gallego, Matthew G Perich, Raeed H Chowdhury, Sara A Solla, and Lee E Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, 23 (2):260–270, 2020.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Bo S Goh. Global stability in many-species systems. *The American Naturalist*, 111(977):135–143, 1977.

Jaime Gomez-Ramirez and Ricardo Sanz. On the limitations of standard statistical modeling in biological systems: a full bayesian approach for biology. *Progress in biophysics and molecular biology*, 113(1):80–91, 2013.

Claudius Gros. *Complex and adaptive dynamical systems*, volume 990. Springer, 2010.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Li Hu, André Mouraux, Yong Hu, and Gian Domenico Iannetti. A novel approach for enhancing the signal-to-noise ratio and detecting automatically event-related potentials (erps) in single trials. *Neuroimage*, 50(1):99–111, 2010.

Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2): 328–373, 2013.

Hidehiko K Inagaki, Lorenzo Fontolan, Sandro Romani, and Karel Svoboda. Discrete attractor dynamics underlies persistent activity in the frontal cortex. *Nature*, 566(7743):212–217, 2019.

Zhong-Ping Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

Zhong-Ping Jiang, Iven MY Mareels, and Yuan Wang. A lyapunov formulation of the nonlinear small-gain theorem for interconnected iss systems. *Automatica*, 32(8):1211–1215, 1996.

Xiaoyong Jin, Youngsuk Park, Danielle Maddix, Hao Wang, and Yuyang Wang. Domain adaptation for time series forecasting via attention sharing. In *International Conference on Machine Learning*, pp. 10280–10297. PMLR, 2022.

Ce Ju and Cuntai Guan. Deep optimal transport for domain adaptation on spd manifolds. *arXiv preprint arXiv:2201.05745*, 2022.

Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 10462–10475. PMLR, 2022.

Brianna M Karpowicz, Yahia H Ali, Lahiru N Wimalasena, Andrew R Sedler, Mohammad Reza Keshtkaran, Kevin Bodkin, Xuan Ma, Lee E Miller, and Chethan Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *bioRxiv*, pp. 2022–04, 2022.

Mikail Khona and Ila R Fiete. Attractor and integrator networks in the brain. *Nature Reviews Neuroscience*, 23(12):744–766, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. Spd domain-specific batch normalization to crack interpretable unsupervised domain adaptation in eeg. *Advances in Neural Information Processing Systems*, 35:6219–6235, 2022.

Trung Le and Eli Shlizerman. Stndt: Modeling neural population activity with spatiotemporal transformers. *Advances in Neural Information Processing Systems*, 35:17926–17939, 2022.

Seunghun Lee, Sunghyun Cho, and Sunghoon Im. Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15252–15261, 2021.

Yuening Li, Zhengzhang Chen, Daochen Zha, Mengnan Du, Jingchao Ni, Denghui Zhang, Haifeng Chen, and Xia Hu. Towards learning disentangled representations for time series. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3270–3278, 2022.

Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. *Advances in neural information processing systems*, 34:10587–10599, 2021.

Ran Liu, Mehdi Azabou, Max Dabagia, Jingyun Xiao, and Eva Dyer. Seeing the forest and the tree: Building representations of both individual and collective dynamics with transformers. *Advances in neural information processing systems*, 35:2377–2391, 2022.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017a.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017b.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8046–8056, 2022.

Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

Xuan Ma, Fabio Rizzoglio, Kevin L Bodkin, Eric Perreault, Lee E Miller, and Ann Kennedy. Using adversarial networks to extend brain computer interface decoding accuracy over time. *elife*, 12: e84296, 2023.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.

Sean L Metzger, Kaylo T Littlejohn, Alexander B Silva, David A Moses, Margaret P Seaton, Ran Wang, Maximilian E Dougherty, Jessie R Liu, Peter Wu, Michael A Berger, et al. A high-performance neuroprosthesis for speech decoding and avatar control. *Nature*, 620(7976): 1037–1046, 2023.

Rufus Mitchell-Heggs, Seigfred Prado, Giuseppe P Gava, Mary Ann Go, and Simon R Schultz. Neural manifold analysis of brain circuit dynamics in health and disease. *Journal of computational neuroscience*, 51(1):1–21, 2023.

Chethan Pandarinath and Sliman J Bensmaia. The science and engineering behind sensitized brain-controlled bionic hands. *Physiological Reviews*, 102(2):551–604, 2022.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Archit Parnami and Minwoo Lee. Learning from few examples: A summary of approaches to few-shot learning. *arXiv preprint arXiv:2203.04291*, 2022.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019a.

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pp. 5102–5112. PMLR, 2019b.

János A Perge, Mark L Homer, Wasim Q Malik, Sydney Cash, Emad Eskandar, Gerhard Friehs, John P Donoghue, and Leigh R Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of neural engineering*, 10(3):036004, 2013.

Antônio H Ribeiro, Koen Tiels, Luis A Aguirre, and Thomas Schön. Beyond exploding and vanishing gradients: analysing rnn training using attractors and smoothness. In *International conference on artificial intelligence and statistics*, pp. 2370–2380. PMLR, 2020.

Mostafa Safaie, Joanna C Chang, Junchol Park, Lee E Miller, Joshua T Dudman, Matthew G Perich, and Juan A Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, pp. 1–7, 2023.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8503–8512, 2018.

Gopal Santhanam, Michael D Linderman, Vikash Gilja, Afsheen Afshar, Stephen I Ryu, Teresa H Meng, and Krishna V Shenoy. Hermesb: a continuous neural recording system for freely behaving primates. *IEEE Transactions on Biomedical Engineering*, 54(11):2037–2050, 2007.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Wilson Truccolo, Gerhard M Friehs, John P Donoghue, and Leigh R Hochberg. Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia. *Journal of Neuroscience*, 28 (5):1163–1178, 2008.

Ayesha Vermani, Il Memming Park, and Josue Nassar. Leveraging generative models for unsupervised alignment of neural time series data. In *The Twelfth International Conference on Learning Representations*, 2023.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Yule Wang, Zijing Wu, Chengrui Li, and Anqi Wu. Extraction and recovery of spatio-temporal structure in latent dynamics alignment with diffusion model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=AuXd54odxm.

Heinz Wässle. Parallel processing in the mammalian retina. *Nature Reviews Neuroscience*, 5(10): 747–757, 2004.

Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. Toalign: task-oriented alignment for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:13834–13846, 2021.

Pengfei Wei, Lingdong Kong, Xinghua Qu, Yi Ren, Jing Jiang, Xiang Yin, et al. Unsupervised video domain adaptation for action recognition: A disentanglement perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Francis R Willett, Donald T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. High-performance brain-to-text communication via handwriting. *Nature*, 593(7858): 249–254, 2021.

Francis R Willett, Erin M Kunz, Chaofei Fan, Donald T Avansino, Guy H Wilson, Eun Young Choi, Foram Kamdar, Matthew F Glasser, Leigh R Hochberg, Shaul Druckmann, et al. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036, 2023.

Lahiru N Wimalasena, Lee E Miller, and Chethan Pandarinath. From unstable input to robust output. *Nature Biomedical Engineering*, 4(7):665–667, 2020.

Kevin Woeppel, Christopher Hughes, Angelica J Herrera, James R Eles, Elizabeth C Tyler-Kabara, Robert A Gaunt, Jennifer L Collinger, and Xinyan Tracy Cui. Explant analysis of utah electrode arrays implanted in human cortex for brain-computer-interfaces. *Frontiers in Bioengineering and Biotechnology*, 9:1137, 2021.

Alan Wolf, Jack B Swift, Harry L Swinney, and John A Vastano. Determining lyapunov exponents from a time series. *Physica D: nonlinear phenomena*, 16(3):285–317, 1985.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *Advances in Neural Information Processing Systems*, 33: 7234–7247, 2020.

A Hierarchical Domain Adaptation

A.1 Architecture Details

We present the detailed architecture of our main modules as follows. The input neural signals have the shape of (Batch size=256, Window size=w, Number of channels=L). The latent dimensions of $\tilde{Z}_y$ and $\tilde{Z}_o$ are denoted as k_y, the dimension of latent states extracted by the nonlinear dynamical system as k_h. The dropout value is represented as v_d. The architectures of E_γ, Q_ϕ, R_θ, $D^y_{\beta_y}$, and $D^o_{\beta_o}$ can be seen in Table 4.

Table 4: Detailed Architectures of Modules

E_γ	LSTM(L, k_h)
Q_ϕ	FC(k_h, $2k_y$, v_d)×2
$D^y_{\beta_y}$	FC(k_y, k_y, v_d), ReLU(), FC(k_y, k_y, v_d), ReLU(), FC(k_y, 1), Sigmoid()
$D^o_{\beta_o}$	FC(k_y, k_y, v_d), ReLU(), FC(k_y, k_y, v_d), ReLU(), FC(k_y, 1), Sigmoid()

Here, we use the term FC to refer to fully connected layers, LSTM to represent Long Short-Term Memory layers, and ReLU and Sigmoid to denote the corresponding activation functions.

Moreover, default dimensions k_y, k_h, and value v_d mentioned above are configured as shown in Table 5 according to different datasets.

Table 5: Default Value Setup on Different Datasets

	k_y	k_h	v_d
CO-C	32	32	0.01
CO-M	32	32	0.01
RT-M	32	32	0.01

A.2 Lyapunov Stability Theory

A.2.1 Related Work

Lyapunov stability examines the stability of latent state trajectories within a dynamical system when its initial conditions or external inputs experience perturbations (Jiang & Wang, 2001). This concept, introduced by Lyapunov (Lyapunov, 1992), has been widely utilized in the stability analysis of various dynamical systems, including discrete linear systems (Goh, 1977) and nonlinear non-autonomous systems (Jiang et al., 1996). With the advent of deep learning, neural networks such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) are frequently employed to model complex nonlinear dynamical systems, with their hidden variables corresponding to the system's latent states. Some studies have integrated deep learning with Lyapunov stability to explore stability during network training (Engelken et al., 2023) and the network robustness (Ribeiro et al., 2020).

In the field of neuroscience, dynamical systems are frequently employed to model cognitive processes (Beer, 2000) and neural activities within the motor cortex (Ijspeert et al., 2013). The stability theory has also been leveraged to analyze these neural activities. For example, studies have identified that discrete attractors in the prefrontal cortex (Inagaki et al., 2019) are related to Lyapunov stability and lay the foundation for the working memory performance of animals undertaking delayed alternation tasks. Inspired by these insights and considering the presence of stable dynamical systems within the brain, we integrated the concept of Lyapunov stability with the process of extracting stable latent features from non-stationary neural signals.

A.2.2 Rationale for Validation via Lyapunov Stability

In biological systems, similar behaviors often manifest analogous activities within neural populations. However, neural signals from the target domain may deviate from expected similarities with the source domain due to various factors. These stochastic factors can cause drifts in any stochastic dimensions. Here, we argue that Lyapunov stability effectively characterizes the stability of extracted latent features against random perturbations in original signals. That is to say, the enhancement in Lyapunov stability of the dynamical system indicates the stabilization of dynamical latent features. Therefore, we utilize Lyapunov stability as a tool for a quantitative representation of the system's stability to validate the stability of dynamical latent features.

A.2.3 More explanations on maximum Lyapunov exponent (MLE)

Here, we give a theoretical explanation on how the pre-alignment of HDA improves dynamical stability. According to the definition in (Jiang & Wang, 2001), stability measures the distance between any two hidden states at time t, denoted as $z_i(t)$ and $z_j(t)$. Since these states are extracted using LSTMs, their distance can be expressed through the Lipschitz continuity of the activation layers:

$$\|z_i(t) - z_j(t)\| \leqslant \mathbf{K}_z + \mathbf{K}_i \mathbf{L}_a \|W_c\| \|x_i(t) - x_j(t)\|, \tag{6}$$

where $\mathbf{K}_z$ and $\mathbf{K}_x$ are constants independent of $\|x_i(t) - x_j(t)\|$, and $\mathbf{L}_a$ is the Lipschitz constant of activation functions. Thus, the pre-alignment, which helps minimize $\|x_i(t) - x_j(t)\|$, aids in controlling the upper bound of $\|z_i(t) - z_j(t)\|$, enhancing the efficiency of latent feature alignment.

The stability defined in (Jiang & Wang, 2001) can be determined based on (Wolf et al., 1985) to estimate the stability of $z(t)$ as follows:

Step 1:
Select N sample points, denoted one as $z_1(t_0)$, find j such that $j = \arg\min_k \|z_1(t_0) - z_k(t_0)\|$, and let $L_0(t_0) = \|z_1(t_0) - z_j(t_0)\|$.

Step 2:
Find t_i, for a given constant ϵ, such that $t_0 \leqslant t < t_i$, $L_0(t) \leqslant \epsilon$; $L_0(t_i) > \epsilon$. Let $L_0' = L_0(t_i)$. Continue with $z_1(t_i)$ as the next sample point following Step 1.

Step 3:
The maximum Lyapunov exponent(MLE) λ is approximately as follows:

$$\lambda \approx \frac{1}{N\Delta t} \sum_{s=1}^{M} \log_2 \left(\frac{L_0'}{L_0(t_0)} \right),$$

where Δt is the time step interval and M is the number of steps in a single orbit.

B Experimental Details

B.1 Dataset Description

CO-C&CO-M. Monkeys C and M performed a center-out (CO) reaching task while grasping an upright handle. Monkey C used its right hand and Monkey M its left. Each trial began with the monkey moving its hand to the workspace center. Following a random wait, one of eight equally-spaced outer targets in a circular arrangement appeared. The monkey then held through a variable delay until an auditory go cue. To receive liquid reward, the monkey had to reach the outer target within 1.0 second and hold for 0.5 seconds.

RT-M. Monkey M also performed a random-target (RT) task, reaching to sequences of three targets presented in random screen locations to complete a trial. The RT task used the same apparatus as the CO reaching task. Each trial began with the monkey moving its hand to the workspace center. Three targets were then sequentially presented, and the monkey had to move the cursor into each within 2.0 seconds of viewing. As the target positions were randomized, the cursor trajectory took on a "random-target" form each trial.

Detailed Preprocess Process. For all datasets, we extracted trials from 'gocue time' to 'trial end' and preprocessed the neural signals by digitizing, bandpass filtering (250-5000 Hz), and spike detection

based on root-mean square activity thresholds. The data was then timestamped and smoothed with a Gaussian kernel to estimate firing rate over 50 ms bins.

B.2 Baseline Implementation

CEBRA. CEBRA is a sophisticated machine-learning technique developed for the analysis and compression of time series data, particularly enhancing the study of behavioral and neural data. This method is capable of uncovering hidden structures within data variability and has been successfully applied to decode activity in the mouse brain's visual cortex, even reconstructing what a subject has viewed. The code is available from https://github.com/AdaptiveMotorControlLab/cebra.

DAF. The Domain Adaptation Forecaster (DAF) utilizes abundant data from a relevant source domain to enhance performance in a target domain with limited data. DAF employs an attention-based shared module with a domain discriminator and private modules for each domain, promoting the extraction of domain-invariant latent features while simultaneously retraining domain-specific features. Our approach effectively aligns keys from the source and target domains, allowing for effective knowledge transfer despite differing characteristics. Extensive experiments show that DAF outperforms state-of-the-art methods on both synthetic and real-world datasets, and ablation studies confirm the efficacy of our design choices.

ERDiff. This work proposes leveraging diffusion models to first extract latent dynamic structures in the source domain, then recover them well in the target domain through maximum likelihood alignment. Empirical evaluation on synthetic and neural recording datasets demonstrates this approach outperforms others by better preserving latent dynamic structures longitudinally and between individuals. We implement this based on the openly available code at https://github.com/yulewang97/ERDiff.

NoMAD. NoMAD leverages the latent manifold structure inherent in neural population activity to establish a stable link between brain activity and motor behavior. It demonstrates the ability to provide accurate and highly stable behavioral decoding over extended periods, eliminating the need for supervised recalibration. In this study, we implemented NoMAD using the LFADS code available at https://github.com/arsedler9/lfads-torch/tree/main. As a result, there may be some deviations from the original implementation.

Cycle-GAN. This work proposes utilizing Cycle-GAN to align the distributions of full-dimensional neural recordings and stabilize the original decoding model without requiring recalibration. Through evaluating Cycle-GAN and a related approach (ADAN) on multiple monkey and task datasets, Cycle-GAN demonstrated superior performance for robustly maintaining BCI accuracy longitudinally without additional training. As the study utilizes the same datasets, we directly implement its openly available code from https://github.com/limblab/adversarial_BCI.

B.3 Training Details

The main configurations for model training included the learning rate, weight decay parameters of the Adam optimizer, batch sizes, number of training epochs, and GPU hardware. Details of these hyperparameters are provided in Table 6.

Table 6: Detailed Training Setup

	Learning Rate	Weight Decay	Epoch Number	Batch Size	GPU
CO-C	2e-3	1e-5	2500	256	NVIDIA 3080Ti
CO-M	2e-3	5e-7	2000	256	NVIDIA 3080Ti
RT-M	2e-3	5e-7	3000	256	NVIDIA 3080Ti

Main hyper-parameters, the signal window size (w), and the weights balancing terms in the final loss function ($\lambda_{y,b,o}$) are set as shown in Table 7.

B.4 Detailed Test Procedure

Specifically, during the test phase, we employed neural signals $\mathbf{X}^T$ from the target domain, which were not leveraged during the training phase, to evaluate the efficacy of our alignment

Table 7: Hyper-parameter Setup

	w	λ_y	λ_b	λ_o
CO-C	6	1	1e-2	1
CO-M	6	1	1e-2	1
RT-M	5	1	1e-2	1

approach. This evaluation is based on the decoding performance, as represented by $\mathcal{L}_y$: $\mathcal{L}_y = \left\| \mathbf{Y}^T - C_\eta \left(Q_{\phi_y} \left(E_\gamma \left(G_\alpha \left(\mathbf{X}^T \right) \right) \right) \right) \right\|_2$, where $\mathbf{Y}^T$ signifies the actual reaching velocity corresponding to $\mathbf{X}^T$.

C Additional Results

C.1 Comparative Study

The comprehensive results of average R^2 scores for cross-session velocity decoding on the CO-C dataset are detailed in Table 8.

Table 8: Average R^2 Scores (%) of Cross-session Velocity Decoding

Data	Session	LSTM	Cebra	DAF	ERDiff	NoMAD	Cycle-GAN	**HDA**	retrain
CO-C	Day 0	$86.65_{\pm 1.18}$	$88.30_{\pm 1.66}$	$86.25_{\pm 0.87}$	$\mathbf{88.52}_{\pm 0.72}$	$31.99_{\pm 9.45}$	$78.29_{\pm 1.93}$	$86.66_{\pm 0.29}$	$86.66_{\pm 0.29}$
	Day 1	$5.04_{\pm 27.90}$	$15.41_{\pm 14.89}$	$-6.40_{\pm 4.97}$	$-7.59_{\pm 12.30}$	$44.39_{\pm 5.49}$	$70.31_{\pm 4.23}$	$\mathbf{83.32}_{\pm 0.77}$	$86.03_{\pm 0.56}$
	Day 2	$9.25_{\pm 32.85}$	$53.00_{\pm 6.85}$	$-5.86_{\pm 3.90}$	$6.03_{\pm 8.44}$	$31.53_{\pm 6.13}$	$80.82_{\pm 1.36}$	$\mathbf{84.84}_{\pm 4.68}$	$89.60_{\pm 0.52}$
	Day 3	$-128.25_{\pm 65.07}$	$23.32_{\pm 13.39}$	$-2.09_{\pm 2.34}$	$6.32_{\pm 13.51}$	$25.11_{\pm 12.50}$	$68.66_{\pm 2.24}$	$\mathbf{77.69}_{\pm 2.91}$	$86.35_{\pm 0.99}$
	Day 9	$-24.15_{\pm 33.53}$	$-5.20_{\pm 21.77}$	$-1.80_{\pm 2.15}$	$-76.27_{\pm 50.66}$	$38.72_{\pm 6.35}$	$74.84_{\pm 1.52}$	$\mathbf{84.14}_{\pm 1.96}$	$88.55_{\pm 0.68}$
	Day 10	$-70.33_{\pm 65.25}$	$-2.22_{\pm 20.13}$	$-3.70_{\pm 3.36}$	$3.23_{\pm 8.19}$	$42.12_{\pm 9.81}$	$74.61_{\pm 1.14}$	$\mathbf{82.18}_{\pm 1.17}$	$89.19_{\pm 0.80}$
	Day 14	$-65.46_{\pm 24.55}$	$-13.54_{\pm 26.38}$	$-0.87_{\pm 0.82}$	$-38.13_{\pm 72.01}$	$39.90_{\pm 20.83}$	$63.52_{\pm 1.53}$	$\mathbf{73.95}_{\pm 2.67}$	$85.16_{\pm 0.64}$
	Day 15	$-32.08_{\pm 24.64}$	$-31.94_{\pm 17.11}$	$-4.45_{\pm 2.39}$	$-9.75_{\pm 16.73}$	$35.71_{\pm 15.39}$	$78.00_{\pm 0.39}$	$\mathbf{84.41}_{\pm 0.68}$	$91.39_{\pm 0.56}$
	Day 16	$-123.74_{\pm 63.89}$	$-10.21_{\pm 17.65}$	$-2.26_{\pm 1.04}$	$-29.42_{\pm 57.08}$	$41.33_{\pm 13.65}$	$74.52_{\pm 0.34}$	$\mathbf{80.91}_{\pm 0.95}$	$90.80_{\pm 0.34}$
	Day 36	$-70.67_{\pm 99.37}$	$-55.33_{\pm 19.88}$	$-4.24_{\pm 3.58}$	$-29.41_{\pm 56.85}$	$35.17_{\pm 8.61}$	$39.70_{\pm 34.29}$	$\mathbf{74.02}_{\pm 2.78}$	$89.56_{\pm 0.51}$
	Day 37	$-29.54_{\pm 59.36}$	$-44.82_{\pm 31.72}$	$-3.78_{\pm 3.13}$	$-2.44_{\pm 10.22}$	$51.48_{\pm 10.78}$	$67.46_{\pm 3.59}$	$\mathbf{81.31}_{\pm 1.67}$	$91.80_{\pm 0.42}$
	Day 38	$-112.02_{\pm 132.39}$	$-23.46_{\pm 22.71}$	$-2.53_{\pm 1.87}$	$-4.37_{\pm 5.70}$	$41.33_{\pm 8.24}$	$28.18_{\pm 1.15}$	$\mathbf{64.68}_{\pm 2.72}$	$77.45_{\pm 0.48}$

C.2 Decoded Cursor Velocity Visualization

As shown in Fig. 4 and Fig. 5, we visualized reach trajectories of CO-M integrated from the decoded cursor velocity.

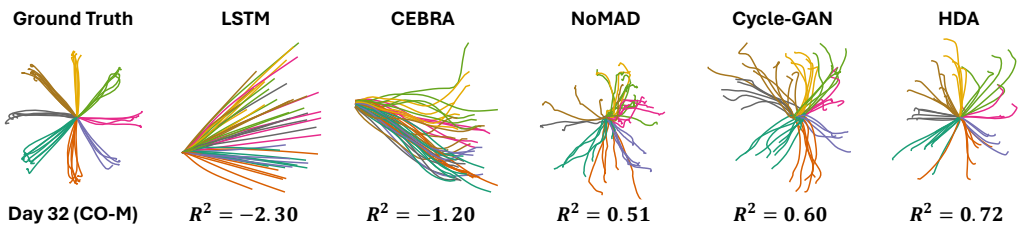

Figure 4: True and decoded cursor trajectories, integrated from the decoded velocity, are presented for baselines and HDA after aligning Day 32 to Day 0 on the CO-M. Different colors represent eight different reach directions.

C.3 Additional Analysis on HDA

We also conducted experiments using Day 0 as the source session, with source and target training ratios set at $0.1, 0.2, 0.4, 0.6$. As illustrated in Fig. 6(b), HDA shows effective performance even with a relatively small number of trials.

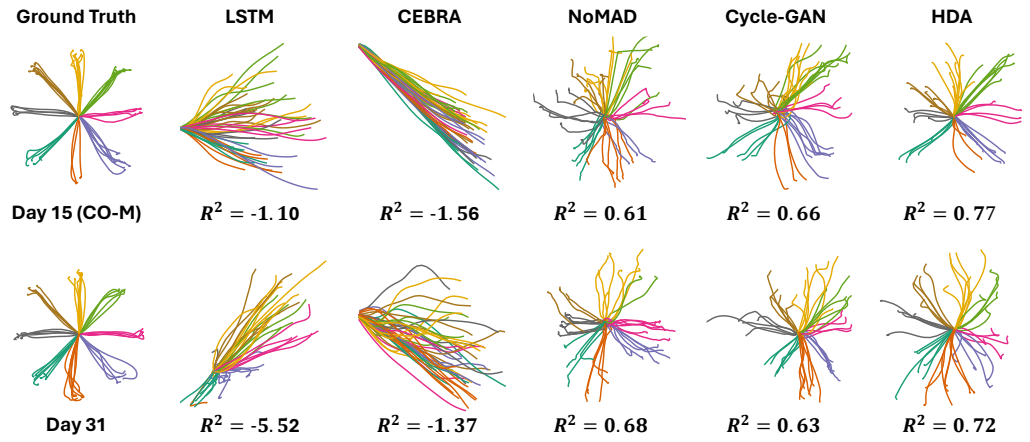

Figure 5: True and decoded cursor trajectories, integrated from the decoded velocity, are presented for baselines and HDA on Day 15 and Day31 of CO-M dataset. Different colors represent eight different reach directions.

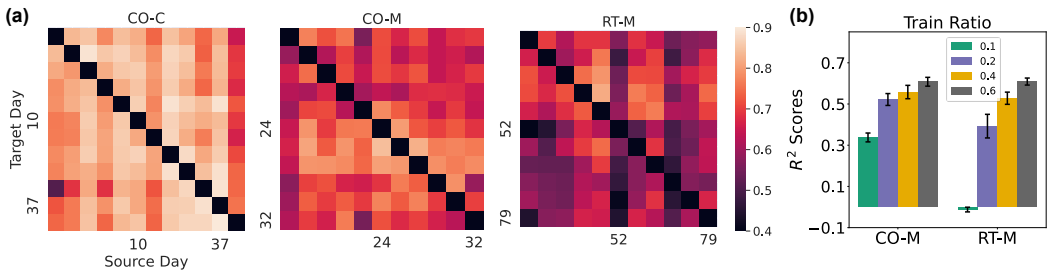

Figure 6: (a) Overall performance of average R^2 scores on all the sessions(days) from CO-C, CO-M, and RT-M. (b) HDA's performance on CO-M and RT-M at different training ratios (0.1, 0.2, 0.4, and 0.6). Here, the source session is Day0.

C.4 Feature Stability Comparison

As for feature stability, we compared our MLE with those from ERDiff and NoMAD, as MLE can only be derived from sequential models. Fig. 7(d) shows the average MLE for each target session and their overall averages.

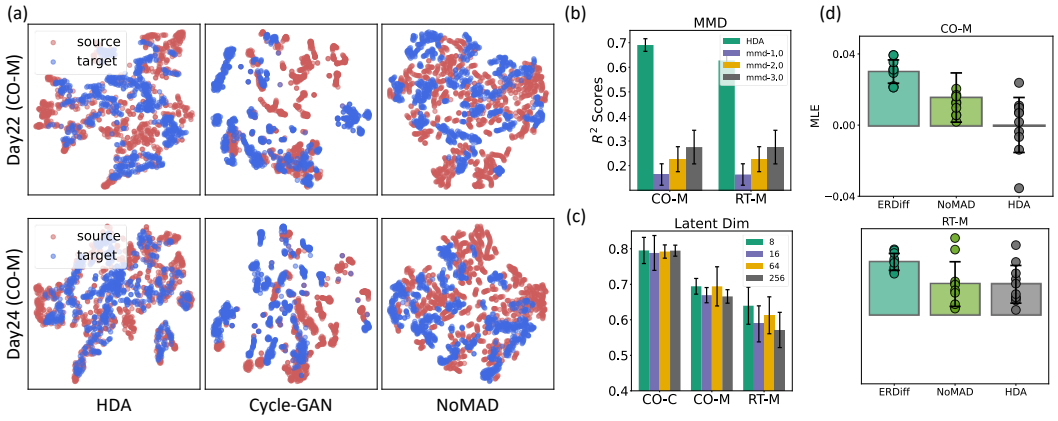

Figure 7: (a) t-SNE visualizations of HDA compared to Cycle-GAN and NoMAD on Day22 and 24 of CO-M. (b) R^2 scores of HDA and MMD with different Gaussian kernels (μ=1.0, 2.0, 3.0). (c) R^2 score performance of HDA across different latent dimensions. ($\tilde{k}_o = \tilde{k}_y$ =8, 16, 64, 256). (d) Total average Maximum Lyapunov Exponent (MLE) for baselines containing sequential models (ERDiff, NoMAD) and HDA on CO-M and RT-M. Dots represent average MLE from an individual target session.

C.5 Latent Feature Visualization

As shown in Fig. 7(a), the t-SNE results are compared with the top two baselines, demonstrating our superior alignment performance.

C.6 Visualization of Dynamical Latent Features

To examine our decomposition of the latent spaces, we selected CO-M as the representative dataset for visualization. We presented a visualization of the semantic dynamical latent features $\tilde{\mathbf{Z}}_y$, the domain-related latent features $\tilde{\mathbf{Z}}_o$, and original latent features $\mathbf{Z}$ from both the source and several target sessions, utilizing t-SNE for dimensionality reduction. These visualizations are depicted in Fig. 8 and Fig. 9. Our analysis reveals that the semantic latent features of the source and target sessions are closely aligned, while a discrepancy is observed in the distribution of the domain-related and original features. This observation suggests that HDA has effectively decomposed the latent space into semantic and domain-related subspaces.

C.7 Hyper-parameter Sensitivity Analysis

The main hyper-parameters of our method include the signal window size (w), and the weights balancing terms in the final loss function ($\lambda_{b,o}$), and latent feature dimensions($\tilde{k}_{o,y}$). The results of their sensitivity analysis are shown in Tables 9 to 11, and Fig. 7(c).

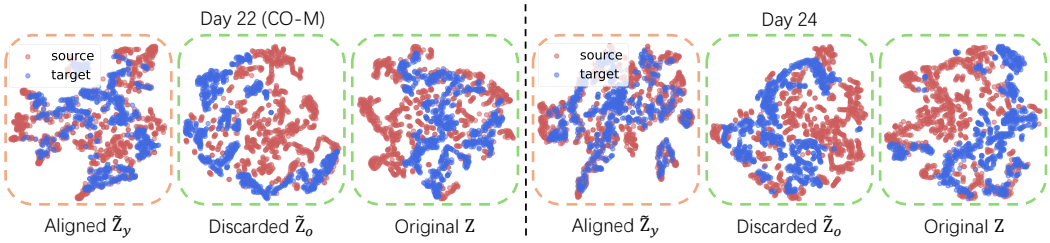

Figure 8: Visualizations via t-SNE are presented, depicting the semantic latent features $\tilde{\mathbf{Z}}_y$, the domain-related latent features $\tilde{\mathbf{Z}}_o$, and original latent features $\mathbf{Z}$. Each figure shows latent features from the source session and a specific target session from CO-M, represented by different colors.

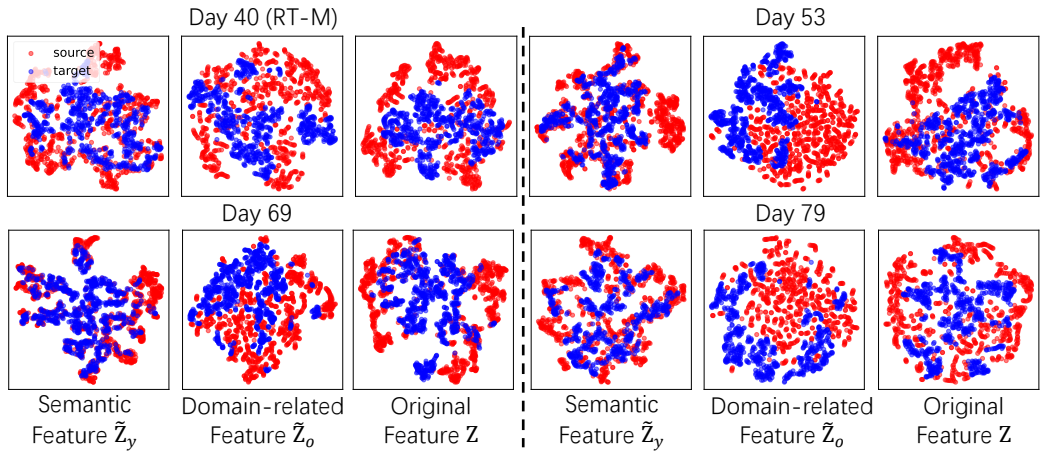

Figure 9: Visualizations via t-SNE are presented, depicting the semantic latent features $\tilde{\mathbf{Z}}_y$, the domain-related latent features $\tilde{\mathbf{Z}}_o$, and original latent features $\mathbf{Z}$. Each figure shows latent features from the source session and a specific target session from RT-M, represented by different colors.

Table 9: Average R^2 scores for different datasets with varying λ_b.

λ_b	CO-C	CO-M	RT-M
0.0001	0.7912 ± 0.0233	0.6619 ± 0.0502	0.6448 ± 0.0527
0.001	0.7924 ± 0.0237	0.6641 ± 0.0496	0.6446 ± 0.0536
0.01	0.7984 ± 0.0194	0.6921 ± 0.0289	0.6437 ± 0.0206
0.1	0.8109 ± 0.0177	0.6838 ± 0.0425	0.6563 ± 0.0529

Table 10: Average R^2 scores for different datasets with varying λ_o.

λ_o	CO-C	CO-M	RT-M
0	0.7924 ± 0.0190	0.6768 ± 0.0431	0.6502 ± 0.0492
0.1	0.7960 ± 0.0168	0.6793 ± 0.0350	0.6512 ± 0.0350
1	0.7992 ± 0.0194	0.6921 ± 0.0289	0.6437 ± 0.0206
2	0.7984 ± 0.0154	0.6838 ± 0.0425	0.6563 ± 0.0529

Table 11: Average R^2 scores for different datasets with varying w.

w	CO-C	CO-M	RT-M
4	0.7640 ± 0.0351	0.6704 ± 0.0339	0.6273 ± 0.0534
5/6	0.7984 ± 0.0194	0.6921 ± 0.0289	0.6437 ± 0.0206
7	0.7769 ± 0.0489	0.6519 ± 0.0781	0.6559 ± 0.0473
8	0.8074 ± 0.0348	0.6703 ± 0.0765	0.6240 ± 0.0618