Gated Adaptation for Continual Learning in Human Activity Recognition

Reza Rahimi Azghan*, Gautham Krishna Gudur†, Mohit Malu*
Edison Thomaz†, Giulia Pedrielli*, Pavan Turaga*, Hassan Ghasemzadeh*

***Abstract*—Wearable sensors in Internet of Things (IoT) ecosystems increasingly support applications such as remote health monitoring, elderly care, and smart home automation, all of which rely on robust human activity recognition (HAR). Continual learning systems must balance plasticity (the ability to learn new tasks) with stability (the retention of previously acquired knowledge). However, AI models often exhibit catastrophic forgetting, where learning new tasks degrades performance on earlier ones. This challenge is particularly acute in domain-incremental settings such as HAR, where on-device models must adapt to new subjects with distinct movement characteristics while maintaining accuracy on previously seen subjects without transmitting sensitive data to the cloud. In this work, we propose a parameter-efficient continual learning framework based on channel-wise gated modulation applied to frozen pretrained representations. Our key insight is that adaptation should operate through feature *selection* rather than feature *generation*: by restricting learned transformations to diagonal scaling of existing features, we preserve the geometric structure of pretrained representations while enabling subject-specific modulation. We provide a theoretical analysis showing that gating implements a bounded diagonal operator that limits representational drift compared to unconstrained linear transformations. Empirically, we demonstrate that freezing the backbone substantially reduces forgetting and that lightweight gates restore the adaptation capacity lost from freezing, achieving both stability and plasticity simultaneously. On the PAMAP2 dataset with 8 sequential subjects, our approach reduces forgetting from 39.7% (trainable backbone) to 16.2% and improves final accuracy from 56.7% to 77.7%, while training less than 2% of model parameters. Our method matches or exceeds standard continual learning baselines without requiring replay buffers or task-specific regularization, confirming that structured diagonal operators provide an effective and efficient mechanism for continual learning under distribution shift.**

Index Terms—Continual learning, catastrophic forgetting, parameter-efficient fine-tuning, human activity recognition, wearable sensors, gated neural networks, domain-incremental learning.

I. Introduction

Adapting to a dynamic environment is one of the fundamental attributes of intelligent systems. Human beings, as the most capable example of intelligence in the natural world, have developed this ability through years of evolution. They are able to learn new concepts and acquire new knowledge without forgetting what they have previously learned. In artificial intelligence, this capability is studied under the framework of continual learning [1], which refers to models that incrementally learn from a continuous stream of data and tasks over time, adapt to changing environments without retraining from scratch, and, critically, retain knowledge acquired from earlier experiences. To support such behavior, continual learning systems must address the well-known stability–plasticity dilemma [2], where models need to remain sufficiently plastic to learn new tasks while also being stable enough to preserve previously learned information.

In practice, learning systems often favor plasticity over stability, which leads to a phenomenon known as catastrophic forgetting [3], where performance on previously learned tasks deteriorates as new tasks are introduced. This issue becomes more severe in real-world deployments, where models are exposed to large volumes of data originating from multiple, potentially complex and evolving distributions. In these settings, data is commonly organized into tasks, where each task consists of a set of samples generated under consistent conditions. More precisely, a task consists of data points drawn from a shared generative distribution. While different tasks may vary in their data distributions, samples within a task are assumed to exhibit similar statistical properties. Continual learning systems are therefore required to accommodate shifts between such distributions over time while maintaining performance on previously encountered tasks.

The proliferation of wearable sensors in Internet of Things (IoT) ecosystems has enabled a wide range of applications that depend on robust human activity recognition (HAR), including remote health monitoring [4], [5], elderly care and fall detection [6], smart home automation [7], [8], and fitness tracking. In these deployments, HAR models typically execute on resource-constrained edge devices such as smartwatches, fitness bands, or smartphones, where they must operate under strict memory, compute, and energy budgets [9]. A critical requirement for such systems is the ability to personalize to new users over time: as a wearable device is adopted by different household members or as a telehealth system onboards new patients, the model must adapt to each individual's movement characteristics without forgetting previously learned users.

Sensor-based HAR [10] provides a natural testbed for studying this challenge, as individual subjects define task boundaries [11]. Each person exhibits distinct movement

*Arizona State University, Phoenix, AZ, USA,

† The University of Texas at Austin, Austin, TX, USA
Email: {rrahimia, hassan.ghasemzadeh, pavan.turaga, giulia.pedrielli, mmalu}@asu.edu, {ethomaz, gauthamkrishna}@utexas.edu

This work was supported in part by the National Science Foundation under grant IIS-2402650. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organization.

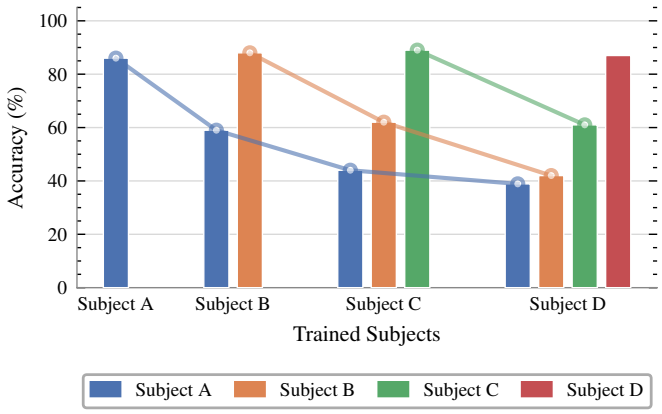

Fig. 1. Catastrophic forgetting in subject-incremental HAR: accuracy on Subject 1 drops from 85% to 40% after training on just three additional subjects (PAMAP2 dataset).

patterns, physiological characteristics, and sensor placement variations. While activity labels remain constant (e.g., walking, sitting, running), the manner in which these activities are performed varies substantially across subjects, inducing subject-specific data distributions. Importantly, transmitting raw sensor data to cloud servers for centralized retraining raises significant privacy concerns, as movement patterns can reveal sensitive health information and behavioral profiles [12]. This motivates *on-device continual learning*, where models adapt locally to new subjects without requiring data transmission. Figure 1 illustrates the severity of this challenge on the PAMAP2 dataset [13], where a multilayer perceptron model suffers catastrophic forgetting: after training on only four subjects, the accuracy on the first subject drops from 85% to 40%, a decline of 45 percentage points.

To address catastrophic forgetting, three main paradigms have emerged in the continual learning literature. *Regularization-based methods* [14], [15] constrain parameter updates by penalizing changes to weights deemed important for previous tasks, typically through Fisher information or importance estimates. *Replay-based methods* [16], [17] store or generate samples from prior tasks to maintain performance through periodic rehearsal. *Architectural methods* [18], [19] expand model capacity or selectively activate subnetworks for different tasks. While these approaches have demonstrated effectiveness in controlled settings, they face practical limitations for IoT deployment: regularization methods can be overly conservative and require storing per-task importance weights, replay methods raise privacy concerns and demand persistent storage of sensitive sensor data, and architectural expansion increases model size beyond the memory constraints of edge devices.

More recently, *frozen backbone* strategies have gained traction [20], [21], where a pretrained feature extractor remains fixed while only task-specific components are trained. This paradigm offers computational efficiency and inherent stability, as the shared representation cannot be corrupted by new task updates. However, a critical question remains: how should adaptation occur on top of a frozen backbone? Common approaches include training only the final classifier or adding trainable adapter layers [22]. The former provides minimal adaptation capacity, often insufficient for significant distribution shifts, while the latter introduces dense transformations that can generate entirely new feature spaces, undermining the stability benefits of the frozen backbone. This motivates our focus on structured, low-capacity adaptation mechanisms that balance expressiveness with stability.

Within the context of parameter-efficient fine-tuning, we propose the use of gated modulation mechanisms to selectively adapt intermediate representations of a neural network. Specifically, we introduce lightweight channel-wise gates that operate on the outputs of intermediate layers and modulate feature activations based on the input. The proposed gating mechanism is inspired by the design principles of squeeze-and-excitation networks [23], but is employed here as a means of continual adaptation rather than static feature recalibration. By placing these gates on top of a frozen pretrained backbone, the model is able to adjust the relative importance of learned features while preserving the stability of the underlying representation.

Our analysis demonstrates that this form of structured, low-capacity adaptation reduces catastrophic forgetting compared to fully fine-tuning all model parameters, as updates are constrained to bounded, task-adaptive gates and a shared classifier. Moreover, in contrast to approaches that introduce additional trainable layers on top of frozen backbones, gated modulation provides a more controlled form of adaptation by restricting changes to channel-wise reweighting rather than arbitrary feature transformations. We support these claims through both theoretical analysis and extensive empirical evaluation on multiple sensor-based human activity recognition datasets.

In summary, the contributions of this paper are as follows:

- We propose a parameter-efficient continual learning framework that employs channel-wise gated modulation to adapt frozen pretrained neural representations under subject-induced distribution shifts. The approach requires less than 2% of trainable parameters compared to full fine-tuning, making it suitable for deployment on resource-constrained IoT edge devices.
- We provide a theoretical analysis demonstrating that gated adaptation implements a diagonal operator. This mechanism bounds representational drift through structured channel-wise reweighting, limiting interference with previously learned tasks.
- We conduct extensive empirical evaluations across three sensor-based HAR benchmarks (PAMAP2, UCI-HAR, DSA) under domain-incremental settings with up to 30 sequential subjects, demonstrating consistent improvements in the stability-plasticity tradeoff. Our approach substantially outperforms both minimal adaptation (frozen backbone with classifier only) and high-capacity adaptation (stacked trainable layers) across all datasets, validating our theoretical predictions.

II. Related Work

Our work sits at the intersection of continual learning, parameter-efficient fine-tuning, attention mechanisms, and hu-

man activity recognition. We review each area and identify the specific gap our approach addresses.

A. Continual Learning and Catastrophic Forgetting

Continual learning addresses the challenge of training neural networks on sequential tasks without forgetting previously acquired knowledge, a phenomenon known as catastrophic forgetting [2], [14]. Three main paradigms have emerged to address this challenge. *Regularization-based methods* constrain parameter updates to preserve important weights: Elastic Weight Consolidation (EWC) [14] penalizes changes to parameters deemed important for prior tasks via the Fisher information matrix, while Learning without Forgetting (LwF) [24] uses knowledge distillation to maintain output consistency. *Replay-based methods* store or generate exemplars from previous tasks [16], [25], though this raises privacy and memory concerns. *Architecture-based methods* allocate task-specific parameters [18], [26], but often incur growing model complexity. Recent comprehensive surveys [27]–[29] highlight that while significant progress has been made, existing approaches involve trade-offs between stability (retaining old knowledge) and plasticity (learning new tasks). Our work contributes to the architecture-based paradigm by introducing lightweight, bounded gating mechanisms that modulate frozen representations.

B. Parameter-Efficient Fine-Tuning for Continual Learning

The rise of large pretrained models has motivated parameter-efficient fine-tuning (PEFT) methods that adapt models by updating only a small subset of parameters while keeping the backbone frozen [22], [30]. This frozen backbone paradigm provides inherent stability against forgetting, as the core representations remain unchanged. Recent work has explicitly connected PEFT with continual learning: Learning to Prompt (L2P) [20] maintains a learnable prompt pool to dynamically instruct a frozen Vision Transformer across sequential tasks without rehearsal. DualPrompt [21] extends this with task-invariant and task-specific prompt components, while CODA-Prompt [31] introduces end-to-end attention-based prompt assembly. A recent survey on pretrained model-based continual learning [32] systematically analyzes these approaches, noting their success in rehearsal-free settings. However, critical analysis [33] reveals that simple baselines with frozen backbones can be surprisingly competitive, questioning the necessity of complex prompt mechanisms.

C. Gating and Channel Attention Mechanisms

Squeeze-and-Excitation Networks (SE-Nets) [23] introduced channel-wise recalibration, learning to weight feature channels based on global context via a squeeze-excitation block. This mechanism has been extended to spatial attention in CBAM [34] and applied to feature conditioning in FiLM [35]. In the continual learning context, Masse et al. [36] demonstrated that context-dependent gating inspired by neuroscience can alleviate forgetting by selectively activating subnetworks. For human activity recognition specifically, SE-blocks have been integrated into deep residual networks for wearable sensor data [37], showing improved recognition accuracy through channel attention. Our work is inspired by SE-Nets but applies gating in a fundamentally different context: rather than learning static attention weights, we train gates that modulate a frozen pretrained backbone for domain-incremental adaptation. This provides bounded feature drift while maintaining representational stability.

D. Continual Learning for Human Activity Recognition

Human activity recognition (HAR) from wearable sensors presents a natural testbed for continual learning, as deployed systems must adapt to new users, devices, and environments over time [38]–[40]. Subject-wise variation, where different individuals perform the same activity with distinct motion patterns, induces domain shifts that cause catastrophic forgetting when models are transferred across users. Jha et al. [39] provided an empirical benchmark evaluating standard continual learning techniques (EWC, LwF, replay) on HAR datasets, finding that replay-based methods generally outperform regularization approaches but at the cost of memory overhead. Recent advances include online continual learning frameworks [41] that handle streaming sensor data, prototype-based approaches [42], [43] using experience replay with continual prototype evolution, and self-supervised methods that leverage unlabeled data [44]. Most recently, CLAD-Net [45] combines self-supervised representation learning with knowledge distillation for domain-incremental HAR.

In addition to these efforts, on-device continual learning has also been explored for resource-constrained platforms. HARNet [46] proposes a lightweight deep ensemble architecture designed for incremental learning directly on constrained mobile devices. By enabling models to adapt to new activities without retraining from scratch and without relying on cloud resources, HARNet demonstrates that continual adaptation can be achieved efficiently under strict computational and memory budgets. This line of work highlights the growing need for continual learning systems that remain practical in real-world, embedded HAR deployments.

Recent research has also examined lifelong adaptation through metric-based learning. Adaimi and Thomaz [42] introduced a prototypical-network-based framework that incrementally updates class prototypes to support lifelong sensor-based activity recognition. Their approach reduces catastrophic forgetting by anchoring new representations around stable prototypes while enabling rapid adaptation to additional classes or users. This prototypical perspective complements architecture- and replay-based approaches by emphasizing representation stability during incremental updates.

E. Summary and Research Gap

While frozen backbone approaches with PEFT have shown promise for continual learning, existing methods rely on additive mechanisms (prompts, adapters) that lack theoretical guarantees on representation drift. Simultaneously, gating mechanisms have proven effective for channel recalibration but have not been systematically applied to frozen backbones in domain-incremental settings. In HAR specifically, continual

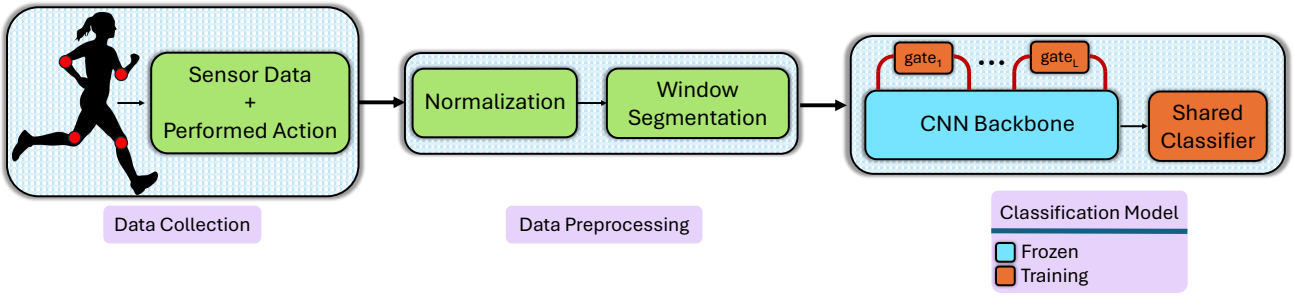

Fig. 2. Overview of the proposed continual learning system for HAR, comprising data collection, preprocessing, and a classification model with a frozen pretrained backbone modulated by lightweight trainable gates.

learning research has focused on applying general-purpose methods without leveraging the channel-wise structure of subject-induced domain shifts. Our work addresses this gap by combining a frozen pretrained backbone with lightweight, channel-wise gates, providing both theoretical bounds on feature drift and empirical improvements in knowledge retention.

Although our gates share the parameter-efficient, frozen-backbone philosophy of prompt- and adapter-based methods [20]–[22], [30], they differ in a fundamental way. Adapters and prompts are additive: they introduce new parameters that generate feature subspaces outside the pretrained representation, and the resulting drift is unconstrained, since the injected transformation can in principle move features arbitrarily. Our gates are instead multiplicative and diagonal: they reweight existing channels rather than synthesize new ones, which bounds representational drift by construction (Theorem 1) and underlies our stability guarantees. Our approach also differs from masking-based methods such as HAT [19] and PackNet [26]. These partition network capacity by learning hard, typically binary masks that allocate disjoint parameter subsets per task, requiring per-task masks and explicit task identity. In contrast, our gates apply continuous channel-wise scaling to a single shared representation, store one set of parameters regardless of the number of tasks, and operate without task boundaries at test time. In short, prior methods adapt through feature generation or capacity partitioning, whereas our method adapts through bounded feature selection

III. Proposed Approach

A. System Overview

This section presents our proposed system, structured as a pipeline from raw sensor reading to continual model adaptation. As illustrated in Figure 2, the system comprises three core components.

During data collection, inertial measurement units (IMUs) are placed on multiple body locations while subjects perform a predefined set of activities. Subjects provide activity labels indicating the action performed at each time instant, enabling supervised training. Next, raw sensor streams from each subject undergo per-subject normalization to account for individual differences in sensor placement and motion intensity. A sliding window segmentation algorithm then partitions the continuous data into fixed-length segments. Each window is represented as a tuple, $(\mathbf{x}, y, t)$, which represents the normalized input data, the activity associated with the input, and the user ID from whom the data was collected.

The final component is a continual learning model designed to classify segmented windows into activity categories. The central objective is to achieve high recognition accuracy on incoming subjects (plasticity) while preserving performance on previously encountered subjects (stability). To address this stability-plasticity trade-off, we introduce lightweight trainable gates that modulate a pretrained backbone's representations to accommodate subject-specific domain shifts. The architecture and its associated training objectives are detailed in the following subsections.

B. Problem Formulation

We formulate continual learning for human activity recognition as a *domain-incremental* learning problem. The learner encounters a sequence of T tasks, where each task $t \in \{1, \ldots, T\}$ corresponds to a distinct data distribution representing a new subject. At task t, the model receives training data,

$$\mathcal{D}_t = \{(\mathbf{x}_{i,t}, y_{i,t})\}_{i=1}^{N_t},$$

where $\mathbf{x}_{i,t} \in \mathbb{R}^{c_0 \times d_0}$ is a multivariate time series of length d_0 with c_0 sensor channels, and $y_{i,t} \in \mathcal{Y}$ is the corresponding activity label from a fixed label set. Each task is drawn from a subject-specific distribution $\mathcal{D}_t \sim \mathbf{P}_t(\mathbf{x}, y)$.

Under the domain-incremental assumption, the label distribution remains stationary across tasks,

$$\mathbf{P}_t(y) = \mathbf{P}_{t+1}(y), \quad \forall t \in \{1, \ldots, T-1\},$$

while the marginal distribution over inputs may shift due to subject-specific characteristics,

$$\mathbf{P}_t(\mathbf{x}) \neq \mathbf{P}_{t'}(\mathbf{x}), \quad \forall t \neq t'.$$

This captures the HAR scenario: all subjects perform the same activities (fixed label semantics), but exhibit individual variations in movement patterns, sensor placement, and physiological characteristics (varying input statistics).

The model is exposed to tasks sequentially and cannot store or revisit data from previous tasks. This reflects memory constraints in embedded and mobile deployment scenarios. The objective is to maintain high classification accuracy on

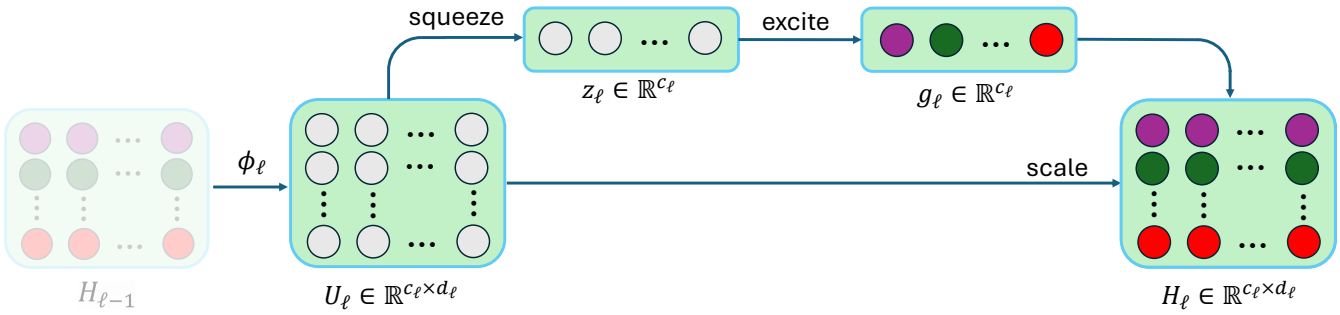

Fig. 3. Channel-wise gating mechanism. Given an intermediate feature map U_ℓ, global average pooling (squeeze) produces a channel descriptor z_ℓ, which is transformed through bottleneck layers (excitation) to produce gate values $g_\ell \in (0,1)^{c_\ell}$. The gates then scale each channel independently, preserving feature directions while modulating magnitudes.

all encountered tasks, including both the current task and all previous tasks, thereby balancing *plasticity* (adaptation to new subjects) with *stability* (retention of knowledge about earlier subjects).

C. Model Overview

As illustrated in Figure 2, our proposed model consists of a frozen pretrained backbone, a set of lightweight channel-wise gating modules inserted after each backbone block, and a shared classifier trained across all tasks.

a) Frozen Pretrained Backbone: The backbone comprises L residual convolutional blocks $\{\phi_\ell\}_{\ell=1}^{L}$, each mapping $\phi_\ell : \mathbb{R}^{c_{\ell-1} \times d_{\ell-1}} \to \mathbb{R}^{c_\ell \times d_\ell}$. Given an input sequence $\mathbf{x}$, the output of block ℓ is,

$$U_\ell = \phi_\ell(H_{\ell-1}), \quad H_0 = \mathbf{x}$$

These blocks are pretrained on a source HAR dataset and remain fixed throughout continual learning. Freezing the backbone ensures representational stability, as the shared feature extractor does not undergo updates that could degrade performance on earlier subjects. Moreover, this strategy reduces the number of trainable parameters by orders of magnitude, yielding substantial computational efficiency and enabling deployment on lightweight devices.

b) Channel-Wise Gates: At the output of each intermediate layer in the backbone, we introduce lightweight gating modules that adaptively reweight channel activations. The gating mechanism, illustrated in Figure 3, is inspired by Squeeze-and-Excitation Networks (SENets) [23] but repurposed for continual adaptation. For a feature map $U_\ell \in \mathbb{R}^{c_\ell \times d_\ell}$ at an intermediate layer, the gate starts by computing a global descriptor by aggregating temporal information.

$$z_\ell = \frac{1}{d_\ell} \sum_{j=1}^{d_\ell} U_{\ell,:,j}$$

where $z_\ell \in \mathbb{R}^{c_\ell}$ denotes the channel-wise mean activation aggregated over all temporal positions. In the original SENet formulation, this operation corresponds to the squeeze stage.

The resulting descriptor z_ℓ is then passed through an excitation mapping parameterized by learnable weights. In our implementation, the excitation function is implemented using two fully connected layers that transform the compressed representation z_ℓ into channel-wise gating coefficients g_ℓ via a sigmoid nonlinearity, given by,

$$g_\ell = \sigma(\mathbf{W}_{2,\ell} \cdot \mathrm{ReLU}(\mathbf{W}_{1,\ell} \cdot z_\ell))$$

where $\mathbf{W}_{1,\ell} \in \mathbb{R}^{c_\ell/r \times c_\ell}$ and $\mathbf{W}_{2,\ell} \in \mathbb{R}^{c_\ell \times c_\ell/r}$ are learnable bottleneck layers with a reduction ratio r, and σ is the sigmoid activation that constrains the gating vector to $g_\ell \in (0,1)^{c_\ell}$.

The final step of the gating module applies the learned channel-wise coefficients to the intermediate feature representation. Specifically, the gating vector g_ℓ is used to reweight the channels of U_ℓ through a diagonal scaling operation,

$$H_\ell = \mathrm{D}(g_\ell) \cdot U_\ell,$$

where

$$\mathrm{D}(g_\ell) = \begin{bmatrix} g_{\ell,1} & 0 & \cdots & 0 \\ 0 & g_{\ell,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{\ell,c_\ell} \end{bmatrix}$$

This diagonal operation restricts adaptation to channel-wise *magnitude scaling* while preserving the *directional structure* of the pretrained features. This bounded form of modulation limits its representational drift and enables stable continual learning.

c) Shared Classifier: Finally, we employ a single-layer linear classifier shared across all tasks and continuously updated as new subjects are introduced. First, a temporal pooling operator is applied to the final gated feature map H_L to obtain a fixed-dimensional representation,

$$h = \frac{1}{d_L} \sum_{j=1}^{d_L} H_{L,:,j}$$

followed by a linear operation,

$$\hat{y} = \mathbf{W}_{\mathrm{cl}} h$$

where $\mathbf{W}_{\mathrm{cl}} \in \mathbb{R}^{|\mathcal{Y}| \times c_L}$, and is trained continually as new tasks are encountered. This design forces the model to learn a common decision boundary in the gated feature space rather than task-specific output mappings.

Algorithm 1 summarizes the training procedure. Here, only the gate parameters $\mathbf{W}_{1,\ell}, \mathbf{W}_{2,\ell}$ and the classifier $\mathbf{W}_{\text{cl}}$ are updated for each task. The backbone blocks remain frozen throughout. This dramatically reduces the number of trainable parameters per task (typically less than 2% of total model parameters) and enforces stability by preventing changes to the shared feature representation. Furthermore, the algorithm does not store data from previous tasks and does not require explicit task boundary signals at test time. The model must infer subject identity implicitly from input statistics via the gating mechanism.

Algorithm 1 Continual Training with Interleaved Channel-Wise Gates

Require: Frozen pretrained backbone blocks $\{\phi_\ell\}_{\ell=1}^{L}$, gate modules with parameters $\{\mathbf{W}_{1,\ell}\,\mathbf{W}_{2,\ell}\}_{\ell=1}^{L}$, classifier with parameter $\mathbf{W}_{\text{cl}}$
Require: data stream $(\mathbf{x}_t, y_t)$, epochs E, optimizer $\mathcal{O}$
Initialize all parameters $\mathbf{\Theta} = \{\{\mathbf{W}_{1,\ell}\,\mathbf{W}_{2,\ell}\}_{\ell=1}^{L}, \mathbf{W}_{\text{cl}}\}$; freeze $\{\phi_\ell\}_{\ell=1}^{L}$
for $e = 1, \ldots, E$ **do**
  for each minibatch $\mathcal{B}$ drawn from $(\mathbf{x}_t, y_t)$ **do**
    for each $(\mathbf{x}, y) \in \mathcal{B}$ **do**
      $H_0 \leftarrow \mathbf{x}$
      for $\ell = 1, \ldots, L$ **do**
        $U_\ell \leftarrow \phi_\ell(H_{\ell-1})$
        z_ℓ = Pool(U_ℓ)
        $g_\ell \leftarrow \sigma(\mathbf{W}_{2,\ell} \cdot \text{ReLU}(\mathbf{W}_{1,\ell} \cdot z_\ell))$
        $H_\ell \leftarrow \text{D}(g_\ell) \cdot U_\ell$
      end for
      $h \leftarrow \text{Pool}(H_L)$
      $\hat{y} \leftarrow \mathbf{W}_{\text{cl}} \cdot h$
      $\mathcal{L} \leftarrow \text{CrossEntropy}(\hat{y}, y) + \lambda\|\mathbf{\Theta}\|_2^2$
    end for
    Compute gradients $\nabla_{\mathbf{\Theta}}\mathcal{L}$
    Update model parameters $\mathbf{\Theta} \leftarrow \mathcal{O}(\mathbf{\Theta}, \nabla_{\mathbf{\Theta}}\mathcal{L})$
  end for
end for

IV. Stability–Plasticity Trade-offs in Gated Continual Learning

The theoretical analysis clarifies why channel-wise gating yields improved stability in domain-incremental HAR. By restricting adaptation to bounded, diagonal modulation of frozen backbone features, updates induced by new subjects result in limited functional drift on previously seen data. This contrasts with full fine-tuning, classifier-only, or stacked-layer adaptations, where accommodating subject-specific shifts often requires larger or less structured parameter updates, increasing interference across tasks. In the following, we establish formal guarantees on stability, characterize the expressiveness of diagonal gating, and analyze why this mechanism achieves a favorable stability–plasticity trade-off.

A. Stability Guarantees

We begin by analyzing how gated adaptation affects the internal representations learned for previously seen subjects. Because the backbone is frozen, all representational changes induced by learning a new task arise exclusively from updates to the channel-wise gates. This allows us to characterize forgetting in terms of how much the gated feature maps drift on inputs from earlier tasks.

Theorem 1 (Bounded Feature Drift under Diagonal Gating)**.** *Let $U(\mathbf{x}) \in \mathbb{R}^{C\times d}$ denote the ungated feature map produced by a frozen backbone for an input $\mathbf{x}$. Let $g(\mathbf{x}), g'(\mathbf{x}) \in (0,1)^C$ be the channel-wise gate vectors before and after learning a new task, and define the corresponding gated representations,*

$$H(\mathbf{x}) = D(g(\mathbf{x}))\,U(\mathbf{x}), \qquad H'(\mathbf{x}) = D(g'(\mathbf{x}))\,U(\mathbf{x}),$$

where $D(\cdot)$ denotes a diagonal matrix. Then the feature drift satisfies,

$$\|H'(\mathbf{x}) - H(\mathbf{x})\|_F \;\le\; \delta(\mathbf{x})\,\|U(\mathbf{x})\|_F,$$

where $\delta(\mathbf{x}) = \|g'(\mathbf{x}) - g(\mathbf{x})\|_\infty < 1$.

Proof. Because the backbone parameters are frozen, the ungated feature map $U(\mathbf{x})$ remains unchanged before and after learning a new task. The difference between the gated representations can therefore be written as,

$$\begin{aligned} H'(\mathbf{x}) - H(\mathbf{x}) &= \big(D(g'(\mathbf{x})) - D(g(\mathbf{x}))\big)\,U(\mathbf{x}) \\ &= D\big(g'(\mathbf{x}) - g(\mathbf{x})\big)\,U(\mathbf{x}). \end{aligned}$$

Applying submultiplicativity of matrix norms and using the fact that the spectral norm of a diagonal matrix equals the infinity norm of its diagonal entries, we obtain,

$$\begin{aligned} \|H'(\mathbf{x}) - H(\mathbf{x})\|_F &\le \|D(g'(\mathbf{x}) - g(\mathbf{x}))\|_2\,\|U(\mathbf{x})\|_F \\ &= \|g'(\mathbf{x}) - g(\mathbf{x})\|_\infty\,\|U(\mathbf{x})\|_F. \end{aligned}$$

Finally, because each gate component is produced by a sigmoid nonlinearity, we have $g_i(\mathbf{x}) \in (0,1)$ for all channels i, implying $\delta(\mathbf{x}) < 1$. □

Remark 1 (Contrast with Unconstrained Adaptation)**.** *If the backbone were trainable, the feature drift would instead be bounded by $\|U'(\mathbf{x}) - U(\mathbf{x})\|_F$, which can be arbitrarily large since U' depends on all backbone parameters. By freezing the backbone and restricting updates to gates, we transform an unbounded representation shift into a multiplicatively bounded one. This is the fundamental mechanism underlying the stability of our approach.*

Having established that gated adaptation induces a bounded change in the intermediate feature maps, we now examine how this drift propagates through temporal pooling and the shared linear classifier. Ultimately, forgetting occurs when the logits associated with an input from a previously seen subject shift enough to alter the predicted label.

Theorem 2 (Bounded Logit Drift)**.** *Let $U(\mathbf{x}) \in \mathbb{R}^{C\times d}$ denote the ungated backbone feature map of temporal length d. Let*

$$H(\mathbf{x}) = D(g(\mathbf{x}))\,U(\mathbf{x}), \qquad H'(\mathbf{x}) = D(g'(\mathbf{x}))\,U(\mathbf{x}),$$

be the gated representations before and after learning a new task. Define the pooled vectors

$$h(\mathbf{x}) = \frac{1}{d}\,H(\mathbf{x})\mathbf{1}, \qquad h'(\mathbf{x}) = \frac{1}{d}\,H'(\mathbf{x})\mathbf{1},$$

and let the classifier change from W to W', with predictions

$$\hat{y}(\mathbf{x}) = Wh(\mathbf{x}), \qquad \hat{y}'(\mathbf{x}) = W'h'(\mathbf{x}).$$

Then the logit drift decomposes as

$$\|\hat{y}'(\mathbf{x}) - \hat{y}(\mathbf{x})\|_2 \leq \overbrace{\|W'\|_2 \cdot \frac{\delta(\mathbf{x})}{\sqrt{d}} \|U(\mathbf{x})\|_F}^{\text{gate-induced drift}} + \underbrace{\|W' - W\|_2 \|h(\mathbf{x})\|_2}_{\text{classifier-induced drift}}, \tag{1}$$

where $\delta(\mathbf{x}) = \|g'(\mathbf{x}) - g(\mathbf{x})\|_\infty < 1$.

Proof. From Theorem 1, the gated feature drift satisfies

$$\|H'(\mathbf{x}) - H(\mathbf{x})\|_F \leq \delta(\mathbf{x}) \|U(\mathbf{x})\|_F. \tag{2}$$

Temporal average pooling is a linear operator. For $v = \frac{1}{d} M\mathbf{1}$ where $M \in \mathbb{R}^{C \times d}$, we have $\|v\|_2 \leq \frac{1}{\sqrt{d}} \|M\|_F$ by Cauchy-Schwarz. Thus

$$\|h'(\mathbf{x}) - h(\mathbf{x})\|_2 \leq \frac{1}{\sqrt{d}} \|H'(\mathbf{x}) - H(\mathbf{x})\|_F \leq \frac{\delta(\mathbf{x})}{\sqrt{d}} \|U(\mathbf{x})\|_F. \tag{3}$$

Decomposing the logit difference:

$$\begin{aligned} \hat{y}'(\mathbf{x}) - \hat{y}(\mathbf{x}) &= W'h'(\mathbf{x}) - Wh(\mathbf{x}) \\ &= W'\big(h'(\mathbf{x}) - h(\mathbf{x})\big) + \big(W' - W\big) h(\mathbf{x}). \end{aligned}$$

The result follows from the triangle inequality and submultiplicativity. □

The decomposition in (1) reveals that logit drift has two independent sources: gate changes and classifier changes. This separation is crucial. It shows that even when the classifier must adapt significantly for a new task, the gate-induced drift on old tasks remains controlled by $\delta(\mathbf{x})$.

Having bounded the logit drift, we now relate this perturbation to decision stability.

Theorem 3 (Sufficient Condition for Prediction Stability)**.** *Let $\hat{y}(\mathbf{x})$ and $\hat{y}'(\mathbf{x})$ denote the logits of an earlier-task sample $\mathbf{x}$ before and after learning a new task. Let the classification margin be*

$$m(\mathbf{x}) = \hat{y}_y(\mathbf{x}) - \max_{k \neq y} \hat{y}_k(\mathbf{x}),$$

where y is the true label. If

$$\|\hat{y}'(\mathbf{x}) - \hat{y}(\mathbf{x})\|_\infty < \frac{m(\mathbf{x})}{2},$$

then the predicted label is preserved: $\arg\max_k \hat{y}'_k(\mathbf{x}) = y$.

Proof. Let $k^* = \arg\max_{k \neq y} \hat{y}_k(\mathbf{x})$ be the runner-up class. By the margin definition and the perturbation bound:

$$\hat{y}'_y(\mathbf{x}) \geq \hat{y}_y(\mathbf{x}) - \|\hat{y}' - \hat{y}\|_\infty > \hat{y}_y(\mathbf{x}) - \frac{m(\mathbf{x})}{2},$$

$$\hat{y}'_{k^*}(\mathbf{x}) \leq \hat{y}_{k^*}(\mathbf{x}) + \|\hat{y}' - \hat{y}\|_\infty < \hat{y}_{k^*}(\mathbf{x}) + \frac{m(\mathbf{x})}{2}.$$

Since $\hat{y}_y(\mathbf{x}) - \hat{y}_{k^*}(\mathbf{x}) = m(\mathbf{x})$, we have $\hat{y}'_y(\mathbf{x}) > \hat{y}'_{k^*}(\mathbf{x})$, preserving the prediction. □

Combining Theorems 2 and 3, we obtain an explicit sufficient condition for zero forgetting:

Corollary 1 (Margin-Based Forgetting Guarantee)**.** *For a sample $\mathbf{x}$ from a previous task with margin $m(\mathbf{x}) > 0$, the prediction is preserved if*

$$\|W'\|_2 \cdot \frac{\delta(\mathbf{x})}{\sqrt{d}} \|U(\mathbf{x})\|_F + \|W' - W\|_2 \|h(\mathbf{x})\|_2 < \frac{m(\mathbf{x})}{2}.$$

This condition reveals the interplay between representation quality and forgetting: samples with larger margins (more confident predictions) and smaller feature norms are more robust to task interference.

B. Expressiveness of Diagonal Gating

While Theorems 1–3 establish the boundary for decision stability, they raise a fundamental question of plasticity: is the constrained space of diagonal gating expressive enough to achieve high accuracy on new subjects?

In domain-incremental HAR, subject variability typically manifests through several mechanisms that align well with channel-wise modulation:

- **Biomechanical differences**: Variations in limb length, body mass, and movement patterns affect the amplitude of accelerometer and gyroscope signals in a sensor-specific manner.
- **Sensor placement variability**: Even with standardized protocols, slight differences in sensor attachment result in consistent per-channel scaling and offset effects.
- **Device heterogeneity**: Different sensor models or calibration states introduce multiplicative gains that are channel-specific.

These observations motivate the following structural assumption:

Assumption 1 (Channel-wise Domain Shift)**.** *For input $\mathbf{x}$ from subject t, the backbone features satisfy*

$$U(\mathbf{x}) \approx D(s_t)\, \bar{U}(\mathbf{x}) + \varepsilon_t(\mathbf{x}),$$

where $\bar{U}(\mathbf{x})$ represents a canonical (subject-agnostic) feature map, $s_t \in \mathbb{R}^C_{>0}$ captures subject-specific channel scaling, and $\varepsilon_t(\mathbf{x})$ is a residual with $\|\varepsilon_t(\mathbf{x})\|_F \ll \|U_t(\mathbf{x})\|_F$.

Remark 2 (Empirical Support for Assumption 1)**.** *To validate this assumption, we analyzed the cross-subject correlation structure of backbone features on PAMAP2. For each subject, we computed the mean 512-dimensional feature vector (centroid) for each activity class using the frozen pretrained backbone. For each pair of subjects sharing K common activities, we then computed a $C \times C$ correlation matrix where entry (c, c') is the Pearson correlation between channel c of subject i and channel c' of subject j across the K paired centroids. Figure 4 shows the average correlation matrix across all 28 subject pairs.*

The resulting matrix exhibits a pronounced diagonal structure: the mean diagonal correlation is 0.78, indicating that the same channel responds similarly across subjects when performing the same activity. In contrast, the mean off-diagonal correlation is approximately zero, suggesting that different channels do not systematically covary across subjects. This pattern confirms that cross-subject variability

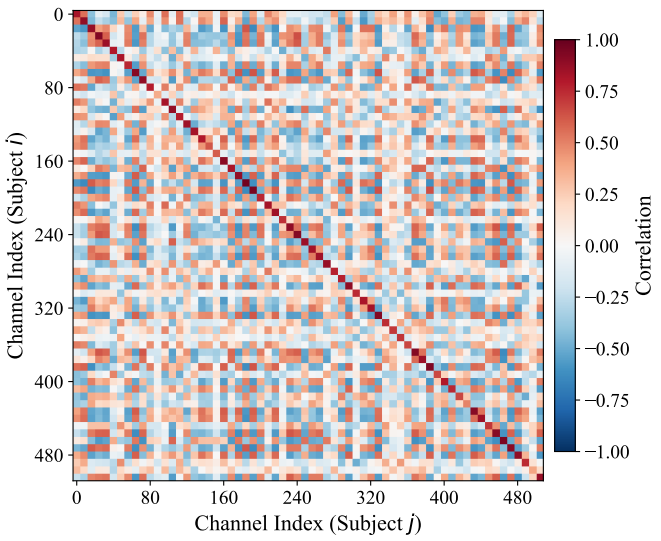

Fig. 4. Cross-subject feature channel correlation matrix averaged over all subject pairs in PAMAP2. Entry (c, c') shows the Pearson correlation between channel c of subject i and channel c' of subject j, computed using activity-class centroids as paired observations. The strong diagonal (mean $\rho = 0.78$) indicates that channels preserve their identity across subjects, while the weak off-diagonal (mean $\rho \approx 0$) suggests minimal cross-channel mixing. This supports Assumption 1: cross-subject variation is predominantly channel-wise.

in the learned feature space is predominantly channel-wise rather than involving complex cross-channel interactions.

Under this assumption, diagonal gating is provably sufficient for adaptation:

Theorem 4 (Expressiveness of Diagonal Gating)**.** *Under Assumption 1, for each subject t there exists a gate vector $g_t \in (0,1)^C$ and a global scalar $\alpha_t > 0$ such that the gated representation,*

$$H_t(\mathbf{x}) = D(g_t)\, U(\mathbf{x})$$

satisfies,

$$\|H_t(\mathbf{x}) - \alpha_t^{-1} U_t(\mathbf{x})\|_F \le \alpha_t^{-1} \|\varepsilon_t(\mathbf{x})\|_F$$

for all $\mathbf{x}$*. When $\varepsilon_t = 0$, the match is exact up to global scaling.*

Proof. Choose $\alpha_t > \|s_t\|_\infty$ (e.g., $\alpha_t = (1+\epsilon)\|s_t\|_\infty$ for small $\epsilon > 0$) and define $g_t = s_t/\alpha_t$. Then $g_t \in (0,1)^C$ and

$$D(g_t)\, U(\mathbf{x}) = \frac{1}{\alpha_t} D(s_t)\, U(\mathbf{x}).$$

In the exact case ($\varepsilon_t = 0$), we have $U_t(\mathbf{x}) = D(s_t)U(\mathbf{x})$, so $H_t(\mathbf{x}) = \alpha_t^{-1} U_t(\mathbf{x})$ exactly.

For the approximate case:

$$\begin{aligned} H_t(\mathbf{x}) - \alpha_t^{-1} U_t(\mathbf{x}) &= \frac{1}{\alpha_t} D(s_t) U(\mathbf{x}) \\ &\quad - \frac{1}{\alpha_t}\big(D(s_t)U(\mathbf{x}) + \varepsilon_t(\mathbf{x})\big) \\ &= -\frac{1}{\alpha_t}\varepsilon_t(\mathbf{x}). \end{aligned}$$

Taking norms yields the stated bound. □

Remark 3 (Invariance to Global Scaling)**.** *The factor α_t^{-1} is absorbed by the subsequent linear classifier without affecting decision boundaries. Specifically, if W achieves correct classification on $U_t(\mathbf{x})$, then $\alpha_t W$ achieves the same on $H_t(\mathbf{x}) = \alpha_t^{-1} U_t(\mathbf{x})$. This degree of freedom allows diagonal gating to handle arbitrary positive scaling differences across subjects.*

In summary, this section examined why channel-wise gating provides a stable and effective mechanism for continual adaptation in domain-incremental HAR. By freezing the backbone and restricting updates to diagonal, bounded scaling of intermediate features, the model limits representational drift and protects knowledge from earlier subjects. At the same time, these gates are expressive enough to account for the dominant forms of cross-subject variability, enabling meaningful adaptation without introducing the destructive interference commonly observed with more flexible adapter layers. This structure naturally supports a favorable stability–plasticity trade-off, which aligns with the empirical performance observed in our experiments.

A key limitation of this analysis is its reliance on domain shifts that are approximately channel-wise, an assumption that holds well in HAR but may not generalize to domains with richer cross-channel interactions. Additionally, the stability guarantees depend on non-trivial prediction margins, which can vary across activities and subjects. While the gating mechanism performs well under typical HAR conditions, more expressive forms of adaptation may be needed when feature shifts deviate significantly from this structure.

V. Experimental Setup

In this section, we describe the HAR benchmark datasets we used, followed by the implementation details and system configuration. We will also define the evaluation metrics used to assess our experiments, details of which will be elaborated in the next section.

A. Benchmarks

1) PAMAP2 Physical Activity Monitoring [13]*:* The PAMAP2 dataset contains inertial measurements from 8 subjects performing 12 predefined physical activities. Although the original release includes 6 additional optional activities, these were excluded from our evaluation due to inconsistent coverage across participants. The dataset provides accelerometer, gyroscope, and magnetometer signals, totaling 9 synchronized sensor channels sampled at 100 Hz.

2) Daily and Sports Activities [47]*:* The Daily and Sports Activities (DSA) dataset comprises recordings from 8 subjects executing 18 activities of daily living and sport-related motions. The dataset includes accelerometer, gyroscope, and magnetometer signals collected across 15 channels, sampled at 25 Hz.

3) UCI Human Activity Recognition Using Smartphones [48]*:* The UCI HAR dataset consists of motion sensor recordings from 30 participants performing 6 common activities. It provides accelerometer and gyroscope measurements across 9 sensor channels, sampled at 50 Hz. The dataset includes both raw signals and preprocessed segments commonly used in human activity recognition research.

4) RealWorld HAR [49]*:* The RealWorld dataset contains recordings from 15 subjects performing 8 activities. The subjects span a broad demographic range in age, height, and weight, and the data were collected in everyday, in-the-wild settings rather than a fixed laboratory protocol. Each subject wore sensors at seven body locations simultaneously, sampled at 50 Hz.

Relative to the other three benchmarks, which are collected under more controlled conditions, RealWorld is recorded during free, unscripted execution of each activity and exhibits greater variability in subject demographics and in how sensors are worn and oriented across the body. This makes it a useful complement to the laboratory-style datasets, providing a less controlled setting in which the marginal distribution of the sensor signals departs more substantially across subjects.

B. Implementation Details

1) Model Architecture: Our CNN architecture consists of an initial 1×1 convolutional embedding layer that projects input channels to 256 dimensions, followed by four residual stacks with progressive channel expansion and temporal downsampling. Table I details the layer configurations.

TABLE I
CNN ARCHITECTURE WITH 4 RESIDUAL STACKS. EACH STACK CONTAINS MULTIPLE CONVOLUTIONAL LAYERS WITH BATCH NORMALIZATION AND RELU ACTIVATIONS.

Stack	Channels	Layers	Stride
Embedding	$C_0 \to 256$	Conv1d (k=1)	1
Stack 1	256	4 conv layers (k=1,3,3,1)	1
Stack 2	$256 \to 384$	4 conv layers (k=1,5,3,1)	1
Stack 3	$384 \to 512$	4 conv layers (k=1,3,3,1)	2
Stack 4	512	4 conv layers (k=1,3,3,1)	2
Pooling	512	AdaptiveAvgPool1d	–
Classifier	$512 \to K$	Linear	–

Each stack follows a bottleneck design: 1×1 compression, large-kernel feature extraction (3×3 or 5×5), 1×1 expansion, with residual connections around each stack and batch normalization after every convolutional layer. Global average pooling aggregates the final 512-channel feature map into a single vector, followed by a linear classifier. The total model contains approximately 7.4 million parameters.

2) Pre-training Procedure: We pre-trained the CNN backbone on the WISDM dataset [50], a smartphone-based HAR dataset with 18 activity classes collected from 51 subjects. Raw tri-axial accelerometer data sampled at 20 Hz was segmented into fixed-length windows of 200 timesteps (10 seconds) with 50% overlap (step size of 100 timesteps). Per-subject z-score normalization was applied independently to each axis to account for inter-subject variability in sensor calibration and placement.

The model was trained for 300 epochs using the Adam optimizer with an initial learning rate of 0.001, cosine annealing schedule decaying to 10^{-6}, mild weight decay of 0.0001, gradient clipping with max norm 1.0, and batch size 64. Cross-entropy loss was used, and the best model checkpoint based on validation accuracy was saved. Early stopping with patience of 20 epochs prevented overfitting.

Once the pre-training is done, after each stack's residual block and batch normalization, we optionally insert a channel gate module. Each gate computes a channel-wise scaling vector $g \in (0,1)^C$ via global average pooling, a two-layer bottleneck MLP with reduction ratio $r = 8$ (hidden dimension $\max(C/8, 16)$), ReLU activation, and sigmoid output. The gate is applied via element-wise multiplication. For a 512-channel stack, each gate contains $(512 \times 64) + (64 \times 512) = 65{,}536$ parameters.

3) Data Preprocessing: For each target dataset, raw sensor data was segmented using a sliding window with 50% overlap. Window sizes matched the temporal resolution of each dataset: UCI-HAR used 128 timesteps (2.56s at 50 Hz), PAMAP2 used 200 timesteps (2s at 100 Hz), DSA used 125 timesteps (5s at 25 Hz), and RealWorld used 200 timesteps (4s at 50 Hz). Per-subject z-score normalization was applied independently to each channel before windowing. To match the pretrained model's expected input length (200 timesteps), windows were resized via linear interpolation.

Each subject defines one task in the continual learning sequence. Within each subject's data, an 80/20 stratified train/test split by activity class prevents data leakage. Subjects are presented in random order, with 10 different random permutations tested per experiment to account for task-order variability.

4) Training Procedure: For each subject (task), we train only the gate parameters (if present) and classifier while keeping the backbone frozen. Training proceeds for up to 100 epochs with early stopping. We use the Adam optimizer with learning rate 0.001, L2 regularization ($\lambda = 0.0001$), batch size 64, and gradient clipping (max norm 1.0). After training on subject t, the model is evaluated on test sets from all subjects $1, \ldots, t$ to measure forgetting. No data replay or task-specific regularization is used, and stability emerges purely from the frozen backbone and bounded gate updates.

C. Evaluation Metrics

1) Final Accuracy (FA): This metric evaluates the model's overall performance across all subjects after sequential training has been completed. It directly reflects the model's ability to retain what it has learned throughout the entire training process.

$$\text{FA} = \frac{1}{T} \sum_{t=1}^{T} A_{t,T} \tag{4}$$

where $A_{t,T}$ is the model's test accuracy on subject t after it has been trained on all T subjects.

2) Forgetting Measure (FM): This metric quantifies the degree of forgetting that occurs over training. After completing training on all subjects, FM captures how much the model's performance on earlier subjects has degraded relative to its best performance during training.

$$\text{FM} = \frac{1}{T} \sum_{t=1}^{T} \left(\max_{t' \in \{1,\ldots,T\}} A_{t,t'} - A_{t,T} \right) \tag{5}$$

where $A_{t,t'}$ denotes the test accuracy on subject t after training up through subject t'.

3) Learning Accuracy (LA): This metric evaluates the model's plasticity, measuring how effectively it acquires new information when each subject is introduced. LA reflects the model's immediate performance on a newly encountered subject before any interference from subsequent tasks.

$$\text{LA} = \frac{1}{T}\sum_{t=1}^{T} A_{t,t} \tag{6}$$

where $A_{t,t}$ denotes the test accuracy on subject t immediately after the model has been trained on that same subject.

VI. Experimental Results

In this section, we present the different types of experiments we conducted and their results. Our experiments include comparing our method against the baseline algorithms, as well as ablation studies that quantify the contribution of each component of the proposed method.

A. Comparison with Replay-Free Baselines

We first evaluate our approach against continual learning methods that, like ours, do not store past samples. This comparison isolates the effectiveness of different adaptation mechanisms under identical memory constraints. All baseline methods use a trainable backbone initialized from the same WISDM-pretrained weights, following their original formulations.

Regularization-based methods. Elastic Weight Consolidation (EWC) [14] penalizes changes to parameters deemed important for previous tasks, using the Fisher information matrix to estimate importance. Learning without Forgetting (LwF) [24] preserves prior knowledge through knowledge distillation, using the previous model's outputs as soft targets during training on new tasks.

Architecture-based methods. Hard Attention to Task (HAT) [19] learns binary attention masks over network units, with each task receiving a dedicated mask that protects important units from modification during subsequent learning.

Table II summarizes the results. Our gating approach consistently achieves the highest final accuracy and lowest forgetting across all four benchmarks among replay-free methods. On PAMAP2, we outperform the next-best method (HAT) by 9.9 percentage points in final accuracy while reducing forgetting by 12.3 points. Notably, our approach also exhibits substantially lower variance across task orderings (std of 2.5% vs 6–10% for baselines), indicating more robust performance regardless of the order in which subjects are encountered. These results confirm that structured diagonal adaptation provides a more effective stability-plasticity tradeoff than either regularization-based constraints or binary masking.

B. Comparison with Replay-Based Methods

Replay-based methods represent a fundamentally different tradeoff: they achieve stronger performance by maintaining a memory buffer of past samples, incurring additional storage costs and, in the case of sensor data, potential privacy implications. We include this comparison to contextualize our results within the broader continual learning landscape.

Table III presents results for Dark Experience Replay (DER) [17] and its extension DER++, both using a buffer of 500 samples. As expected, replay-based methods achieve higher final accuracy than replay-free approaches, with DER++ reaching 90.1% on PAMAP2 compared to 77.7% for our method. However, this performance gap must be weighed against several practical considerations, which we now quantify directly rather than describe qualitatively.

First, replay methods require training the full backbone, updating 100% of model parameters per task compared to only 2.5% for our approach. As reported in Table IV, this difference is concrete: our method updates roughly 0.19M parameters per task versus 7.44M for full-backbone methods. Because no gradient flows through the frozen backbone, the measured per-step training cost falls to 0.60× that of full-backbone training and peak training memory drops by about a quarter (e.g., 561 MB vs. 764 MB on PAMAP2), since optimizer state and activation gradients are not maintained for the backbone. All methods share an identical inference forward pass (1.67 GFLOPs per window), so these savings are in training cost rather than inference latency.

Second, storing 500 sensor samples requires persistent on-device memory that scales with buffer size. Concretely, each buffered window is the raw multi-sensor segment together with its label and stored logits; at a 500-sample buffer this amounts to 10.3 MB on PAMAP2, 17.2 MB on DSA, and 2.3 MB on UCI-HAR (Table IV). For resource-constrained IoT devices such as smartwatches or fitness bands, this memory overhead may be prohibitive, particularly when multiple applications compete for limited storage.

Third, and perhaps most critically for IoT deployments, retaining raw accelerometer and gyroscope data raises privacy concerns. Movement patterns can reveal sensitive information about health conditions, daily routines, and behavioral profiles [12]. Because the buffer stores raw sensor windows rather than compressed embeddings, this storage cost is simultaneously a privacy exposure: every retained window is fully reconstructable motion data. For applications such as elderly care monitoring or remote patient tracking, such data retention may conflict with privacy regulations or user expectations. Our replay-free approach achieves strong performance without storing any historical data, making it suitable for privacy-sensitive edge deployments.

Taken together, these measurements make explicit when the accuracy–retention trade-off favors each approach, and also delimit the deployment assumptions of our framework. Replay buys higher final accuracy (e.g., +12 points on PAMAP2) at the cost of persistently retained raw sensor data; it is the better choice where storage and raw-data retention are unconstrained and peak accuracy is paramount. Our replay-free method is preferable in the opposite regime (tight storage budgets, constrained training energy, or restrictions on raw-sensor retention from privacy regulation) which are precisely the defining conditions of on-device wearable deployment. We emphasize, however, that this trade-off is contingent on those

TABLE II
COMPARISON WITH REPLAY-FREE CONTINUAL LEARNING METHODS (MEAN ± STD OVER 10 PERMUTATIONS). FA: FINAL ACCURACY (%), FM: FORGETTING MEASURE (%), LA: LEARNING ACCURACY (%)

	PAMAP2 8 subj., 12 act.			**DSA** 8 subj., 18 act.			**UCI-HAR** 30 subj., 6 act.			**RealWorld** 15 subj., 8 act.		
	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑
EWC	66.2±7.5	29.8±7.5	95.6±0.6	68.5±5.2	30.1±5.4	98.5±0.3	76.2±8.5	22.3±8.8	98.9±0.5	56.4±8.1	38.6±8.0	89.9±0.7
LwF	64.8±10.0	31.3±10.3	96.8±0.5	69.8±6.0	29.2±5.6	99.0±0.3	74.7±11.3	23.6±11.7	98.4±0.7	55.2±9.6	39.5±9.8	90.5±0.6
HAT	67.8±8.2	28.5±8.4	96.6±0.4	70.2±5.8	28.3±5.6	99.1±0.2	77.5±6.2	21.0±6.4	98.7±0.4	57.9±7.0	36.8±7.2	90.6±0.5
Ours	77.7±2.5	16.2±2.6	93.8±0.4	78.7±1.7	19.6±1.7	98.3±0.3	80.3±5.3	14.5±5.4	94.9±1.2	65.6±3.2	26.4±3.3	86.5±0.8
Upper Bound	92.8±0.9	0.0±0.0	92.8±0.9	97.4±0.6	0.0±0.0	97.4±0.6	94.1±0.6	0.0±0.0	94.1±0.6	79.9±0.9	0.0±0.0	79.9±0.9

TABLE III
COMPARISON WITH REPLAY-BASED METHODS (MEAN ± STD OVER 10 PERMUTATIONS). PARAMS: PERCENTAGE OF MODEL PARAMETERS UPDATED PER TASK. BUFFER: NUMBER OF STORED SAMPLES.

	PAMAP2			**DSA**			**UCI-HAR**			**RealWorld**		
	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑
DER	88.5±0.6	8.1±0.8	95.6±0.4	92.3±1.8	6.9±1.8	98.2±0.2	89.5±1.9	9.8±2.0	98.4±0.2	75.6±1.4	16.3±1.5	90.4±0.4
DER++	90.1±1.6	6.3±1.4	95.4±0.4	94.1±1.7	5.0±1.6	98.0±0.2	90.5±2.6	8.8±2.6	99.4±0.2	77.1±1.9	14.4±1.8	90.7±0.3
Ours	77.7±2.5	16.2±2.6	93.8±0.4	78.7±1.7	19.6±1.7	98.3±0.3	80.3±5.3	14.5±5.4	94.9±1.2	65.6±3.2	26.4±3.3	86.5±0.8

conditions rather than universal: in centralized or privacy-permissive settings where a buffer is acceptable, replay (or the hybrid of Section VI-H3) remains the stronger option. The framework also assumes that subject-induced shift is approximately channel-wise and that a sufficiently diverse pretrained backbone is available; where these assumptions weaken (for example under large cross-channel distribution shift, or when the source and target domains differ substantially) the advantage of frozen, diagonally gated adaptation may narrow, and more expressive adaptation could be required.

We note that gating and replay address forgetting through complementary mechanisms: one architectural, one data-driven. In the following subsections, we will explore a hybrid approach that combines our gating mechanism with experience replay, demonstrating that these techniques can be effectively integrated when deployment constraints permit data retention.

C. Efficiency, Storage, and Deployment Cost

To support the efficiency and privacy claims above with concrete numbers rather than qualitative argument, we measured the per-task cost of each method family on identical hardware. Trainable parameters, retained storage, and inference FLOPs are exact and hardware-independent; training time and peak memory are inherently device-specific, so we report training time as a ratio to full-backbone training measured on the same device, which transfers across platforms more reliably than absolute timings. Table IV summarizes the results. Our gated method updates only about 2.5% of parameters, retains no data, trains at roughly 0.60× the per-step cost of full-backbone methods, and reduces peak training memory by about a quarter, while sharing the identical inference forward pass of all methods. The retained-storage column also quantifies the privacy exposure of replay discussed above: between 2.3 and 17.2 MB of raw, fully reconstructable sensor windows per device at a 500-sample buffer, versus zero for our approach.

D. Effect of Backbone Freezing

A central design choice in our approach is the use of a frozen pretrained backbone. To assess the impact of this decision, we compare models with trainable versus frozen backbones, both with and without gating mechanisms. Table V presents results across all four benchmarks.

a) Freezing without gates.: Comparing *Base* (trainable backbone, no gates) to *Pretrained* (frozen backbone, no gates), we observe that freezing the backbone consistently reduces forgetting, demonstrating that a stable feature representation limits catastrophic interference. On PAMAP2, FM drops dramatically from 39.7% to 17.5%, a 56% relative reduction. However, this stability comes at the cost of reduced learning accuracy: LA drops from 96.5% to 93.9%, as the frozen backbone cannot fully adjust to subject-specific variations. Despite this, the net effect on final accuracy is strongly positive: FA improves from 56.7% to 76.5% due to the substantial reduction in forgetting. We note that this favorable balance presumes the pretrained representation transfers reasonably to the target subjects; under a pronounced source–target mismatch the frozen backbone would forgo more plasticity than gating can recover, and partial unfreezing or a more expressive adapter could become preferable.

b) Freezing with gates.: When gating mechanisms are introduced, the benefits compound. Comparing *Base + Gates* (trainable backbone with gates) to *Pretrained + Gates* (frozen backbone with gates), we find that freezing still reduces forgetting (FM drops from 31.3% to 16.2% on PAMAP2), and final accuracy improves substantially (FA increases from 64.9% to 77.7%). Gates provide the frozen backbone with targeted adaptation capacity, allowing it to match or exceed

TABLE IV
EFFICIENCY, STORAGE, AND DEPLOYMENT COST. TRAINABLE PARAMETERS AND RETAINED STORAGE ARE EXACT; INFERENCE FLOPS ARE IDENTICAL ACROSS METHODS (SAME FORWARD PASS); TRAINING-STEP TIME IS REPORTED RELATIVE TO FULL-BACKBONE TRAINING, AND PEAK TRAINING MEMORY IS MEASURED ON THE SAME GPU. RETAINED STORAGE IS GIVEN PER DATASET AT A 500-SAMPLE BUFFER.

Method	Trainable params	Train step (rel.)	Inf. FLOPs /window	Peak train mem. (rel.)	Retained storage (MB) PAMAP2	DSA	UCI	RealWorld
Ours (gates, frozen)	0.19 M (2.5%)	0.60×	1.67 G	∼0.74×	0	0	0	0
Full-backbone (EWC, HAT)	7.44 M (100%)	1.0×	1.67 G	1.0×	0	0	0	0
Replay (DER, DER++)	7.44 M (100%)	1.0×	1.67 G	1.0×	10.3	17.2	2.3	10.3
Gates + DER	0.19 M (2.5%)	0.60×	1.67 G	∼0.74×	10.3	17.2	2.3	10.3

Per-sample buffer cost: 21.2 / 35.2 / 4.7 / 21.1 KB for PAMAP2 / DSA / UCI-HAR / RealWorld. EWC and HAT retain no samples but store auxiliary state (Fisher matrix / task masks).

TABLE V
EFFECT OF BACKBONE FREEZING AND GATING MECHANISMS ACROSS HAR BENCHMARKS (MEAN ± STD OVER 10 PERMUTATIONS).

	PAMAP2 8 subj., 12 act.			**DSA** 8 subj., 18 act.			**UCI-HAR** 30 subj., 6 act.			**RealWorld** 15 subj., 8 act.		
	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑
Trainable Backbone												
Base	56.7±8.9	39.7±8.9	96.5±0.3	68.7±3.9	30.4±4.1	99.1±0.3	75.6±8.6	23.3±8.8	98.9±0.6	51.4±7.8	40.9±7.9	90.6±0.4
Base + Gates	64.9±10.0	31.3±10.3	96.2±0.5	69.8±6.0	29.2±5.9	99.0±0.3	74.7±11.3	23.7±11.7	98.4±0.7	56.3±9.1	36.6±9.3	90.3±0.5
Frozen Backbone												
Pretrained	76.5±4.0	17.5±4.0	93.9±0.4	75.7±2.7	22.1±2.7	97.9±0.2	80.1±9.4	13.9±9.5	94.0±1.5	64.4±3.6	27.3±3.6	86.2±0.9
Pretrained + Gates (Ours)	77.7±2.5	16.2±2.6	93.8±0.4	78.7±1.7	19.6±1.7	98.3±0.3	80.3±5.3	14.5±5.4	94.9±1.2	65.6±3.2	26.4±3.3	86.5±0.8

the plasticity of trainable backbones while maintaining the stability advantages of frozen representations.

E. Effect of Gating Mechanisms

We next isolate the contribution of channel-wise gates by comparing models with and without gating, holding the backbone training strategy fixed. Results are shown in Table V.

a) Gates on trainable backbones.: Comparing *Base* (no gates) to *Base + Gates*, we observe meaningful improvement from gating even when the backbone is trainable. On PAMAP2, FA improves from 56.7% to 64.9%, and FM decreases from 39.7% to 31.3%. This suggests that gates provide useful regularization even when the entire network can adapt, though the effect is more modest than with frozen backbones.

b) Gates on frozen backbones.: The effect of gating becomes most pronounced when the backbone is frozen. Comparing *Pretrained* (no gates) to *Pretrained + Gates*, we see improvements in final accuracy on PAMAP2 (FA increases from 76.5% to 77.7%) while maintaining low forgetting. This demonstrates that gates provide the critical adaptation capacity needed when the backbone is frozen, enabling subject-specific feature modulation without disrupting learned representations.

c) Gates reduce variance.: Beyond mean performance, we observe that gating reduces variability across task orderings, particularly on PAMAP2, where the standard deviation of FA decreases from 8.9% (Base) to 2.5% (Pretrained + Gates). This suggests that gates help stabilize learning across different subject sequences.

F. Gating vs. Stacked Trainable Layers

A central theoretical claim of our work is that gated adaptation, which performs feature *selection* through channel-wise magnitude scaling, offers superior stability compared to stacked trainable layers, which perform feature *generation* through arbitrary linear transformations. To test this hypothesis, we compare our gating approach against architectures with varying numbers of fully connected layers inserted between the frozen backbone and classifier. Each stacked layer consists of a linear transformation (512 → 512), ReLU activation, and dropout (0.3).

Table VI presents results across all four benchmarks. On PAMAP2, we observe a clear pattern as the number of stacked layers increases: learning accuracy (LA) improves monotonically from 93.9% (no layer) to 95.3% (3 layers), indicating that additional capacity helps the model fit each new subject. However, forgetting (FM) also increases substantially from 17.5% to 25.3%, demonstrating that this added capacity comes at a steep cost to stability. The net effect on final accuracy is negative: FA drops from 76.5% to 70.1% as forgetting outweighs the plasticity gains.

In contrast, our gating approach achieves FM of only 16.2% while maintaining competitive FA of 77.7%, striking a balance between stability and plasticity. This pattern holds across datasets. On DSA, three stacked layers reach FM of 25.6% compared to 19.6% for gates, while achieving lower FA (73.0% vs 78.7%). On UCI-HAR, the gap is similarly pronounced.

a) Capacity is not the answer.: An important observation from these results is that simply adding more trainable parameters does not reduce forgetting; in fact, it exacerbates it. The stacked 3-layer architecture contains significantly more parameters than our gating mechanism, yet performs substantially worse on forgetting. This confirms that the *structure* of adaptation matters more than the sheer number of parameters, and that bounded, interpretable transformations (diagonal scaling) are preferable to unconstrained transformations (dense layers) in continual learning settings.

TABLE VI
COMPARISON OF GATING (SELECTION) VERSUS STACKED FULLY CONNECTED LAYERS (GENERATION) ACROSS BENCHMARKS (MEAN ± STD OVER 10 PERMUTATIONS). STACKED LAYERS ARE 512→512 WITH RELU AND DROPOUT.

	PAMAP2 8 subj., 12 act.			**DSA** 8 subj., 18 act.			**UCI-HAR** 30 subj., 6 act.			**RealWorld** 15 subj., 8 act.		
	FA	FM	LA	FA	FM	LA	FA	FM	LA	FA	FM	LA
Frozen Backbone with Stacked Adaptation Layers												
No stacked layers	76.5±4.0	17.5±4.0	93.9±0.4	75.7±2.7	22.1±2.7	97.9±0.2	80.1±9.4	13.9±9.5	94.0±1.5	64.4±3.6	27.3±3.6	86.2±0.9
1 stacked layer	74.2±4.8	20.1±4.9	94.3±0.4	74.8±3.2	23.5±3.3	98.2±0.3	78.9±5.8	16.2±5.8	95.1±1.1	62.8±4.1	29.6±4.2	87.0±0.8
2 stacked layers	72.1±6.2	22.7±6.3	94.8±0.4	73.9±4.3	24.6±4.5	98.4±0.3	78.1±6.1	18.0±6.1	96.1±1.1	61.5±5.2	31.1±5.3	87.6±0.8
3 stacked layers	70.1±7.7	25.3±7.8	95.3±0.4	73.0±5.3	25.6±5.5	98.6±0.3	77.3±6.4	19.7±6.4	97.0±1.2	60.2±6.4	32.9±6.5	88.1±0.9
Gates (Ours)	77.7±2.5	16.2±2.6	93.8±0.4	78.7±1.7	19.6±1.7	98.3±0.3	80.3±5.3	14.5±5.4	94.9±1.2	65.6±3.2	26.4±3.3	86.5±0.8

TABLE VII
EFFECT OF KNOWLEDGE DISTILLATION, TASK-AWARE GATES, AND EXPERIENCE REPLAY (MEAN ± STD OVER PERMUTATIONS).

	PAMAP2 8 subj., 12 act.			**DSA** 8 subj., 18 act.			**UCI-HAR** 30 subj., 6 act.			**RealWorld** 15 subj., 8 act.		
	FA	FM	LA	FA	FM	LA	FA	FM	LA	FA	FM	LA
Task-Free Gates (Single Shared Gates)												
Without KD (Ours)	77.7±2.5	16.2±2.6	93.8±0.4	78.7±1.7	19.6±1.7	98.3±0.3	80.3±5.3	14.5±5.4	94.9±1.2	65.6±3.2	26.4±3.3	86.5±0.8
With KD	77.5±3.5	12.7±4.0	90.2±1.0	78.1±3.6	17.3±3.0	95.4±0.7	82.1±4.4	8.9±5.0	90.9±2.6	65.2±3.9	22.9±3.7	83.4±1.4
Task-Aware Gates (Separate Gates per Task)												
Task-Aware	78.9±4.4	12.3±4.2	91.2±0.6	80.1±3.0	16.4±2.9	96.5±0.5	82.9±3.0	8.9±1.8	91.8±2.3	66.8±3.3	21.9±3.0	84.3±1.1
Gates + Experience Replay (500 sample buffer)												
Gates + Replay	84.0±2.1	6.6±1.6	90.6±1.0	85.0±3.0	10.2±2.4	95.2±1.1	83.7±5.9	7.1±4.5	90.8±3.6	71.1±3.5	16.2±2.9	83.5±1.9
Gates + DER	84.3±2.0	6.1±1.5	90.4±0.9	85.7±2.6	9.6±2.1	95.3±1.0	85.0±5.0	6.1±2.6	91.1±4.0	71.8±3.1	15.3±2.4	83.7±2.0

G. Effect of the Number of Gates

Our default inserts a channel gate after every backbone stack. To verify that the stability–plasticity benefit comes from the bounded, diagonal form of the gating rather than from a particular amount of gating capacity, we vary the number of gates inserted into the frozen backbone, adding them to the first N stacks for $N \in \{0, 1, 2, 3, 4\}$. This mirrors the stacked-layer ablation of Table VI, but along the feature-*selection* axis instead of feature-*generation*; the $N{=}0$ and $N{=}4$ settings coincide with the *Pretrained* and *Pretrained + Gates (Ours)* configurations.

Table VIII reports the results across all four benchmarks. In contrast to stacking trainable layers, where adding capacity steadily increased forgetting and reduced final accuracy, varying the number of gates produces no such degradation: final accuracy stays within a narrow band across all gate counts, and the full configuration ($N{=}4$) attains the best or near-best final accuracy on every dataset while keeping forgetting low. Intermediate gate counts are competitive but do not consistently improve over the full configuration, indicating that the advantage of the mechanism is not sensitive to a finely tuned number of gates. These results support our main claim that bounded channel-wise selection, rather than added adaptation capacity per se, is what yields a favorable stability–plasticity trade-off.

H. Improvement Strategies

Having established that task-free gates with frozen backbones provide superior stability-plasticity tradeoffs, we now examine three extensions: (1) knowledge distillation to further reduce forgetting, (2) task-aware gates that leverage task identity at test time, and (3) combining our approach with experience replay.

1) Knowledge Distillation: Knowledge distillation (KD) [51] is a widely used technique in continual learning that preserves the model's predictions on previous tasks by matching the output distributions of the updated model to those of the model before the update. We augment our gating approach with KD loss:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{CE}}(y, \hat{y}) + (1 - \alpha)\mathcal{L}_{\text{KD}}(\hat{y}, \hat{y}_{\text{old}}),$$

where $\mathcal{L}_{\text{CE}}$ is cross-entropy on the current task, $\mathcal{L}_{\text{KD}}$ is the KL divergence between current and previous model outputs, and $\alpha = 0.5$ balances the two objectives.

Table VII shows that KD consistently reduces forgetting across all datasets. On PAMAP2, KD brings FM from 16.2% down to 12.7%, a 22% relative reduction. However, this comes at the cost of reduced plasticity: LA drops from 93.8% to 90.2%, indicating that the model becomes more conservative in adapting to new subjects. The net effect on FA is minimal (77.7% vs 77.5%), suggesting that the forgetting reduction and plasticity loss roughly cancel out. While KD is effective at preserving stability, it may be too restrictive for scenarios where rapid adaptation to new subjects is critical.

2) Task-Aware Gates: In our task-free setting, a single set of shared gates is trained across all subjects, and the model must infer subject identity implicitly from input statistics. An alternative design is to maintain separate gates for each subject and provide explicit task identity at test time. This

TABLE VIII
EFFECT OF THE NUMBER OF CHANNEL GATES INSERTED INTO THE FROZEN BACKBONE (MEAN ± STD OVER 10 PERMUTATIONS). GATES ARE ADDED TO THE FIRST N STACKS, FROM 0 (FROZEN CLASSIFIER ONLY) TO 4 (ALL STACKS, OUR DEFAULT). 0 AND 4 GATES CORRESPOND TO THE *Pretrained* AND *Pretrained + Gates (Ours)* ROWS OF TABLE V.

	PAMAP2 8 subj., 12 act.			**DSA** 8 subj., 18 act.			**UCI-HAR** 30 subj., 6 act.			**RealWorld** 15 subj., 8 act.		
	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑	FA ↑	FM ↓	LA ↑
0 gates (classifier only)	76.5±4.0	17.5±4.0	93.9±0.4	75.7±2.7	22.1±2.7	97.9±0.2	80.1±9.4	13.9±9.5	94.0±1.5	64.4±3.6	27.3±3.6	86.2±0.9
1 gate	75.3±2.4	15.2±2.7	90.5±0.8	70.5±4.0	25.4±3.6	95.9±0.8	79.5±3.6	11.7±4.2	91.2±2.8	62.8±3.8	28.9±3.7	85.4±1.0
2 gates	72.6±3.2	18.2±3.0	90.8±0.7	71.5±2.7	24.8±2.2	96.3±0.8	78.7±6.6	14.0±5.7	92.8±2.3	61.9±4.5	29.6±4.4	86.0±1.0
3 gates	77.0±2.5	13.8±2.1	90.8±0.6	71.9±6.0	24.0±5.8	95.9±0.3	79.8±6.0	12.2±5.8	92.0±3.0	64.1±4.1	27.1±4.0	86.1±0.9
4 gates (Ours)	77.7±2.5	16.2±2.6	93.8±0.4	78.7±1.7	19.6±1.7	98.3±0.3	80.3±5.3	14.5±5.4	94.9±1.2	65.6±3.2	26.4±3.3	86.5±0.8

task-aware approach represents an oracle setting where perfect task segmentation is available.

As expected, task-aware gates achieve lower forgetting than task-free gates. On PAMAP2, FM drops from 16.2% (task-free) to 12.3% (task-aware), and FA improves from 77.7% to 78.9%. The improvement is consistent but modest (2-4 percentage points), suggesting that task-free gates already capture much of the benefit of subject-specific adaptation without requiring explicit task boundaries. This is encouraging for real-world deployments, where task identity may not be available or may be ambiguous.

Task-aware gates require storing one set of gate parameters per subject, increasing memory linearly with the number of tasks. For UCI-HAR with 30 subjects, this means 30× the gate parameters. In contrast, task-free gates require only a single set regardless of the number of subjects. For resource-constrained devices such as smartphones or wearables, task-free gates are thus the more practical choice.

3) Combining Gates with Experience Replay: To demonstrate that our gating mechanism provides benefits orthogonal to replay-based methods, we evaluate a variant that combines gates with Dark Experience Replay (DER). This hybrid approach maintains a small memory buffer (500 samples) and applies the DER++ loss while training only the gates and classifier, keeping the backbone frozen.

Results show that Gates + DER achieves substantial improvements over gates alone. On PAMAP2, FA improves from 77.7% to 84.3%, while FM drops dramatically from 16.2% to 6.1%, a 62% relative reduction in forgetting. Similar patterns emerge on DSA (FA: 78.7% → 85.7%, FM: 19.6% → 9.6%) and UCI-HAR (FA: 80.3% → 85.0%, FM: 14.5% → 6.1%). Importantly, this performance is competitive with standard DER++ while updating only 2.5% of model parameters (gates + classifier vs. full backbone). This demonstrates that architectural gating and replay provide complementary mechanisms for preventing forgetting, and can be effectively combined when memory constraints permit storing past samples.

However, maintaining a replay buffer may be impractical for on-device deployment scenarios. Storing raw sensor data on smartphones or wearables raises storage concerns. More critically, retaining historical movement patterns poses privacy risks, as such data can reveal sensitive information about users' daily routines and health conditions. For these reasons, our replay-free gating approach remains the preferred choice for privacy-sensitive edge deployments, achieving strong performance without requiring any data retention.

VII. CONCLUSION AND FUTURE WORK

We presented a parameter-efficient continual learning framework for human activity recognition that combines frozen pretrained backbones with lightweight channel-wise gating mechanisms. Our key insight is that adaptation should operate through feature *selection* rather than feature *generation*: by restricting learned transformations to diagonal scaling of existing features, we preserve the geometric structure of pretrained representations while enabling subject-specific modulation. This bounded form of adaptation limits representational drift and reduces catastrophic forgetting without requiring replay buffers or task-specific regularization. These properties make our approach well-suited for on-device deployment in IoT applications such as wearable health monitoring, where privacy constraints preclude data transmission and memory limitations prohibit replay buffers.

Figure 5 illustrates the effectiveness of our approach on the same PAMAP2 subject sequence shown in Figure 1, demonstrating substantially reduced forgetting. Extensive experiments across foure HAR benchmarks further validated our approach. On PAMAP2, freezing the backbone reduced forgetting from 39.7% to 17.5%, while adding gates further improved final accuracy to 77.7% with only 16.2% forgetting, outperforming regularization-based methods (EWC, LwF) and architecture-based methods (HAT) while training less than 2% of model parameters. We also demonstrated that gating is complementary to replay: combining gates with DER achieved 84.3% final accuracy and 6.1% forgetting, competitive with full-model replay methods at a fraction of the parameter cost.

Our approach has limitations that suggest directions for future work. The reliance on a pretrained backbone assumes access to a sufficiently diverse source dataset; performance may degrade when source and target domains differ substantially. Additionally, while task-free gates perform well, explicitly modeling task structure could further reduce forgetting in scenarios where task boundaries are available. Future work could explore adaptive gating architectures that automatically determine the optimal number and placement of gates, as well as extensions to other domain-incremental settings beyond HAR.

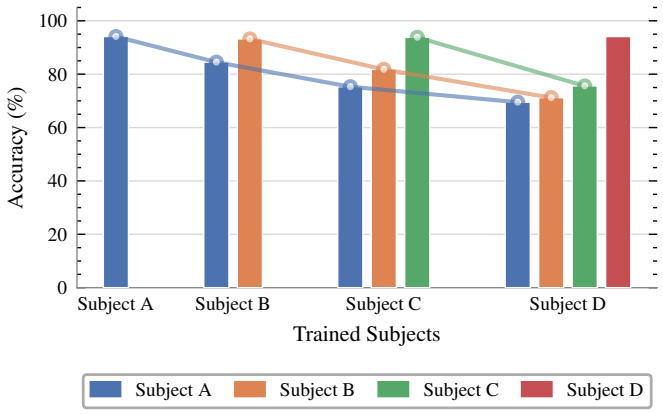

Fig. 5. Per-subject accuracy evolution with our gating approach on PAMAP2. Unlike the baseline in Figure 1, which dropped to 40% on Subject 1 after four subjects, our method retains 69.5% accuracy while achieving 93.8% learning accuracy on new subjects.

REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608019300231

[2] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[3] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, pp. 128–135, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364661399012942

[4] J. Qi, P. Yang, G. Min, O. Amft, F. Dong, and L. Xu, "Advanced internet of things for personalised healthcare systems: A survey," *Pervasive and Mobile Computing*, vol. 41, pp. 132–149, 2017.

[5] S. Hashemi, S. Ebrahimibasabi, M. Sajjadi, N. Shahraki, D. Tamjid Shabestari, M. Golshahi, S. Zeinolabedinzadeh, H. Arami, and L. Khalifehzadeh, "Ultra-sensitive wireless capacitive nanocomposite-based pressure sensors for health monitoring," *Advanced Materials Technologies*, vol. 10, no. 19, p. e01316, 2025.

[6] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Systems with Applications*, vol. 137, pp. 167–190, 2019.

[7] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 351–360.

[8] Z. Mohammadibalini, A. Patel, C. Phillips, and M. Adams, "Disparities in microscale pedestrian environment features by income and demographic context across four us cities," in *ANNALS OF BEHAVIORAL MEDICINE*, vol. 60. OXFORD UNIV PRESS INC JOURNALS DEPT, 2001 EVANS RD, CARY, NC 27513 USA, 2026, pp. S656–S656.

[9] N. D. Lane, P. Georgiev, and L. Qendro, "Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015, pp. 283–294.

[10] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–40, 2021.

[11] R. K. Sah, S. I. Mirzadeh, and H. Ghasemzadeh, "Continual learning for activity recognition," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2022, pp. 2416–2420.

[12] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[13] A. Reiss, "PAMAP2 Physical Activity Monitoring," UCI Machine Learning Repository, 2012, DOI: https://doi.org/10.24432/C5NW2H.

[14] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[15] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[16] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.

[17] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," 2020. [Online]. Available: https://arxiv.org/abs/2004.07211

[18] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[19] J. Serrà, D. Surís, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," 2018. [Online]. Available: https://arxiv.org/abs/1801.01423

[20] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 139–149.

[21] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 631–648.

[22] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019, pp. 2790–2799.

[23] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[24] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[25] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[26] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7765–7773.

[27] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 5362–5383, 2024.

[28] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Class-incremental learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5513–5533, 2024.

[29] G. M. van de Ven, N. Soures, and D. Kudithipudi, "Continual learning and catastrophic forgetting," *arXiv preprint arXiv:2403.05175*, 2024.

[30] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022.

[31] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira, "CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 11 909–11 919.

[32] D.-W. Zhou, H.-L. Sun, J. Ning, H.-J. Ye, and D.-C. Zhan, "Continual learning with pre-trained models: A survey," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024, pp. 6593–6601.

[33] L. Thede, M. Mundt, B. Sick, and V. Ramesh, "Reflecting on the state of rehearsal-free continual learning with pretrained models," *arXiv preprint arXiv:2406.09384*, 2024.

[34] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 3–19.

[35] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[36] N. Y. Masse, G. D. Grant, and D. J. Freedman, "Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization,"

Proceedings of the National Academy of Sciences, vol. 115, no. 44, pp. E10 467–E10 475, 2018.

[37] S. Mekruksavanich, A. Jitpattanakul, K. Sitthithakerngkiet, P. Youplao, and P. Yupapin, "ResNet-SE: Channel attention-based deep residual network for complex activity recognition using wrist-worn wearable sensors," *IEEE Access*, vol. 10, pp. 51 142–51 154, 2022.

[38] K. Chen, L. Yao, D. Zhang, B. Wang, Z. Lu, M. Hong, and C. Lofi, "Continual learning for activity recognition," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2022, pp. 1–10.

[39] S. Jha, M. Schiemer, and J. Ye, "Continual learning in sensor-based human activity recognition: An empirical benchmark analysis," *Information Sciences*, vol. 575, pp. 1–21, 2021.

[40] H. Partovi Aria, H. Kim, S. Meshkat Alsadat, and Z. Xu, "Learning causal dynamics and reward machines: A framework for faster reinforcement learning with extended temporal tasks," in *2025 5th International Conference on Computer, Control and Robotics (ICCCR)*, 2025, pp. 519–525.

[41] M. Schiemer, L. Fang, S. Dobson, and J. Ye, "Online continual learning for human activity recognition," *Pervasive and Mobile Computing*, vol. 93, p. 101817, 2023.

[42] R. Adaimi and E. Thomaz, "Lifelong adaptive machine learning for sensor-based human activity recognition using prototypical networks," *Sensors*, vol. 22, no. 18, p. 6881, 2022.

[43] E. Farahmand, R. R. Azghan, N. T. Chatrudi, E. Kim, G. K. Gudur, E. Thomaz, G. Pedrielli, P. Turaga, and H. Ghasemzadeh, "Attengluco: Multimodal transformer-based blood glucose forecasting on ai-readi dataset," 2025. [Online]. Available: https://arxiv.org/abs/2502.09919

[44] M. Böck, M. Moser, S. Yan, S. Suh, and T. Plötz, "Balancing continual learning and fine-tuning for human activity recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 268–276.

[45] R. R. Azghan, G. K. Gudur, M. Malu, E. Thomaz, G. Pedrielli, P. Turaga, and H. Ghasemzadeh, "Clad-net: Continual activity recognition in multi-sensor wearable systems," 2025. [Online]. Available: https://arxiv.org/abs/2509.23077

[46] P. Sundaramoorthy, G. K. Gudur, M. R. Moorthy, R. N. Bhandari, and V. Vijayaraghavan, "Harnet: Towards on-device incremental learning using deep ensembles on constrained devices," in *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning (EMDL '18)*, Munich, Germany, 2018, pp. 1–6.

[47] B. Barshan and K. Altun, "Daily and Sports Activities," UCI Machine Learning Repository, 2010, DOI: https://doi.org/10.24432/C5C59F.

[48] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, "Human activity recognition using smartphones," UCI Machine Learning Repository, 2013, dOI: https://doi.org/10.24432/C54S4K.

[49] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2016, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/7456521

[50] G. Weiss, "WISDM Smartphone and Smartwatch Activity and Biometrics Dataset ," UCI Machine Learning Repository, 2019, DOI: https://doi.org/10.24432/C5HK59.

[51] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531