PROTOROUTE: DUAL-TIMESCALE PERSISTENT EVIDENCE LEARNING FOR MEDICAL TIME-SERIES CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing medical time-series classification (MedTSC) methods primarily encode class knowledge in network parameters or prototypes optimized on the same gradient timescale as the encoder. We argue that transient *within-sample* structure and persistent *cross-sample* class evidence have different statistical scopes and should therefore be modeled and updated separately. Based on this perspective, we propose **ProtoRoute**, a dual-timescale framework that couples a *fast* parametric encoder with a *slow*-evolving non-parametric class-evidence dictionary. The encoder-agnostic fast pathway captures transient representations of the current sample, while the persistent pathway maintains reusable class evidence across training observations. To read from this persistent state, we introduce hierarchical Token–Class–Prototype Slot routing, where Top-r prototype aggregation preserves heterogeneous within-class evidence modes and Top-q token aggregation suppresses dilution from weak background evidence. To evolve the persistent state, we further introduce Loss-Adaptive Prototype Evolution, an explicit epoch-level state transition that selectively incorporates reliable and complementary evidence while adapting class-wise replacement to training dynamics. Experiments on 8 datasets demonstrate the strong superiority of ProtoRoute.

1 INTRODUCTION

Medical time-series (MedTS), such as EEG and ECG, are critical in modern healthcare (Huang et al., 2026; Ye et al., 2024). In realistic deployment, models must generalize to subjects unseen during training. Subject-specific physiological baselines, waveform morphology, and noise characteristics can provide strong shortcuts under subject-dependent evaluation, yet such shortcuts do not reliably transfer across individuals (Wang et al., 2024; 2023). This challenge is closely related to two statistical properties of diagnostic evidence. (*i*) class-relevant information is rarely distributed uniformly over an entire recording; it is often concentrated in a small number of temporal segments, frequency regions or physiological channels (Li et al., 2026; Yuan et al., 2025). (*ii*) the same class may be supported by heterogeneous local manifestations, including distinct morphologies, spectral signatures, and channel-localized patterns (Zhang et al., 2026; Du et al., 2026). MedTSC must therefore simultaneously address ***within-sample background dilution***, which requires identifying discriminative tokens amid predominantly non-informative or subject-specific variation, and ***within-class evidence heterogeneity***, which requires representing more than a single class center.

Most existing methods tackle these difficulties by improving representations using only the current sample itself, through temporal, spectral, channel, or sparse-token modeling (Wang et al., 2024; Fan et al., 2025; Chen et al., 2024; Li et al., 2026). Prototype-based approaches provide an explicit form of class representation (Chen et al., 2019; Ming et al., 2019; Sethi et al., 2025), while clustering- or momentum-based methods decouple prototype estimation from standard network optimization (Li et al., 2020; 2021). However, a complementary question remains underexplored: *how can reusable class-conditioned evidence be explicitly represented across samples and integrated with the representation of the current recording as the encoder evolves during training?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We build on the observation that **transient within-sample structure and persistent cross-sample class evidence have different statistical scopes, and should therefore be represented as distinct model states and updated at different timescales**. We explicitly separate

$$\underbrace{\textit{Transient within-sample structure}}_{\textbf{\textit{Fast} parametric state}} \longleftrightarrow \underbrace{\textit{Persistent cross-sample class evidence}}_{\textbf{\textit{Slow}-evolving non-parametric state}}$$

where ***fast*** and ***slow****-evolving* refer to state-update timescales rather than computational speed. The fast parametric state flexibly encodes the current sample and is continuously optimized during training, while the persistent state accumulates reusable class evidence across observations and evolves on a slower timescale. This decomposition provides a direct modeling route toward subject-independent generalization: sample-specific structure can adapt rapidly, while transferable class evidence is maintained more conservatively.

Based on this principle, we propose **ProtoRoute**, a dual-timescale framework that separates transient ***within-sample*** representation learning from the accumulation of persistent ***cross-sample*** class evidence, as illustrated in Figure 1. It couples an encoder-agnostic fast pathway with a label-indexed, multi-slot dictionary of persistent token-level class evidence. The persistent pathway contains two complementary operations. *Persistent Evidence Routing (PER)* **reads** the dictionary through hierarchical Token–Class–Prototype Slot routing. PER performs sparse evidence aggregation along two orthogonal axes: Top-r aggregation over prototype slots preserves multiple locally compatible evidence modes within each class, while Top-q aggregation over tokens suppresses dilution from weak background evidence. Persistent evidence therefore contributes at both the representation and decision levels. *Loss-Adaptive Prototype Evolution (LAPE)* **evolves** the dictionary only at epoch boundaries, avoiding sensitivity to rapidly changing mini-batch representations. It combines classifier-aligned candidate mining, reliability-aware admission, class-dependent plasticity, diversity-aware selection, and usage–redundancy–conflict-based slot replacement. The dictionary is therefore kept outside mini-batch gradient optimization and evolves on a slower timescale than the parametric encoder.

Together, PER and LAPE define the **read** and **evolve** operations of persistent evidence learning. This design explicitly represents transferable class evidence as a persistent model state, while the parametric pathway continues to capture transient sample-specific structure at the mini-batch timescale.

Our contributions are summarized as follows:

- **Dual-timescale persistent evidence learning.** We formulate MedTSC as the interaction between a fast parametric state for transient *within-sample* structure and a slow-evolving non-parametric state for persistent *cross-sample* class evidence.
- **Persistent Evidence Routing for sparse and heterogeneous evidence.** We introduce hierarchical ***Token–Class–Prototype Slot*** routing with Top-r prototype aggregation and Top-q token aggregation, preserving multiple within-class evidence modes while suppressing background dilution.
- **Loss-Adaptive Prototype Evolution for persistent-state updates.** We propose an epoch-level state-transition mechanism that conservatively evolves the class-evidence dictionary outside mini-batch backpropagation.
- **Comprehensive empirical validation.** Experiments on eight public benchmarks, together with controlled ablations, robustness analyses, persistent-state dynamics, interpretability studies, and backbone-portability experiments, validate the effectiveness of persistent evidence learning.

2 Related Work

Medical Time-Series Representation Learning. Recent MedTSC methods typically produce a global representation of a MedTS window using only the current recording itself. Medformer (Wang et al., 2024) models multi-scale physiological structure, MedGNN (Fan et al., 2025) captures multi-resolution spatiotemporal dependencies, and TimeMIL (Chen et al., 2024) exploits localized discriminative patterns through multiple-instance learning. More recent methods such as TS-Fingerprint and MedSpaformer explicitly reduce token redundancy or emphasize sparse informative

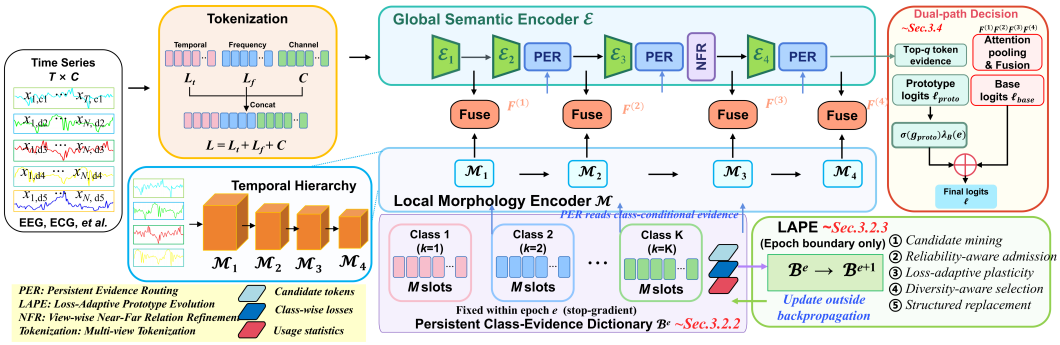

Figure 1: **Overview of ProtoRoute-TSC,** which couples a fast parametric encoder with a slow-evolving persistent class-evidence dictionary for subject-independent MedTSC.

patterns (Li et al., 2026; Ye et al., 2026). These methods provide strong sample-specific representations, while transferable class knowledge remains largely implicit in network parameters or input-dependent features, with limited explicit modeling of reusable class-level evidence across samples.

Prototype Learning. Prototype learning provides another route to explicit class representation. ProSeNet and ProtoECGNet use prototypes for case-based or clinically structured reasoning (Ming et al., 2019; Sethi et al., 2025), while PCL and MoPro estimate or update prototypes through clustering or momentum mechanisms (Li et al., 2020; 2021). In these formulations, prototypes primarily serve as semantic anchors for matching, prediction, or representation alignment. How prototype-based representations capture heterogeneous intra-class structure and remain well aligned as the feature space evolves during training is less well understood.

3 Methodology

3.1 Dual-Timescale Formulation

Let $\mathcal{D}_{\mathrm{tr}} = \{(\boldsymbol{X}_n, y_n)\}_{n=1}^{N}, \boldsymbol{X}_n \in \mathbb{R}^{C\times T}, y_n \in \{1,\dots,K\}$ denote a MedTS training set, where C and T are the numbers of channels and time samples, respectively. We learn $f_{\theta,\mathcal{B}} : \mathbb{R}^{C\times T} \to \mathbb{R}^K$, where θ parameterizes the neural encoder and $\mathcal{B}$ is a persistent class-evidence dictionary.

Definition 3.1 (Dual-Timescale Model State)**.** *At epoch e and mini-batch b, the model state is*

$$\Theta_{e,b} = (\theta_{e,b}, \mathcal{B}_e, \mathcal{U}_{e,b}), \tag{1}$$

where $\theta_{e,b}$ is the fast parametric state, $\mathcal{B}_e$ is the persistent non-parametric prototype state, and $\mathcal{U}_{e,b}$ contains online, non-gradient statistics used for dictionary maintenance.

Within epoch e, the prototype vectors remain fixed while the parametric state is optimized over mini-batches:

$$\theta_{e,b+1} = \theta_{e,b} - \eta\nabla_\theta \mathcal{L}\left(\theta_{e,b}; \mathrm{sg}(\mathcal{B}_e)\right), \qquad \mathcal{B}_{e,b} \equiv \mathcal{B}_e, \tag{2}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The auxiliary statistics $\mathcal{U}_{e,b}$ may be accumulated online but are not optimized by backpropagation.

At the epoch boundary, the persistent dictionary is updated through an explicit non-parametric evolution operator $\mathscr{E}$:

$$\mathcal{B}_{e+1} = \mathscr{E}\left(\mathcal{B}_e, \mathcal{C}_e, \boldsymbol{\mathcal{L}}^{(e)}, \boldsymbol{\mathcal{L}}^{(e-1)}, \mathcal{U}_e\right), \tag{3}$$

where $\mathcal{C}_e$ contains the selected token candidates during e. $\mathscr{E}$ is factorized as: $\mathscr{E} = \mathcal{R}_{\mathrm{replace}} \circ \mathcal{L}_{\mathrm{select}} \circ \mathcal{A}_{\mathrm{admit}} \circ \mathcal{M}_{\mathrm{mine}}$, corresponding respectively to classifier-aligned candidate mining, reliability-aware admission, diversity-aware selection, and usage–redundancy–conflict replacement.

This decomposition assigns complementary statistical roles to the two states: θ models transient within-sample structure through mini-batch gradient updates, whereas $\mathcal{B}$ accumulates persistent cross-sample class evidence and evolves only at epoch boundaries. During validation and testing, θ, $\mathcal{B}$, and the usage statistics are frozen, preventing evaluation-time state mutation.

3.2 PERSISTENT EVIDENCE LEARNING

The persistent pathway represents class evidence explicitly rather than encoding it exclusively in the parameters of the fast encoder. Given token representations produced by the current parametric state, it performs two complementary operations: *Persistent Evidence Routing (PER)* reads class-conditioned evidence from a slowly evolving dictionary, whereas *Loss-Adaptive Prototype Evolution (LAPE)* updates this dictionary only at epoch boundaries. Thus, the persistent pathway can be summarized as:

$$\underbrace{\mathcal{B}_e \xrightarrow{\text{PER}} \text{sample representation / prediction}}_{\textbf{\textit{Read} persistent evidence}} \qquad \underbrace{\mathcal{B}_e \xrightarrow{\text{LAPE}} \mathcal{B}_{e+1}}_{\textbf{\textit{Evolve} persistent evidence}} .$$

The persistent mechanism operates on token representations and does not assume a particular fast encoder; Sec. 3.3 describes the parametric encoder used in our main implementation.

3.2.1 PERSISTENT CLASS-EVIDENCE DICTIONARY

We maintain a fixed-capacity class-evidence dictionary $\mathcal{B} = \{\boldsymbol{b}_{k,m}\} \in \mathbb{R}^{K\times M\times D}$, $\|\boldsymbol{b}_{k,m}\|_2 = 1$, with M unordered evidence slots per class. Unlike ordinary trainable prototypes, $\mathcal{B}$ is a persistent *non-gradient* state: its vectors remain fixed during mini-batch optimization ($\nabla_{\mathcal{B}}\mathcal{L} = 0$) and are modified only by the epoch-level evolution operator in Sec. 3.2.3.

3.2.2 HIERARCHICAL TOKEN–CLASS–PROTOTYPE ROUTING (PER)

Given a stage token $\boldsymbol{z}_i$, we form the normalized query $\boldsymbol{q}_i = \mathrm{Norm}(\mathrm{LN}(\boldsymbol{z}_i))$ and compute $s_{i,k,m} = \boldsymbol{q}_i^\top \boldsymbol{b}_{k,m}$. Token-level class evidence is estimated from the r most compatible slots:

$$e_{i,k} = \frac{1}{r\tau} \sum_{m\in \mathrm{TopR}_r(\{s_{i,k,m}\}_{m=1}^{M})} s_{i,k,m}, \qquad \alpha_{i,k} = \mathrm{softmax}_k(e_{i,k}).$$

Top-r aggregation retains multiple locally compatible evidence modes without averaging over unrelated slots. Once class responsibility is obtained, all within-class slots remain available for soft retrieval:

$$\beta_{i,k,m} = \mathrm{softmax}_m(s_{i,k,m}/\tau_s), \qquad \boldsymbol{r}_{i,k} = \sum_{m=1}^{M} \beta_{i,k,m}\boldsymbol{b}_{k,m}, \qquad \boldsymbol{r}_i = \sum_{k=1}^{K} \alpha_{i,k}\boldsymbol{r}_{i,k}. \tag{4}$$

This yields the hierarchical routing process *Token → Class → Prototype Slot*.

The retrieved state is injected through a controlled residual $\boldsymbol{z}_i^{\text{route}} = \boldsymbol{z}_i + \lambda_{\mathcal{B}}(e) g_i^{\text{route}} \phi_{\text{route}}(\boldsymbol{r}_i)$, where $g_i^{\text{route}} = \sigma(\phi_{\text{gate}}(\boldsymbol{z}_i))$, $\lambda_{\mathcal{B}}(e)$ is the epoch-dependent coupling strength.

The final active router additionally forms prototype logits from the q strongest token evidences:

$$\ell_k^{\text{proto}} = \frac{1}{q} \sum_{i\in \mathrm{TopQ}_q(\{e_{i,k}\}_{i=1}^{L})} e_{i,k}. \tag{5}$$

The router thus performs sparse aggregation along two orthogonal dimensions: Top-r preserves heterogeneous within-class evidence, whereas Top-q prevents the dominant local evidence from being diluted by weakly discriminative background tokens.

3.2.3 LOSS-ADAPTIVE PROTOTYPE EVOLUTION (LAPE)

While PER reads from the persistent state, LAPE evolves it. Because updating prototypes with the same mini-batch gradients as the encoder would make them sensitive to a rapidly changing representation space and batch-specific noise, $\mathcal{B}$ is updated only once per epoch through an explicit fixed-budget transition.

Classifier-Aligned Candidate Mining. Let $j^\star = \max \mathcal{S}_{\text{mem}}$ denote the final active routing stage and $\boldsymbol{U}_n^\star = \{\boldsymbol{u}_{n,i}^\star\}_{i=1}^{L}$ its pre-retrieval tokens. Candidates are ranked by $\psi_{n,i} = a_{n,i}^{(J)} \alpha_{n,i,y_n}$, so

dictionary querying, prototype supervision, and dictionary writing share the same representation geometry.

Reliability-aware Admission. For a candidate $\boldsymbol{z}$ of class k, we measure prediction confidence, within-class novelty, and maximum cross-class compatibility:

$$c_k(\boldsymbol{z}) = p_\theta\left(y = k \mid \boldsymbol{X}(\boldsymbol{z})\right), \quad n_k(\boldsymbol{z}) = \frac{1 - \max_m \boldsymbol{z}^\top \boldsymbol{b}_{k,m}}{2}, \quad \xi_k(\boldsymbol{z}) = \max_{\substack{k' \neq k \\ m}} \boldsymbol{z}^\top \boldsymbol{b}_{k',m} \tag{6}$$

The admissible candidate set is $\mathcal{A}_{e,k} = \{\boldsymbol{z} \in \mathcal{C}_{e,k} : c_k(\boldsymbol{z}) \geq \theta_{\text{conf}},\ n_k(\boldsymbol{z}) \geq \theta_{\text{nov}},\ \xi_k(\boldsymbol{z}) \leq \theta_{\text{inter}}\}$.

These constraints reject candidates from uncertain predictions, evidence already well covered by the current class dictionary, and evidence that is excessively compatible with another class. Each admitted candidate is then assigned the quality score $Q_k(\boldsymbol{z})$:

$$Q_k(\boldsymbol{z}) = w_c c_k(\boldsymbol{z}) + w_n n_k(\boldsymbol{z}) + w_d \frac{1 - \xi_k(\boldsymbol{z})}{2}, \qquad w_c + w_n + w_d = 1 \tag{7}$$

Loss-Adaptive Plasticity. We adapt the class-wise replacement budget according to the recent loss trend. Let $\Delta_k^{(e)} = \mathcal{L}_k^{(e-1)} - \mathcal{L}_k^{(e)}$, where a positive value indicates an improving class and a negative value indicates deterioration. We define the replacement fraction and the corresponding slot budget:

$$\gamma_k^{(e)} = \text{clip}(-\alpha \Delta_k^{(e)} + \beta, \gamma_{\min}, \gamma_{\max}), \qquad R_k^{(e)} = \max(1, \text{round}(\gamma_k^{(e)} M)). \tag{8}$$

Thus, improving classes receive smaller replacement budgets and preserve more established evidence, whereas deteriorating classes are allowed greater plasticity through larger replacement budgets.

Diversity-aware Selection and Structured Replacement. From the admissible set $\mathcal{A}_{e,k}$, LAPE greedily selects at most $R_k^{(e)}$ complementary candidates. Starting from $\mathcal{S}_{e,k}^{(0)} = \emptyset$, the t-th candidate is selected as:

$$\boldsymbol{z}_t^\star = \arg \max_{\boldsymbol{z} \in \mathcal{A}_{e,k} \setminus \mathcal{S}_{e,k}^{(t-1)}} \left[Q_k(\boldsymbol{z}) + \lambda_{\text{div}} D\Big(\boldsymbol{z}, \mathcal{S}_{e,k}^{(t-1)}\Big) \right]. \tag{9}$$

Here, all candidate features are ℓ_2-normalized, and $D(\boldsymbol{z}, \mathcal{S}) = (1 - \max_{\tilde{\boldsymbol{z}} \in \mathcal{S}} \boldsymbol{z}^\top \tilde{\boldsymbol{z}})/2$ for nonempty $\mathcal{S}$, with $D(\boldsymbol{z}, \emptyset) = 0$. Thus, D measures the normalized cosine distance to the nearest previously selected candidate. The first candidate is therefore selected solely according to $Q_k(\boldsymbol{z})$, whereas subsequent selections favor candidates that are both reliable and complementary to the evidence already selected. After greedy selection, let $J_k \leq R_k^{(e)}$ denote the number of selected candidates and $\boldsymbol{Z}_{e,k}^\star = [\boldsymbol{z}_1^\star; \ldots; \boldsymbol{z}_{J_k}^\star]$ their stacked representations.

To determine which persistent evidence should be removed, existing slots are ranked according to low usage, within-class redundancy, and cross-class conflict:

$$P_{\text{use}}(k,m) = 1 - \frac{U_{k,m}}{\max_j U_{k,j} + \epsilon}, \quad P_{\text{red}}(k,m) = \frac{1 + \max_{j \neq m} \boldsymbol{b}_{k,m}^\top \boldsymbol{b}_{k,j}}{2}, \tag{10}$$

$$P_{\text{conf}}(k,m) = \frac{1 + \max_{\substack{k' \neq k \\ j}} \boldsymbol{b}_{k,m}^\top \boldsymbol{b}_{k',j}}{2}, \tag{11}$$

and the overall replacement priority is:

$$P_{k,m} = w_u P_{\text{use}}(k,m) + w_r P_{\text{red}}(k,m) + w_x P_{\text{conf}}(k,m). \tag{12}$$

We select the J_k slots with the highest replacement priorities $\mathcal{R}_{e,k} = \text{Top}_{J_k}\left(\{P_{k,m}\}_{m=1}^M\right)$, and collect the selected candidates as $\boldsymbol{Z}_{e,k}^\star = [\boldsymbol{z}_1^\star; \ldots; \boldsymbol{z}_{J_k}^\star]$.

The corresponding dictionary slots are then updated jointly as follows: $\mathcal{B}_k^{e+1}[\mathcal{R}_{e,k}] \leftarrow \text{RowNorm}(\boldsymbol{Z}_{e,k}^\star)$. All other slots are preserved and the usage statistics of replaced slots are reset. LAPE therefore performs a selective epoch-level state transition of the persistent dictionary outside mini-batch backpropagation.

3.3 FAST PARAMETRIC ENCODER: ONE INSTANTIATION

The fast state is encoder-agnostic and models transient within-sample representations. In our main implementation, we instantiate it with a lightweight multi-view semantic–morphology encoder that captures temporal, spectral, channel, and local waveform structure.

Multi-View Tokenization and Semantic–Morphology Encoding. Each channel is standardized along time, $\widetilde{X}_{c,t} = (X_{c,t} - \mu_c)/(\sigma_c + \epsilon)$. Temporal, frequency, and channel tokens are then constructed as:

$$\begin{aligned} \boldsymbol{Z}^t &= \mathcal{T}_\theta(\widetilde{\boldsymbol{X}}) \in \mathbb{R}^{L_t \times D}, \qquad \boldsymbol{Z}^f = \mathcal{F}_\theta\Big(\log(1 + |\,\mathrm{RFFT}(\widetilde{\boldsymbol{X}})|)\Big) \in \mathbb{R}^{L_f \times D}, \\ \boldsymbol{Z}^c &= \mathcal{C}_\theta(\widetilde{\boldsymbol{X}}) \in \mathbb{R}^{C \times D}, \qquad \boldsymbol{H}^{(0)} = \mathrm{PE}\big(\mathrm{LN}[\boldsymbol{Z}^t + \boldsymbol{e}_t; \boldsymbol{Z}^f + \boldsymbol{e}_f; \boldsymbol{Z}^c + \boldsymbol{e}_c]\big)\,. \end{aligned} \tag{13}$$

Here, $\mathcal{T}_\theta$ and $\mathcal{F}_\theta$ are convolutional patch embeddings, while $\mathcal{C}_\theta$ applies channel-wise adaptive pooling followed by a shared projection. The three token views retain their index ranges after concatenation, with total length $L = L_t + L_f + C$.

The semantic encoder contains J Transformer stages. At stage j,

$$\boldsymbol{U}^{(j)} = \mathcal{E}_j(\boldsymbol{H}^{(j-1)}), \qquad \boldsymbol{V}^{(j)} = \begin{cases} \mathcal{R}_{\mathcal{B}}(\boldsymbol{U}^{(j)}), & j \in \mathcal{S}_{\mathrm{mem}}, \\ \boldsymbol{U}^{(j)}, & \text{otherwise}, \end{cases} \tag{14}$$

where $\mathcal{R}_{\mathcal{B}}$ denotes PER (Sec. 3.2.2). A view-wise Near–Far Relation Refiner (NFR), described below, is optionally applied at stages $\mathcal{S}_{\mathrm{rel}}$:

$$\boldsymbol{S}^{(j)} = \begin{cases} \mathcal{G}_j(\boldsymbol{V}^{(j)}), & j \in \mathcal{S}_{\mathrm{rel}}, \\ \boldsymbol{V}^{(j)}, & \text{otherwise}. \end{cases} \tag{15}$$

When both are active, stage j follows $\mathcal{E}_j \to \mathcal{R}_{\mathcal{B}} \to \mathcal{G}_j$, yielding a ***coarse-to-fine*** reasoning path. PER first retrieves class-conditional prototype directions from the slow dictionary, establishing a coarse semantic reference for each token. NFR then performs fine-grained structural adjustment *within* this semantic context, enabling near-far separation to operate on semantically calibrated representations.

In parallel, a lightweight convolutional hierarchy extracts multi-scale local morphological features:

$$\boldsymbol{L}^{(1)} = \mathcal{M}_1(\widetilde{\boldsymbol{X}}), \qquad \boldsymbol{L}^{(j)} = \mathcal{M}_j(\boldsymbol{L}^{(j-1)}), \tag{16}$$

After projection and interpolation to the semantic-token length, $\overline{\boldsymbol{L}}^{(j)} = \boldsymbol{W}_j\,\mathrm{Interp}(\boldsymbol{L}^{(j)}, L)$, the two paths are fused:

$$\boldsymbol{F}^{(j)} = \mathrm{LN}\left(\boldsymbol{S}^{(j)} + \boldsymbol{G}^{(j)} \odot \overline{\boldsymbol{L}}^{(j)}\right), \qquad \boldsymbol{G}^{(j)} = \sigma\left(\phi_j[\boldsymbol{S}^{(j)}; \overline{\boldsymbol{L}}^{(j)}]\right). \tag{17}$$

Each stage is independently attention-pooled, $\boldsymbol{p}^{(j)} = \sum_i a_i^{(j)} \boldsymbol{F}_i^{(j)}$, and the resulting representations are fused as

$$\boldsymbol{h} = \phi_{\mathrm{fuse}}([\boldsymbol{p}^{(1)}; \ldots; \boldsymbol{p}^{(J)}]), \qquad \boldsymbol{\ell}^{\mathrm{base}} = \boldsymbol{W}_h \boldsymbol{h} + \boldsymbol{b}_h. \tag{18}$$

View-Wise Near–Far Relation Refinement. View-wise near–far relation refinement (NFR) provides a lightweight within-sample refinement after persistent routing. Because temporal continuity, spectral proximity, and channel topology induce different relational structures, the routed tokens are partitioned into:

$$\boldsymbol{Z} = [\boldsymbol{Z}^t; \boldsymbol{Z}^f; \boldsymbol{Z}^c], \qquad \boldsymbol{Z}^v \in \mathbb{R}^{P_v \times D}, \quad v \in \{t, f, c\}.$$

For each view, normalized token similarities define a content relation $S_{ij}^v = \lambda_{\mathrm{con}}(\widehat{\boldsymbol{z}}_i^v)^\top \widehat{\boldsymbol{z}}_j^v, \widehat{\boldsymbol{z}}_i^v = \mathrm{Norm}(\mathrm{LN}(\boldsymbol{z}_i^v))$.

For temporal and frequency views, the view-specific near and far neighborhoods are defined by token-index distance with radius d: $\mathcal{N}_v(i) = \{j \neq i : |i-j| \le d\}, \overline{\mathcal{N}}_v(i) = \{j : |i-j| > d\}, v \in \{t, f\}$. For the channel view, we instead use a predefined channel adjacency matrix: $\mathcal{N}_c(i) = \{j \neq i : A_{c,ij} > 0\}, \overline{\mathcal{N}}_c(i) = \{j \neq i : A_{c,ij} = 0\}$. NFR assigns the near and far relations independent normalization budgets:

$$\boldsymbol{A}_{\mathrm{near}}^v = \mathrm{MaskedSoftmax}(\boldsymbol{S}^v, \mathcal{N}_v), \qquad \boldsymbol{A}_{\mathrm{far}}^v = \mathrm{MaskedSoftmax}(\boldsymbol{S}^v, \overline{\mathcal{N}}_v), \tag{19}$$

$$\boldsymbol{O}^v = \boldsymbol{A}_{\mathrm{near}}^v \boldsymbol{V}^v + \boldsymbol{A}_{\mathrm{far}}^v \boldsymbol{V}^v, \qquad \boldsymbol{V}^v = \boldsymbol{W}_V \boldsymbol{Z}^v. \tag{20}$$

Separate normalization prevents local and non-local dependencies from competing for the same probability mass.

The response is written back through token- and model-level residual gates:

$$\overline{\boldsymbol{z}}_i^v = \boldsymbol{z}_i^v + g_i^v \phi_{\mathrm{ref}}(\boldsymbol{O}_i^v), \qquad g_i^v = \sigma\left(\boldsymbol{w}_g^\top \mathrm{LN}(\boldsymbol{z}_i^v) + b_g\right), \tag{21}$$

$$\boldsymbol{Z}^{\mathrm{out}} = \boldsymbol{Z} + \eta_{\mathrm{ref}}(\overline{\boldsymbol{Z}} - \boldsymbol{Z}), \qquad \eta_{\mathrm{ref}} = \sigma(a_{\mathrm{ref}}). \tag{22}$$

Thus, NFR serves as a complementary refinement inside the fast encoder, whereas persistent cross-sample evidence is supplied by PER.

3.4 Dual-Path Decision and Learning Objective

The parametric branch summarizes transient sample representations, whereas the persistent branch measures sparse class-consistent token evidence. Their logits are combined as

$$\boldsymbol{\ell}_n = \boldsymbol{\ell}_n^{\mathrm{base}} + \sigma(g_{\mathrm{proto}})\lambda_{\mathcal{B}}(e)\boldsymbol{\ell}_n^{\mathrm{proto}}, \tag{23}$$

where g_{proto} is learnable and $\lambda_{\mathcal{B}}(e)$ is the same coupling schedule used for feature-level PER. Thus, persistent evidence contributes consistently at both representation and decision levels.

Query-Aligned Prototype Supervision. Prototype-aware supervision is applied directly in the normalized query space used to access the persistent dictionary. Let $U_n^\star = \{u_{n,i}^\star\}_{i=1}^L$ denote the pre-retrieval tokens immediately before the last active router, and define the corresponding dictionary queries as $q_{n,i}^\star = \mathrm{Norm}(\mathrm{LN}(u_{n,i}^\star))$. Using the final-stage classifier attention $a_{n,i}^{(J)}$, we construct the query-aligned sample representation as $\bar{h}_n^B = \mathrm{Norm}\Big(\sum_{i=1}^L a_{n,i}^{(J)} q_{n,i}^\star\Big)$. Its compatibility with class k is then measured using the same Top-r prototype aggregation employed by the token router: $s_{n,k}^B = \frac{1}{r}\sum_{m\in \mathrm{TopR}_r(\{(\bar{h}_n^B)^\top b_{k,m}\}_{m=1}^M)} (\bar{h}_n^B)^\top b_{k,m}$.

The overall differentiable objective is $\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \chi_{\mathcal{B}}(\lambda_p \mathcal{L}_{\mathrm{proto}} + \lambda_m \mathcal{L}_{\mathrm{margin}})$, where $\chi_{\mathcal{B}} \in \{0,1\}$ indicates whether the persistent dictionary has been initialized. The classification loss supervises the final dual-path prediction; the prototype loss aligns the memory-query representation with its ground-truth class dictionary; and the margin loss explicitly separates the positive class from the hardest negative class:

$$\mathcal{L}_{\mathrm{cls}} = \frac{1}{N_b}\sum_{n=1}^{N_b} \mathrm{CE}\left(\boldsymbol{\ell}_n, y_n\right), \tag{24}$$

$$\mathcal{L}_{\mathrm{proto}} = \frac{1}{N_b}\sum_{n=1}^{N_b} \mathrm{CE}\left(\boldsymbol{s}_n^{\mathcal{B}}/\tau_p, y_n\right), \quad \mathcal{L}_{\mathrm{margin}} = \frac{1}{N_b}\sum_{n=1}^{N_b}\left[\mu + \max_{k\neq y_n} s_{n,k}^{\mathcal{B}} - s_{n,y_n}^{\mathcal{B}}\right]_+ \tag{25}$$

During mini-batch optimization, $\mathcal{B}$ is treated as a stop-gradient persistent state. The prototype-aware losses therefore reshape the parametric query space relative to the current dictionary, while prototype vectors are updated only by LAPE.

Alternating Dual-Timescale Optimization. Training starts with a parametric warm-up, $\lambda_{\mathcal{B}}(e) = 0$, after which the dictionary is initialized from selected token features and its coupling strength is gradually increased. Within each epoch, $\mathcal{B}_e$ remains fixed while θ is optimized over mini-batches; at the epoch boundary, the accumulated candidates and class-wise losses are used once to obtain $\mathcal{B}_{e+1}$ through LAPE. This alternating schedule lets the fast representation adapt continuously while the persistent class state evolves on a slower timescale.

4 Experiments

We evaluate ProtoRoute on eight public benchmarks spanning clinical diagnosis, neural decoding, and wearable sensing, including EEG, ECG, and inertial signals. We compare against 11 representative generic, MedTSC and evidence-aware time-series models. Results are reported over five random seeds (41–45) with six **macro**-averaged metrics: Accuracy, Precision, Recall, F1, AUROC, and AUPRC. Details on datasets, baselines and implementation are provided in Appendix B.

Table 1: **Classification results of our ProtoRoute and baseline models on the 8 different datasets under the subject-independent setting.** The best results are highlighted in **red**, and the second-best in **blue**. Note that the percentage symbol is omitted.

Dataset	Metric	Ours ProtoRoute	ICML'26 BioFormer	ICLR'26 TeCh	ICML'26 MedMamba	ICML'26 Fingerprint	WWW'25 MedGNN	NeurIPS'24 Medformer	ACM MM'25 KEMed	ICML'25 SoftShape	ICML'24 TimeMIL	ICLR'24 iTransformer	ICLR'23 PatchTST
APAVA (2)	Accuracy	**90.59**±1.16	82.31±1.55	**86.86**±1.09	84.48±0.55	80.29±0.32	82.60±0.35	78.74±0.64	78.74±1.48	81.22±0.62	79.50±0.69	74.55±1.66	67.38±1.83
	Precision	**91.93**±1.08	**87.87**±0.67	86.85±1.29	86.79±0.62	81.62±0.44	87.70±0.22	81.11±0.84	83.17±2.06	85.65±0.22	83.18±1.67	74.77±2.10	78.31±1.59
	Recall	**89.01**±1.52	78.50±1.94	**86.10**±1.00	81.87±0.27	77.61±0.49	78.93±0.09	75.40±0.66	72.15±2.22	77.55±0.87	75.86±0.73	71.76±1.72	60.41±2.30
	F1 Score	**89.96**±1.34	79.77±2.09	**86.30**±1.06	83.03±0.49	78.50±0.31	80.25±0.16	76.31±0.71	72.58±2.58	78.71±0.91	76.84±0.78	72.30±1.79	56.81±3.60
	AUROC	**96.94**±0.09	88.52±2.25	**94.02**±0.52	86.30±0.17	86.49±1.00	85.93±0.26	83.20±0.91	88.54±1.18	82.24±2.43	79.83±1.76	85.59±1.55	65.20±4.30
	AUPRC	**97.58**±0.07	88.16±2.40	**93.79**±0.56	87.07±0.99	86.74±0.92	87.33±0.82	83.66±0.92	88.68±1.09	83.93±2.19	78.95±1.62	84.39±1.57	67.61±3.76
TDBRAIN (2)	Accuracy	**95.67**±0.98	89.12±1.53	**93.21**±0.61	92.13±0.41	82.58±1.46	91.04±0.09	89.62±0.81	87.29±2.77	89.54±1.40	86.46±5.03	74.73±1.05	79.25±3.79
	Precision	**95.79**±0.92	89.56±1.19	**93.39**±0.58	92.33±0.37	82.71±1.55	91.15±0.12	89.68±0.78	87.46±2.65	89.77±1.42	87.59±4.48	74.78±1.04	79.60±4.09
	Recall	**95.67**±0.98	89.12±1.53	**93.21**±0.61	92.13±0.41	82.58±1.46	91.04±0.20	89.62±0.81	87.29±2.77	89.54±1.40	86.46±5.03	74.73±1.05	79.25±3.79
	F1 Score	**95.66**±0.98	89.09±1.56	**93.20**±0.61	92.12±0.42	82.57±1.44	91.04±0.20	89.62±0.81	87.28±2.78	89.53±1.38	86.32±5.14	74.72±1.05	79.20±3.77
	AUROC	**99.52**±0.19	96.58±0.48	**98.68**±0.19	95.62±0.66	88.55±1.03	96.74±0.08	96.41±0.35	95.25±2.37	96.15±1.14	92.78±4.26	83.38±1.14	87.95±4.96
	AUPRC	**99.56**±0.98	96.67±0.44	**98.72**±0.17	96.68±0.63	89.76±1.24	95.35±0.35	96.51±0.33	95.39±2.44	96.27±1.50	94.35±3.12	83.76±1.29	86.36±6.67
ADFTD (3)	Accuracy	**56.81**±1.02	56.73±2.51	54.54±0.70	**57.56**±0.93	53.91±0.54	56.12±0.11	53.27±1.54	51.85±2.92	54.60±0.82	47.60±4.63	52.60±1.59	43.97±1.63
	Precision	**56.05**±1.57	54.66±2.39	53.02±0.87	**56.01**±0.81	51.89±0.49	55.07±0.09	51.02±1.57	48.75±1.11	50.33±1.24	44.09±5.18	46.78±1.27	42.14±1.46
	Recall	**56.88**±2.04	54.60±3.02	49.25±1.01	**57.41**±0.79	52.10±1.09	55.47±0.34	50.71±1.55	48.25±1.37	49.17±0.89	43.83±3.71	47.27±1.29	41.36±0.85
	F1 Score	**56.40**±1.28	53.82±3.03	48.84±1.72	**55.98**±0.82	51.79±0.74	55.00±0.24	50.65±1.51	47.66±1.10	48.91±1.13	43.27±4.53	46.79±1.13	41.26±1.16
	AUROC	**75.77**±1.55	74.81±2.08	68.67±1.05	**75.72**±0.59	72.11±0.73	74.68±0.33	70.93±1.19	68.19±0.92	70.08±0.98	61.35±3.55	67.26±1.16	59.29±1.04
	AUPRC	**58.94**±1.76	58.07±2.72	50.62±1.26	**58.85**±0.67	50.92±0.78	57.51±0.38	51.21±1.32	51.06±1.30	50.90±1.12	42.57±4.44	49.53±1.21	41.49±1.32
BCI-2a (4)	Accuracy	**51.97**±0.82	**46.68**±1.56	43.90±0.71	44.47±1.47	33.23±1.53	38.91±3.15	42.40±0.69	35.95±1.90	40.87±7.22	36.58±1.96	33.58±1.44	40.64±1.11
	Precision	**52.59**±0.69	**47.24**±1.64	44.32±0.59	44.65±1.89	32.97±1.62	39.39±2.89	42.90±0.72	36.41±1.77	41.68±7.35	35.09±3.21	33.88±1.35	40.78±1.29
	Recall	**51.98**±1.14	**46.68**±1.56	43.90±0.71	44.47±1.47	33.23±1.53	38.91±3.15	42.40±0.69	35.95±1.90	40.87±7.22	36.58±1.96	33.58±1.44	40.64±1.11
	F1 Score	**51.67**±0.77	**46.35**±1.59	43.30±0.82	42.18±2.04	32.89±1.49	38.50±3.28	42.29±0.72	35.24±2.22	38.69±8.67	34.17±2.85	33.23±1.47	40.13±1.30
	AUROC	**77.56**±0.43	**72.49**±1.12	69.68±0.51	70.01±1.09	59.44±0.48	65.15±2.29	69.09±0.23	63.48±1.49	68.57±6.07	63.46±2.50	59.57±0.75	67.63±0.76
	AUPRC	**56.41**±1.01	**47.63**±1.53	44.27±0.57	45.71±1.89	32.60±0.88	38.64±2.65	43.51±0.31	36.55±1.71	44.23±8.13	37.72±2.00	32.87±0.87	41.05±0.88
BCI-2b (2)	Accuracy	**75.04**±0.75	72.83±1.12	66.78±0.75	73.18±1.01	65.58±2.10	67.78±2.12	69.40±0.35	65.10±1.81	71.28±1.02	63.46±2.54	65.94±0.61	**73.26±0.65**
	Precision	**75.15**±0.70	72.92±1.08	67.83±0.75	**73.84**±0.81	65.94±2.33	68.04±2.03	69.66±0.33	65.41±1.59	71.61±0.95	64.45±3.16	66.15±0.68	73.61±0.69
	Recall	**75.04**±0.74	72.83±1.12	66.78±0.75	73.18±1.01	65.58±2.09	67.78±2.12	69.40±0.35	65.10±1.81	71.28±1.02	63.46±2.54	65.94±0.61	**73.26±0.65**
	F1 Score	**75.01**±0.75	72.81±1.13	66.73±0.76	72.99±1.08	65.45±2.21	67.63±2.20	69.30±0.36	64.89±2.01	71.16±1.08	62.90±2.48	65.84±0.58	**73.17±0.68**
	AUROC	**83.59**±0.72	80.38±1.04	72.78±1.59	**81.99**±0.75	72.25±2.24	71.31±2.41	76.03±0.45	71.35±2.00	78.48±1.66	70.56±3.37	73.17±1.36	81.69±0.33
	AUPRC	**82.81**±0.91	79.98±1.22	72.11±1.52	81.33±0.73	71.93±2.38	72.70±3.13	75.30±0.40	70.37±2.21	77.88±1.86	68.65±3.63	71.53±1.43	**81.36±0.41**
PTB (2)	Accuracy	**91.01**±1.22	87.73±1.73	85.96±2.52	**89.36**±0.45	85.39±0.63	84.53±0.28	83.50±2.01	82.47±1.35	83.72±0.83	84.59±3.69	83.89±0.71	74.74±1.62
	Precision	**91.88**±1.85	88.71±1.28	89.92±0.74	**90.01**±0.48	86.01±0.72	87.35±0.45	85.19±0.94	86.57±2.32	86.70±0.67	85.72±2.12	88.25±1.18	76.94±1.51
	Recall	**87.33**±1.61	82.63±2.80	79.43±4.13	**85.70**±1.02	80.87±0.91	77.90±0.66	77.11±3.39	74.51±1.53	75.74±1.28	78.15±6.41	76.39±1.01	63.89±2.71
	F1 Score	**89.08**±1.49	84.71±2.51	81.97±4.07	**87.36**±0.96	82.41±1.11	80.40±0.62	79.18±3.31	77.01±1.78	78.42±1.30	79.97±6.15	79.06±1.06	64.36±3.38
	AUROC	**95.14**±1.60	**94.71**±0.94	94.57±0.70	90.13±0.93	92.99±0.98	93.31±0.46	92.81±1.48	91.71±2.38	94.38±0.76	88.19±4.72	91.18±1.16	88.79±0.91
	AUPRC	**96.13**±1.98	93.53±0.96	**94.36±0.66**	88.73±1.11	92.84±1.03	87.74±2.54	90.32±1.54	90.95±2.41	93.00±0.88	91.54±3.91	90.93±0.98	83.39±0.96
PTB-XL (5)	Accuracy	**75.19**±0.54	73.37±0.39	73.53±0.07	73.86±0.30	**73.98**±0.21	73.87±0.18	72.87±0.23	73.77±0.41	73.94±0.20	73.17±0.68	69.28±0.22	73.23±0.25
	Precision	**67.37**±0.79	65.62±0.52	65.92±0.52	66.15±0.67	66.45±0.38	66.26±0.29	64.14±0.42	66.30±0.90	65.67±0.20	**66.99**±1.28	59.59±0.45	65.70±0.64
	Recall	**63.50**±0.47	60.81±0.36	60.61±0.59	61.30±1.08	**62.21**±0.21	61.13±0.23	60.60±0.46	62.17±0.49	61.97±0.52	59.20±1.45	54.62±0.18	60.82±0.76
	F1 Score	**64.91**±0.47	62.56±0.30	62.44±0.27	63.20±0.91	**63.51**±0.34	62.54±0.20	62.02±0.37	63.09±0.12	62.97±0.52	61.73±1.17	56.20±0.19	62.61±0.34
	AUROC	**91.46**±0.56	90.37±0.12	90.03±0.12	89.33±0.22	90.41±0.11	90.21±0.15	89.66±0.13	89.86±0.17	**90.59**±0.09	88.60±0.48	86.71±0.10	89.74±0.34
	AUPRC	**69.74**±1.06	67.61±0.35	67.19±0.25	66.91±0.75	67.55±0.28	66.01±0.88	66.39±0.22	67.24±0.32	**68.46**±0.24	66.64±1.14	60.27±0.21	67.32±0.22
UCI-HAR (6)	Accuracy	**95.44**±0.39	90.14±0.65	**94.15**±0.96	92.90±0.51	90.87±1.21	89.52±0.34	89.62±0.81	83.94±1.33	88.68±1.54	90.98±4.91	92.41±0.63	87.67±0.39
	Precision	**95.78**±0.35	90.82±0.56	**94.27**±0.96	93.30±0.48	90.61±1.01	90.54±0.31	89.70±0.18	85.30±1.02	88.84±1.26	92.15±3.39	92.24±0.63	88.37±0.43
	Recall	**95.69**±0.41	90.33±0.65	**94.30**±0.97	93.11±0.49	90.87±1.21	89.83±0.29	89.80±0.14	84.25±1.27	88.82±1.05	91.40±4.63	92.33±0.67	87.97±0.37
	F1 Score	**95.69**±0.40	90.49±0.63	**94.26**±0.98	93.14±0.50	90.66±1.19	89.99±0.29	89.62±0.81	84.56±1.22	88.60±1.33	91.40±4.57	92.16±0.64	88.02±0.38
	AUROC	**99.53**±0.11	98.87±0.11	**99.32**±0.05	99.26±0.06	98.67±0.79	98.90±0.11	98.11±0.06	97.71±0.27	98.77±0.76	99.01±0.75	99.07±0.07	98.50±0.09
	AUPRC	**97.86**±0.17	95.45±0.38	**96.74**±0.18	96.31±0.14	94.55±0.83	94.73±0.42	94.80±0.72	91.02±0.98	94.93±1.17	95.87±2.85	96.01±0.39	93.86±0.40

4.1 Main Results

Subject-Independent (SI) Evaluation. As shown in Table 1, ProtoRoute exhibits remarkably consistent gains under unseen-subject evaluation, ranking first in **46 of 48** dataset–metric comparisons and second in the remaining two. Notably, it achieves the best F1, precision, AUROC and AUPRC on **all eight benchmarks**. Particularly, ProtoRoute improves F1 over the strongest competing method by **2+ percentage points on average**, with gains exceeding one point on seven datasets and reaching up to **5 points**. This consistency across clinical diagnosis, neural decoding, and wearable sensing provides strong empirical support for the dual-timescale persistent evidence learning.

4.2 Ablation Studies

Component Ablation. Table 2 shows that the persistent pathway is a key contributor to the performance of ProtoRoute. Across the four ablation datasets, removing PER leads to the largest average F1 degradation, while removing NFR, or multi-view tokenization (MVT) results in smaller or less consistent performance drops. These results highlight the central role of PER, with the remaining fast-encoder components providing additional complementary benefits.

Within PER, dense prototype or token aggregation, flat routing, single-prototype modeling, and pooled memory updates consistently underperform the full sparse token-level design, supporting the joint roles of hierarchical multi-prototype routing and sparse evidence aggregation. Within LAPE, random replacement is particularly detrimental, while the remaining maintenance ablations also produce systematic degradation, showing that performance depends not only on storing additional

Table 2: **Ablation studies of ProtoRoute under the subject-independent setting.** r=M denotes aggregation over all class prototypes and q=L denotes aggregation over all token-level evidence; detailed variant definitions and analyses are deferred to Appendix C.5.1.

Model	APAVA			TDBRAIN			BCI-2a			PTB-XL		
	Accuracy	F1	AUROC	Accuracy	F1	AUROC	Accuracy	F1	AUROC	Accuracy	F1	AUROC
ProtoRoute	**90.59±1.16**	**89.96±1.34**	**96.94±0.09**	**95.67±0.98**	**95.66±0.98**	**99.52±0.19**	**51.97±0.82**	**51.67±0.77**	**77.56±0.43**	**75.19±0.54**	**64.91±0.47**	**91.46±0.56**
Architectural Components												
w/o PER	86.19±3.50	86.08±4.29	95.17±1.88	92.48±2.81	92.43±2.84	98.79±0.63	48.44±1.07	47.94±1.10	74.17±0.63	72.89±0.48	61.55±0.43	87.64±0.39
w/o NFR	87.56±2.32	86.35±2.75	95.83±1.84	94.56±4.75	94.51±4.85	99.26±0.87	50.80±2.39	50.38±2.36	76.55±1.33	74.77±0.30	64.78±0.28	90.87±0.26
w/o MVT	85.10±1.98	83.31±2.51	94.07±1.74	93.79±1.17	93.78±1.17	98.97±0.40	50.59±1.04	49.96±1.16	76.26±0.98	74.76±0.29	64.78±0.27	90.86±0.26
K-means Initialization	89.14±3.71	88.20±4.41	95.69±3.07	94.60±1.40	94.60±1.40	99.30±0.44	51.13±1.10	50.70±1.57	76.93±0.91	74.92±0.18	64.87±0.57	91.02±0.08
Hierarchical Routing and Sparse Evidence Aggregation												
Dense Prototype r=M	89.22±3.75	88.28±4.45	95.82±3.14	95.33±0.96	95.33±0.97	99.42±0.27	50.63±1.55	50.10±1.87	76.50±1.20	74.71±0.42	64.23±0.29	90.81±0.32
Dense Tokens q=L	89.35±2.94	88.50±3.42	96.13±2.15	95.36±1.30	95.36±1.30	99.40±0.32	50.69±1.16	50.09±1.34	76.26±0.99	74.74±0.11	64.23±0.45	90.90±0.18
Flat Routing	88.60±2.73	87.62±3.17	96.02±1.72	95.25±1.21	95.24±1.21	99.45±0.26	50.46±1.22	50.03±1.68	75.98±0.88	74.21±0.33	63.92±0.39	90.07±0.40
Proto-logit-only	89.73±3.11	88.93±3.66	96.40±1.98	95.48±1.37	95.47±1.38	99.52±0.28	51.16±1.29	50.76±1.73	76.93±0.92	75.10±0.53	64.87±0.47	90.88±0.57
Retrieval-only	89.88±2.37	89.15±2.70	96.03±2.33	95.53±1.29	95.53±1.29	99.49±0.25	51.18±1.26	50.77±1.72	76.93±0.93	75.09±0.54	64.87±0.48	90.88±0.57
Single Prototype	87.07±3.06	85.69±3.84	95.52±1.19	95.33±1.11	95.33±1.11	99.45±0.25	50.48±1.17	50.03±1.58	75.97±0.90	74.22±0.38	63.96±0.42	89.96±0.60
Pooled Update	89.18±3.74	88.26±4.45	95.60±3.03	95.58±1.57	95.58±1.57	99.50±0.27	50.51±1.14	50.51±1.59	75.99±0.92	74.59±0.36	64.73±0.36	90.48±0.36
LAPE Memory Maintenance												
Fixed Budget	89.46±2.11	88.84±2.34	95.82±1.34	94.71±1.12	94.70±1.12	98.58±0.25	50.90±1.25	49.66±1.71	76.33±0.93	74.23±0.33	63.96±0.26	90.01±0.32
No Admission	88.90±2.73	88.12±3.22	95.51±1.63	94.10±1.05	94.09±1.06	98.51±0.28	50.37±1.25	49.96±1.71	76.03±0.93	73.98±0.85	63.63±0.66	89.36±0.71
No Diversity	89.20±1.84	88.47±2.12	96.02±1.07	94.56±1.03	94.55±1.03	98.61±0.21	50.58±1.24	50.15±1.66	76.09±0.88	74.20±0.53	63.97±0.47	89.98±0.58
Random Replacement	87.61±3.38	86.61±4.05	94.57±3.01	94.12±1.56	94.11±1.57	98.51±0.34	50.01±1.39	49.59±1.29	75.46±1.25	73.31±0.38	62.96±0.90	89.63±0.74
Usage-only Replacement	89.43±2.02	88.77±2.31	95.75±1.49	94.43±1.66	94.43±1.67	98.57±0.33	50.54±1.26	50.12±1.70	76.09±0.88	74.20±0.53	63.97±0.47	89.98±0.57
EMA Update	89.09±2.56	88.37±2.92	95.12±2.47	94.20±1.14	94.20±1.14	98.52±0.28	50.30±1.30	49.89±1.74	76.03±0.92	73.79±0.75	63.57±0.59	89.48±0.93
Prototype Regularization												
w/o $\mathcal{L}_{\text{proto}}$	90.11±1.95	89.45±2.20	96.26±1.37	95.52±1.20	95.52±1.20	99.53±0.26	51.37±1.19	50.93±1.64	76.99±0.87	74.75±0.12	64.23±0.45	90.90±0.18
w/o $\mathcal{L}_{\text{margin}}$	90.34±1.20	89.71±1.28	96.64±1.02	95.50±1.17	95.50±1.18	99.52±0.24	51.04±1.31	50.61±1.75	76.96±0.87	74.82±0.15	64.55±0.19	90.80±0.16
w/o $\mathcal{L}_{\text{proto}}$&$\mathcal{L}_{\text{margin}}$	89.99±1.25	89.28±1.35	96.15±1.12	95.04±0.98	95.04±0.98	99.39±0.30	50.94±1.29	50.51±1.72	76.96±0.88	74.53±0.37	64.22±0.35	90.72±0.21

evidence but also on how candidates are admitted, selected, and existing slots are replaced. Finally, removing both prototype-aware objectives is generally more detrimental than removing either alone, supporting the complementary roles of prototype alignment and hard-negative separation.

Why Should Persistent Evidence Evolve Slowly? We compare a spectrum of persistent-state update regimes, ranging from gradient and mini-batch updates to epoch-wise, multi-epoch, and frozen updates. Beyond predictive performance, we measure *Dictionary drift*, *Slot turnover*, and *Validation volatility*, which respectively characterize prototype-state movement, evidence replacement, and training instability. **For fair comparison, all variants are trained for the same number of epochs (100 epochs)**, while test results use the best validation checkpoint. As shown in Table 3, rapidly updated prototype states are less stable, with batch-wise updates yielding the highest validation volatility. Conversely, slowing the state transition improves stability, but an overly slow or frozen dictionary sacrifices adaptation. Epoch-wise LAPE achieves the best F1, revealing a clear *plasticity–stability trade-off*: persistent class evidence should evolve more slowly than the parametric encoder, while remaining sufficiently adaptive to track the evolving representation space. More detailed analysis is in Appendix C.5.2.

Table 3: **Effect of persistent-state update timescale on TDBRAIN.** All variants share the same parametric backbone and differ only in the evolution of the class-level state.

Variant	State	Update	F1 ↑	Drift	Turn.	Vol. ↓
Parametric only	none	gradient	94.13±2.87	–	–	0.013
Gradient prototypes	parametric B	opt. step	94.79±4.17	–	–	0.015
Batch-wise non-param.	buffer	mini-batch	94.85±1.14	**0.032**	**0.11±0.12**	**0.016**
Epoch-wise LAPE	**buffer**	**epoch**	**96.16±1.10**	0.020	0.07±0.00	0.011
5-epoch LAPE	buffer	5 epochs	95.70±1.61	0.014	0.02±0.00	0.010
Frozen dictionary	buffer	init. only	95.53±2.25	0	0	0.009

5 Conclusion

We introduced ProtoRoute, a dual-timescale framework that separates transient **within-sample** representation learning from persistent **cross-sample** class-evidence learning. Experiments on 8 EEG, ECG, neural-decoding and wearable-sensing benchmarks demonstrate consistent gains in subject-independent generalization, robustness, and computational efficiency. These results suggest that *transferable class knowledge need not be encoded exclusively in network parameters*; explicitly *maintaining and slowly evolving a persistent evidence state provides a complementary learning mechanism* that can be attached to different parametric time-series encoders.

LLM USAGE

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text.

It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis. The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. In this study, no human subjects or animal experimentation was involved. All datasets used, including APAVA, ADFTD, TDBRAIN, PTB, PTB-XL, BCI-2a, BCI-2b, and UCI-HAR, were sourced in compliance with relevant usage guidelines, ensuring no violation of privacy. We have taken care to avoid any biases or discriminatory outcomes in our research process. No personally identifiable information was used, and no experiments were conducted that could raise privacy or security concerns. We are committed to maintaining transparency and integrity throughout the research process.

REPRODUCIBILITY STATEMENT

We make extensive efforts to ensure that the experiments in this work are reproducible. The paper specifies the complete dual-timescale training procedure, including the mini-batch optimization of the parametric state and the epoch-level evolution of the persistent class-evidence dictionary. Detailed pseudocode for the overall training procedure and Loss-Adaptive Prototype Evolution (LAPE) is provided in the Appendix. We report the model architecture, optimization settings, and all major training hyperparameters. All reported main results are evaluated over five random seeds (41–45), and mean and standard deviation are reported. We further provide detailed dataset statistics, preprocessing procedures, subject-independent splitting protocols, baseline configurations, ablation settings, sensitivity analyses, robustness experiments, and persistent-state dynamics. All eight datasets used in this work are publicly available benchmarks, which facilitates independent reproduction of the experimental protocol and comparison with prior work.

REFERENCES

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3–4, 2013.

Benjamin Blankertz, K-R Muller, Gabriel Curio, Theresa M Vaughan, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Christa Neuper, Gert Pfurtscheller, Thilo Hinterberger, et al. The bci competition 2003: progress and perspectives in detection and discrimination of eeg single trials. *IEEE transactions on biomedical engineering*, 51(6):1044–1051, 2004.

Benjamin Blankertz, K-R Muller, Dean J Krusienski, Gerwin Schalk, Jonathan R Wolpaw, Alois Schlogl, Gert Pfurtscheller, Jd R Millan, Michael Schroder, and Niels Birbaumer. The bci competition iii: Validating alternative approaches to actual bci problems. *IEEE transactions on neural systems and rehabilitation engineering*, 14(2):153–159, 2006.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Timemil: Advancing multivariate time series classification via a time-aware multiple instance learning. *Proceedings of Machine Learning Research*, 235:7190–7206, 2024. ISSN 2640-3498. Publisher Copyright: Copyright 2024 by the author(s); 41st International Conference on Machine Learning, ICML 2024 ; Conference date: 21-07-2024 Through 27-07-2024.

Guikang Du, Haoran Li, Xinyu Liu, Zhibo Zhang, Xiaoli Gong, and Jin Zhang. Bioformer: Rethinking cross-subject generalization via spectral structural alignment in biomedical time-series. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=6I7yMzQBts.

J Escudero, Daniel Abásolo, Roberto Hornero, Pedro Espino, and Miguel López. Analysis of electroencephalograms in alzheimer's disease patients with multiscale entropy. *Physiological measurement*, 27(11):1091–1106, 2006.

Wei Fan, Jingru Fei, Dingyu Guo, Kun Yi, Xiaozhuang Song, Haolong Xiang, Hangting Ye, and Min Li. Towards multi-resolution spatiotemporal graph learning for medical time series classification. In *Proceedings of the ACM on Web Conference 2025*, pp. 5054–5064, 2025.

Yu Guoqi, Juncheng Wang, Chen Yang, Jing Qin, Angelica Aviles-Rivero, and Shujun Wang. Decentralized attention fails centralized signals: Rethinking transformers for medical time series. In *International Conference on Learning Representations*, volume 2026, pp. 108935–108955, 2026.

Nan Huang, Haishuai Wang, Zihuai He, Marinka Zitnik, and Xiang Zhang. Repurposing foundation model for generalizable medical time series classification. In *International Conference on Learning Representations*, volume 2026, pp. 83809–83831, 2026.

Huayu Li, ZhengXiao He, Xiwen Chen, Jingjing Wang, Siyuan Tian, Jinghao Wen, and Ao Li. Learning fingerprints for medical time series with redundancy-constrained information maximization. *arXiv preprint arXiv:2605.00130*, 2026.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

Junnan Li, Caiming Xiong, and Steven Hoi. Mopro: Webly supervised learning with momentum prototypes. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0-EYBhgw80y.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Zhen Liu, Yicheng Luo, Boyuan Li, Emadeldeen Eldele, Min Wu, and Qianli Ma. Learning soft sparse shapes for efficient time-series classification. *arXiv preprint arXiv:2505.06892*, 2025.

Andreas Miltiadous, Katerina D Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G Tsalikakis, Pantelis Angelidis, Markos G Tsipouras, Euripidis Glavas, Nikolaos Giannakeas, et al. A dataset of scalp eeg recordings of alzheimer's disease, frontotemporal dementia and healthy subjects from routine eeg. *Data*, 8(6):95, 2023.

Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 903–913, 2019.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Jing Peng, Meiqi Yang, Qiong Zhang, and Xiaoxiao Li. S4m: S4 for multivariate time series forecasting with missing values. In *International Conference on Learning Representations*, volume 2025, pp. 49369–49395, 2025.

PhysioToolkit PhysioBank. Physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Sahil Sethi, David Chen, Thomas Statchen, Michael C Burkhart, Nipun Bhandari, Bashar Ramadan, and Brett Beaulieu-Jones. Protoecgnet: case-based interpretable deep learning for multi-label ecg classification with contrastive learning. *Proceedings of machine learning research*, 298:https–proceedings, 2025.

Hanneke Van Dijk, Guido Van Wingen, Damiaan Denys, Sebastian Olbrich, Rosalinde Van Ruth, and Martijn Arns. The two decades brainclinics research archive for insights in neurophysiology (tdbrain) database. *Scientific data*, 9(1):333, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. Contrast everything: A hierarchical contrastive framework for medical time-series. *Advances in Neural Information Processing Systems*, 36:55694–55717, 2023.

Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. Medformer: A multi-granularity patching transformer for medical time-series classification. *Advances in Neural Information Processing Systems*, 37:36314–36341, 2024.

Jiexia Ye, Weiqi Zhang, Ziyue Li, Jia Li, Meng Zhao, and Fugee Tsung. Medualtime: A dual-adapter language model for medical time series-text multimodal learning. *arXiv preprint arXiv:2406.06620*, 2024.

Jiexia Ye, Weiqi Zhang, Ziyue Li, Jia Li, and Fugee Tsung. Medspaformer: a transferable transformer with multi-granularity token sparsification for medical time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27791–27799, 2026.

Xiaoyan Yuan, Wei Wang, Junxin Chen, and Xiping Hu. Reading between the channels: Knowledge-augmented medical time series classification. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 8978–8987, 2025.

Da Zhang, Bingyu Li, Zhiyuan Zhao, Hongyuan Zhang, Junyu Gao, and Xuelong Li. Medmamba: Multi-view state space models with adaptive graph learning for medical time series classification. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=qPqJH0heR0.

Algorithm 1 Dual-Timescale Training of ProtoRoute

Input: Training set $\mathcal{D}_{\mathrm{tr}} = \{(\boldsymbol{X}_n, y_n)\}$; validation set $\mathcal{D}_{\mathrm{val}}$; maximum epochs E; warm-up length E_{warm}; memory schedule $\lambda_{\mathcal{B}}(e)$; parametric model θ; dictionary $\mathcal{B}$ with class-wise initialization flags.
Output: Best validation checkpoint $(\theta^\star, \mathcal{B}^\star)$.

Initialize θ, normalized dictionary buffers, usage statistics $\mathcal{U}_0$, and initialization flags
$\boldsymbol{\mathcal{L}}^{(0)} \leftarrow \varnothing$; $v^\star \leftarrow -\infty$
for $e = 1, \ldots, E$ **do**
  Set memory strength to $\lambda_{\mathcal{B}}(e)$
  $\mathcal{C}_e \leftarrow \varnothing$; $\mathcal{Y}_e \leftarrow \varnothing$; $\mathcal{O}_e \leftarrow \varnothing$
  Set the model to training mode
  for each mini-batch $(\boldsymbol{X}, \boldsymbol{y})$ **do**
    $(\boldsymbol{\ell}, \mathcal{A}) \leftarrow f_{\theta,\mathcal{B}}(\boldsymbol{X})$ ▷ $\mathcal{A}$ contains query tokens and routing statistics
    $\chi_{\mathcal{B}} \leftarrow \mathbb{I}[\textsc{Ready}(\mathcal{B})]$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{cls}} + \chi_{\mathcal{B}}\,(\lambda_p \mathcal{L}_{\mathrm{proto}} + \lambda_m \mathcal{L}_{\mathrm{margin}})$
    $\theta \leftarrow \textsc{OptimizerStep}(\theta, \nabla_\theta \mathcal{L})$ ▷ $\mathcal{B}$ is fixed; active usage statistics are accumulated
    $\mathcal{C}_e \leftarrow \mathcal{C}_e \cup \textsc{MineQueryTokens}(\mathcal{A}, \boldsymbol{y}, \boldsymbol{\ell})$
    Append $(\boldsymbol{y}, \boldsymbol{\ell})$ to $(\mathcal{Y}_e, \mathcal{O}_e)$
  end for
  Update the learning-rate scheduler
  $\boldsymbol{\mathcal{L}}^{(e)} \leftarrow \textsc{ClasswiseLoss}(\mathcal{O}_e, \mathcal{Y}_e)$
  if $e > E_{\mathrm{warm}}$ and $\mathcal{C}_e \neq \varnothing$ **then**
    if initialization mode is K-means and $\neg\textsc{Ready}(\mathcal{B})$ **then**
      $\mathcal{B} \leftarrow \textsc{KMeansInitialize}(\mathcal{C}_e)$
    else
      $\mathcal{B} \leftarrow \mathrm{LAPE}(\mathcal{B}, \mathcal{C}_e, \boldsymbol{\mathcal{L}}^{(e)}, \boldsymbol{\mathcal{L}}^{(e-1)}, \mathcal{U}_e)$ ▷ LAPE bootstraps uninitialized classes
    end if
  end if
  $\boldsymbol{\mathcal{L}}^{(e-1)} \leftarrow \boldsymbol{\mathcal{L}}^{(e)}$
  $v_e \leftarrow \textsc{EvaluateFrozen}(f_{\theta,\mathcal{B}}, \mathcal{D}_{\mathrm{val}})$ ▷ No state mutation during evaluation
  if $v_e > v^\star$ **then**
    $(\theta^\star, \mathcal{B}^\star) \leftarrow (\theta, \mathcal{B})$; $v^\star \leftarrow v_e$
    Save optimizer, scheduler, and class-loss state
  end if
  if early-stopping criterion is met **then**
    break
  end if
end for
return $(\theta^\star, \mathcal{B}^\star)$

A ALGORITHMS

This section presents the complete training procedure of ProtoRoute. Algorithm 1 describes the dual-timescale training loop that alternates between fast parametric updates and slow dictionary evolution. Algorithm 2 details the loss-adaptive prototype evolution (LAPE) operator, which is called once per epoch to update the persistent dictionary.

Algorithm 1: Dual-Timescale Training. The main training loop follows a warm-up–ramp–alternation schedule. During warm-up, the dictionary is inactive, allowing the parametric encoder to establish a stable representation space. After initialization, the dictionary is updated at epoch boundaries using candidates collected across all mini-batches. Validation is always performed with a frozen dictionary to prevent data leakage.

Algorithm 2: Loss-Adaptive Prototype Evolution (LAPE). Algorithm 2 details the epoch-level dictionary evolution operator. It processes each class independently, performing: (1) bootstrap initialization for uninitialized classes; (2) reliability-aware admission; (3) loss-adaptive budget allocation; (4) diversity-aware candidate selection; and (5) usage–redundancy–conflict slot replacement.

Algorithm 2 Loss-Adaptive Prototype Evolution (LAPE)

Input: Dictionary $\mathcal{B}_e$; class-wise initialization flags; candidate tuples $\mathcal{C}_e$ containing normalized token features, labels, and prediction probabilities; current and previous class losses $\boldsymbol{\mathcal{L}}^{(e)}$ and $\boldsymbol{\mathcal{L}}^{(e-1)}$; usage statistics $\mathcal{U}_e$.
Output: Updated dictionary $\mathcal{B}_{e+1}$.

for each class $k = 1, \ldots, K$ **do**
  $\mathcal{P}_{e,k} \leftarrow \{(\boldsymbol{z}, p_k) \in \mathcal{C}_e : y = k\}$
  if $\mathcal{P}_{e,k} = \varnothing$ **then**
    continue
  end if
  if class k is not initialized **then**
    $\mathcal{P}_{e,k}^{\text{boot}} \leftarrow \{(\boldsymbol{z}, p_k, s) \in \mathcal{P}_{e,k} : p_k \geq \theta_{\text{conf}}\}$
    if $\mathcal{P}_{e,k}^{\text{boot}} = \varnothing$ and bootstrap fallback is enabled **then**
      $\mathcal{P}_{e,k}^{\text{boot}} \leftarrow \mathcal{P}_{e,k}$
    end if
    $\mathcal{S}_{e,k} \leftarrow \text{DIVERSESELECT}(\mathcal{P}_{e,k}^{\text{boot}}, \text{score} = p_k, \text{budget} = M)$
    if $\mathcal{S}_{e,k} \neq \varnothing$ **then**
      Repeat selected candidates if necessary to fill all M slots
      Initialize $\mathcal{B}_{e,k}$ with the normalized selected candidates
      Mark class k as initialized and reset $\mathcal{U}_{e,k}$
    end if
    continue
  end if
  Compute confidence $c_k(\boldsymbol{z})$, novelty $n_k(\boldsymbol{z})$, cross-class compatibility $\xi_k(\boldsymbol{z})$, and separation $d_k(\boldsymbol{z})$
  Form the admissible set
  $\mathcal{A}_{e,k} \leftarrow \{\boldsymbol{z} \in \mathcal{P}_{e,k} : c_k(\boldsymbol{z}) \geq \theta_{\text{conf}},\ n_k(\boldsymbol{z}) \geq \theta_{\text{nov}},\ \xi_k(\boldsymbol{z}) \leq \theta_{\text{inter}}\}$
  if $\mathcal{A}_{e,k} = \varnothing$ **then**
    continue
  end if
  $\Delta_k^{(e)} \leftarrow \mathcal{L}_k^{(e-1)} - \mathcal{L}_k^{(e)}$
  $\gamma_k^{(e)} \leftarrow \text{clip}(-\alpha\Delta_k^{(e)} + \beta, \gamma_{\min}, \gamma_{\max})$
  $R_k^{(e)} \leftarrow \max\{1, \text{round}(\gamma_k^{(e)} M)\}$
  Compute $Q_k(\boldsymbol{z}) = w_c c_k(\boldsymbol{z}) + w_n n_k(\boldsymbol{z}) + w_d d_k(\boldsymbol{z})$ for $\boldsymbol{z} \in \mathcal{A}_{e,k}$
  $\mathcal{S}_{e,k} \leftarrow \text{DIVERSESELECT}(\mathcal{A}_{e,k}, Q_k, R_k^{(e)})$
  if $\mathcal{S}_{e,k} = \varnothing$ **then**
    continue
  end if
  Compute slot priorities $P_{k,m} = w_u P_{\text{use}}(k,m) + w_r P_{\text{red}}(k,m) + w_x P_{\text{conf}}(k,m)$
  $J_k \leftarrow |\mathcal{S}_{e,k}|$
  $\mathcal{R}_{e,k} \leftarrow \text{Top}_{J_k}\left(\{P_{k,m}\}_{m=1}^{M}\right)$
  $\boldsymbol{Z}_{e,k}^{\star} \leftarrow [\boldsymbol{z}_1^{\star}; \ldots; \boldsymbol{z}_{J_k}^{\star}]$
  $\mathcal{B}_k[\mathcal{R}_{e,k}] \leftarrow \text{RowNorm}(\boldsymbol{Z}_{e,k}^{\star})$
  $U_{k,m} \leftarrow 0, \quad \forall m \in \mathcal{R}_{e,k}$
end for
return $\mathcal{B}_{e+1}$

B Experimental Setup

B.1 Dataset Details

We evaluate ProtoRoute on 8 public benchmarks spanning three application families. The clinical-diagnosis suite contains the EEG datasets APAVA (Escudero et al., 2006), ADFTD (Miltiadous et al., 2023) and TDBRAIN (Van Dijk et al., 2022), together with the ECG datasets PTB (PhysioBank, 2000) and PTB-XL (Wagner et al., 2020). The neural-decoding suite contains the motor-imagery benchmarks BCI-2a and BCI-2b (Blankertz et al., 2004; 2006). We further include UCI-HAR (Anguita et al., 2013), an inertial activity-recognition datasets, to assess whether persistent evidence learning transfers beyond electrophysiological signals.

Table 4: **The information of utilized datasets,** including the number of subjects, samples, classes, sample channels, and timestamps.

Dataset	Application	Modality	Subjects	Samples	Classes	Channel	Timestamps	Frequency
APAVA	Clinical diagnosis	EEG	23	5,967	2	16	256	256
ADFTD	Clinical diagnosis	EEG	88	69,752	3	19	256	256
TDBRAIN	Clinical diagnosis	EEG	72	6,240	2	33	256	256
PTB	Clinical diagnosis	ECG	198	64,356	2	15	300	250
PTB-XL	Clinical diagnosis	ECG	17,596	191,400	5	12	250	250
BCI-2a	Neural decoding	EEG	9	5,184	4	3	500	250
BCI-2b	Neural decoding	EEG	9	6,520	2	3	262	250
UCI-HAR	Activity recognition	Inertial	30	10,299	6	9	128	50

B.1.1 SUBJECT-INDEPENDENT EVALUATION

All benchmarks use subject-disjoint training, validation, and test partitions: every session, trial, and derived window from one subject is assigned to exactly one split. Data-dependent preprocessing statistics are estimated from the training partition only. Prototype candidates and slot-usage statistics are collected exclusively from training data. During validation and testing, the parametric model, persistent dictionary, and usage statistics are frozen, preventing evaluation subjects from modifying the learned class state.

B.1.2 GENERAL PROTOCOL

Let $\mathcal{S}_{\mathrm{tr}}$, $\mathcal{S}_{\mathrm{val}}$, and $\mathcal{S}_{\mathrm{te}}$ denote the training, validation, and test subject sets. All experiments satisfy

$$\mathcal{S}_{\mathrm{tr}} \cap \mathcal{S}_{\mathrm{val}} = \mathcal{S}_{\mathrm{tr}} \cap \mathcal{S}_{\mathrm{te}} = \mathcal{S}_{\mathrm{val}} \cap \mathcal{S}_{\mathrm{te}} = \varnothing. \tag{26}$$

Repeated sessions from the same participant are grouped before splitting, and segmentation is performed without allowing windows from one participant to cross split boundaries. We save the model with the best F1 score on the validation set.

We used patient-wise splitting protocols for all datasets to prevent data leakage. Regarding the allocation of training, validation, and testing sets, we follow the benchmark constructed and preprocessed by **Medformer (Neurips'24)** (Wang et al., 2024) and **Bioformer (ICML'26)** (Du et al., 2026).

B.1.3 BENCHMARKS DESCRIPTIONS

APAVA (Escudero et al., 2006) is a public EEG time series dataset with 2 classes and 23 subjects, including 12 patients with Alzheimer's disease and 11 healthy control subjects. **TDBRAIN** (Van Dijk et al., 2022) is a public EEG time series dataset, which is evaluated using the Parkinson's-disease-versus-healthy-control subset adopted by Medformer, restricted to the eyes-closed condition.**ADFTD** (Miltiadous et al., 2023) is an EEG dataset with a three-class label for each sample, categorizing the subject as Healthy, having Frontotemporal Dementia (FTD), or Alzheimer's disease (AD). **PTB** (PhysioBank, 2000) is an ECG dataset where each sample is assigned a binary label indicating whether the subject has Myocardial Infarction. **PTB-XL** (Wagner et al., 2020) is a large-scale ECG dataset, we use the processed five-class single-label benchmark adopted by Medformer. **BCI Competition IV 2a (BCI-2a)** (Blankertz et al., 2004) is a four-class motor-imagery EEG dataset containing left-hand, right-hand, foot, and tongue imagery from nine subjects. Following Bioformer, we retain only 3 bipolar channels (C3, Cz, and C4) that are most relevant to MI tasks. **BCI Competition IV 2b (BCI-2b)** contains binary left- versus right-hand motor imagery from nine subjects, recorded using C3, Cz, and C4. **UCI-HAR** (Anguita et al., 2013) is a smartphone-based human-activity dataset with six activity classes, containing nine-channel IMU signals sampled at 50 Hz.

B.1.4 PERSISTENT-STATE SAFEGUARDS

The persistent dictionary is maintained without access to validation or test data. During training, candidate tokens, predicted confidences, class-wise losses, and slot-usage statistics are collected only

Table 5: Common architecture and optimization hyperparameters.

Hyperparameter	Value
Embedding dimension D	128
Semantic stages / attention heads	4 / 4
Transformer FFN dimension	512
Temporal patch / stride	32 / 32
Frequency patch / stride	32 / 32
Channel pooling length	32
Routing stages	$\{2, 3, 4\}$
NFR stages	$\{3\}$
Model / local dropout	0.15 / 0.10
Dictionary slots per class M	64
Routing temperature τ	0.2
Top-r / Top-q	2 / 4
Optimizer	AdamW
Weight decay / LR decay	5×10^{-4} / 0.98
Max epochs / early-stop patience	100 / 10
Batch size / workers	128 / 4
Gradient clipping	4.0
Label smoothing	0.05
Seeds	41–45

from mini-batches drawn from $\mathcal{D}_{\mathrm{tr}}$. At validation and test time, candidate collection, usage-statistic accumulation, prototype initialization, and prototype replacement are all disabled. Consequently, the evaluated model state is fixed before observing any sample from $\mathcal{D}_{\mathrm{val}}$ or $\mathcal{D}_{\mathrm{te}}$.

B.2 Implementation Details

Evaluation and Reproducibility. All reported results use the common configuration summarized in Table 5. The initial learning rates: 1×10^{-3} for TDBRAIN and UCI-HAR, 1×10^{-4} for BCI-2a and BCI-2b, 3×10^{-4} for APAVA, ADFTD, and PTB-XL, and 5×10^{-4} for PTB. The maximum number of epochs was set to 100, with an early stopping mechanism (patience of 10 epochs) monitoring the **validation F1** score to prevent overfitting. The training process is conducted with five random seeds (41-45) to compute the mean and standard deviation.

Baselines. We compare ProtoRoute with representative methods from three categories: (i) ***Generic time-series models*** include PatchTST (Nie et al., 2022) and iTransformer (Liu et al., 2024); (ii) ***Medical time-series models*** include TeCh (Guoqi et al., 2026), Medformer (Wang et al., 2024), MedGNN (Fan et al., 2025), BioFormer (Du et al., 2026), MedMamba (Zhang et al., 2026), TS-Fingerprint (Li et al., 2026) and KEMed (Yuan et al., 2025), representing recent approaches tailored to physiological structure and cross-subject generalization; (iii) ***Evidence-aware models*** include TimeMIL (Chen et al., 2024) and SoftShape (Liu et al., 2025), which explicitly emphasize sparse or localized discriminative patterns. Most baseline results are directly adopted from Medformer (NeurIPS'24) (Wang et al., 2024) and BioFormer (ICML'26) (Du et al., 2026), as they are evaluated under the same subject-independent protocol. For the few baseline–dataset combinations not reported in these works, we reproduce them using exactly the same baseline-specific configurations, data splits, preprocessing procedure, and evaluation setting as those used for the reported results. Therefore, all baseline results in our comparison are obtained under a consistent experimental protocol and configuration.

Model Configuration. All experiments use an embedding dimension of $D = 128$. Temporal and frequency tokenizers use a patch length of 32 and a stride of 32, while the channel tokenizer adaptively pools each channel to 32 values before projection. The semantic encoder contains four pre-normalized Transformer (Vaswani et al., 2017) stages, each with one encoder block, four attention heads, a feed-forward width of $4D = 512$, GELU activation, and dropout 0.15. The local morphology encoder is enabled in all default runs and uses dropout 0.10. Its four stages use a residual kernel of 3, a multi-kernel block with kernels $\{3, 7, 15\}$, and residual kernels of 5 with dilations 2 and 4, respectively; stages 2–4 downsample the temporal resolution by a factor of two. Semantic

Table 6: Persistent dictionary and LAPE hyperparameters.

Hyperparameter	Value
Initialization	bootstrap
Warm-up / ramp epochs	10 / 15
Maximum coupling strength	0.5
Selected tokens per sample	2
Confidence threshold	0.75
Minimum novelty	0.05
Maximum inter-class similarity	0.75
Candidate weights (conf./nov./sep.)	0.60 / 0.20 / 0.20
Candidate diversity weight	0.25
Plasticity α, β	0.5, 0.10
Replacement-fraction range	[0.05, 0.15]
Replacement weights (use/red./conf.)	0.45 / 0.35 / 0.20
Prototype loss weight λ_p	0.05
Margin loss weight λ_m	0.05
Prototype temperature τ_p	0.2
Hard-negative margin μ	0.2

and local features are fused after every stage, and the four attention-pooled stage representations are concatenated before the final classifier. Inputs are standardized independently for each channel along the temporal axis.

Persistent Dictionary and LAPE Configuration. The dictionary contains $M = 64$ normalized slots per class. Routing uses cosine temperature $\tau = 0.2$, Top-r prototype aggregation with $r = 2$, and Top-q token aggregation with $q = 4$. The prototype-logit gate is initialized with logit -3. Training starts with a 10-epoch parametric warm-up during which the dictionary coupling strength is zero and dictionary updates are disabled. The dictionary is then initialized by the bootstrap procedure from selected token-level query features. Its coupling strength is linearly increased for 15 epochs to a maximum value of 0.5. *Class-wise K-means initialization is supported as an alternative.*

For candidate construction, the two highest-scoring query tokens are retained from each sample. Tokens are selected in the final active dictionary-query space, rather than from post-routing fused features, and selection is not restricted to correctly classified samples. Candidate admission uses a confidence threshold of 0.75, a minimum normalized novelty of 0.05, and a maximum cross-class cosine similarity of 0.75. Candidate quality combines confidence, novelty, and cross-class separation with weights $(0.60, 0.20, 0.20)$, and greedy selection uses a diversity weight of 0.25. The loss-adaptive replacement fraction is computed using $\alpha = 0.5$ and $\beta = 0.10$, then clipped to $[0.05, 0.15]$. Replaceable slots are ranked using low usage, within-class redundancy, and cross-class conflict with weights $(0.45, 0.35, 0.20)$, respectively. Selected slots are directly replaced by normalized candidate vectors, and the usage statistics of replaced slots are reset. Candidate collection, usage accumulation, dictionary initialization, and slot replacement are disabled during validation and testing.

Global configuration and tuning protocol. All persistent-dictionary and LAPE hyperparameters in Table 6 are fixed globally and shared across all datasets; we do not perform dataset-specific tuning of the admission thresholds, candidate-quality weights, plasticity coefficients, replacement bounds, or slot-priority weights. These parameters are designed as constrained maintenance rules rather than unconstrained optimization knobs: the admission criteria operate on bounded confidence or cosine-similarity quantities, the candidate and replacement weights determine only relative rankings, and class-wise plasticity is explicitly clipped to the narrow replacement range ([0.05,0.15]). Likewise, the structural choices (r), (q), (M), (E_{warm}), and ($\lambda_{\max}$) use a single global configuration in the main experiments. Appendix C.9 further shows that performance remains consistently strong over broad ranges of these choices, with most tested configurations remaining competitive with or surpassing the strongest baselines. This indicates that the gains of ProtoRoute do not depend on a narrowly tuned or dataset-specific operating point.

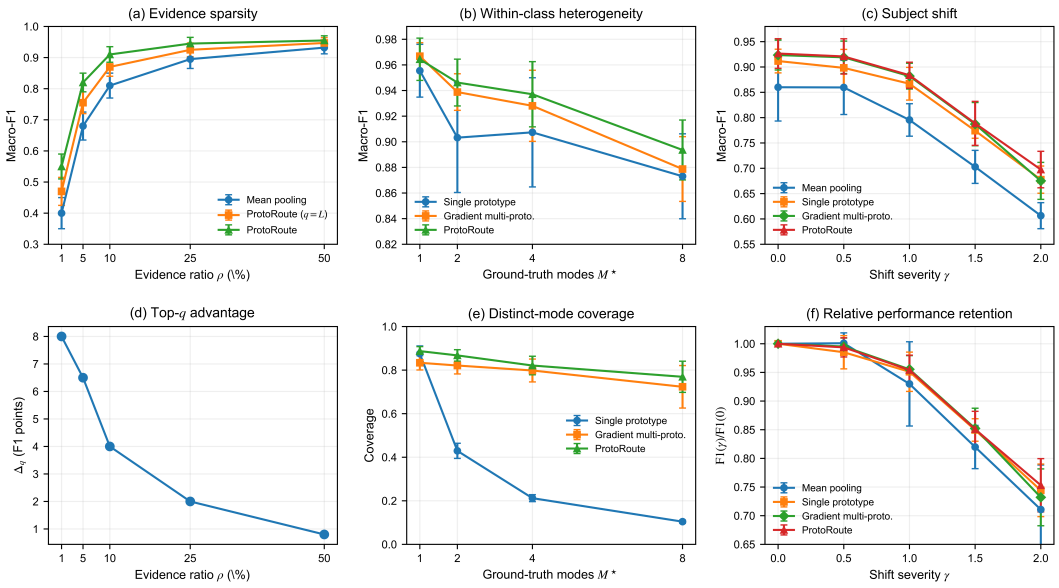

Figure 2: **Controlled analysis of when persistent evidence is useful.** The top row reports predictive performance, while the bottom row exposes the corresponding mechanism. **(a,d) Evidence sparsity:** the performance gap between ProtoRoute and dense token aggregation ($q = L$) is largest when evidence is highly localized and contracts as ρ increases, indicating that Top-q primarily mitigates evidence dilution in sparse regimes. **(b,e) Within-class heterogeneity:** F1 differences remain moderate, whereas distinct-mode coverage clearly separates the methods; a single prototype rapidly loses coverage as $M^\star$ grows, while ProtoRoute preserves multiple latent modes. **(c,f) Subject shift:** prototype-based variants maintain substantially higher absolute F1 than mean pooling as γ increases, although normalized retention among prototype-based methods is similar. Error bars denote standard deviation over five seeds.

C Complete Results

C.1 Controlled Synthetic Study: When Does Persistent Evidence Help?

Real medical datasets do not provide the exact support of discriminative evidence, ground-truth within-class modes, or a continuously controllable subject-shift severity. We therefore construct a subject-disjoint synthetic benchmark in which these factors are explicitly controlled. Study A varies evidence sparsity, Study B varies within-class heterogeneity, and Study C varies test-time subject shift. Figure 2 summarizes the performance trends and mechanism-specific diagnostics, while Table 7 reports the corresponding numerical results.

Study A: Sparse evidence. Figures 2 (a,d) show that the benefit of Top-q aggregation is concentrated in the sparse-evidence regime. While all methods improve as the discriminative interval becomes denser, the gap between ProtoRoute and dense token aggregation ($q = L$) contracts from 8.0 F1 points at $\rho = 1\%$ to 0.8 points at $\rho = 50\%$. The trend in Figure 2 (d) is more informative than absolute F1 alone: it is consistent with Top-q suppressing dilution from irrelevant tokens when class evidence is localized.

Study B: Within-class heterogeneity. Figures 2 (b,e) examine whether persistent multi-prototype maintenance becomes useful when a class contains multiple distinct evidence modes. When $M^\star = 1$, where no within-class heterogeneity is present, the prototype-based methods achieve similar F1, with Grad.-Multi slightly outperforming ProtoRoute (96.68% vs. 96.44%). Once multiple modes are introduced, however, ProtoRoute consistently outperforms the capacity-matched gradient multi-prototype variant, achieving 94.62% vs. 93.88% at $M^\star = 2$, 93.71% vs. 92.81% at $M^\star = 4$, and 89.36% vs. 87.87% at $M^\star = 8$.

Table 7: **Controlled synthetic analysis of persistent evidence.** Study A varies evidence sparsity ρ, Study B varies the number of ground-truth within-class modes $M^\star$, and Study C varies subject-shift severity γ. Δ_q is the F1 improvement of Top-q aggregation over dense token aggregation ($q = L$). Distinct-mode coverage evaluates one-to-one correspondence between ground-truth modes and learned prototype slots. Results are mean±std over five seeds.

(A) Evidence Sparsity

ρ (%)	Mean Pooling F1	$q = L$ F1	ProtoRoute F1	Δ_q (pp)	LocGain
1	40.00±5.00	47.00±4.50	**55.00±4.00**	**+8.0**	0.060±0.015
5	68.00±4.50	75.50±3.50	**82.00±3.00**	**+6.5**	0.110±0.020
10	81.00±4.00	87.00±3.00	**91.00±2.50**	**+4.0**	0.145±0.025
25	89.50±3.00	92.50±2.20	**94.50±2.00**	**+2.0**	0.130±0.025
50	93.20±2.00	94.70±1.70	**95.50±1.50**	**+0.8**	0.105±0.020

(B) Within-Class Heterogeneity

$M^\star$	Single F1	Grad.-Multi F1	ProtoRoute F1	Single Cov.	Grad.-Multi Cov.	ProtoRoute Cov.
1	95.55±2.06	**96.68±1.05**	96.44±1.65	0.875±0.036	0.834±0.033	**0.887±0.021**
2	90.32±4.28	93.88±1.43	**94.62±1.83**	0.429±0.035	0.821±0.039	**0.868±0.026**
4	90.74±4.26	92.81±2.78	**93.71±2.54**	0.212±0.016	0.798±0.053	**0.821±0.043**
8	87.31±3.31	87.87±2.53	**89.36±2.34**	0.104±0.007	0.723±0.098	**0.769±0.072**

(C) Subject Shift

γ	Mean F1	Single F1	Grad.-Multi F1	ProtoRoute F1	ProtoRoute Retention
0.0	86.00±6.67	91.17±2.34	92.35±2.95	**92.66±2.92**	1.000±0.000
0.5	85.96±5.34	89.83±3.64	91.91±3.20	**92.08±3.48**	0.994±0.017
1.0	79.54±3.21	86.69±3.23	88.20±2.54	**88.38±2.57**	0.954±0.026
1.5	70.28±3.26	77.43±1.50	78.72±4.27	**78.87±4.34**	0.851±0.031
2.0	60.65±2.57	67.76±2.69	67.52±3.65	**69.75±3.60**	0.753±0.046

The mechanism-specific distinction is even clearer in distinct-mode coverage. ProtoRoute achieves the highest coverage at every $M^\star$, and at $M^\star = 8$ retains a coverage of 0.769, compared with 0.723 for gradient multi-prototypes and 0.104 for a single prototype. Importantly, the advantage over Grad.-Multi is obtained under matched multi-prototype capacity, indicating that the improvement is not explained simply by maintaining more prototype slots. These results support the role of persistent multi-prototype evolution in preserving heterogeneous within-class evidence modes as class structure becomes more complex.

Study C: Subject shift. Figures 2 (c,f) distinguish absolute robustness from relative degradation. As shift severity increases, all methods deteriorate, while the prototype-based variants consistently remain above the no-persistence mean-pooling baseline. ProtoRoute decreases from 92.66% at $\gamma = 0$ to 69.75% at $\gamma = 2$, compared with $86.00\% \rightarrow 60.65\%$ for mean pooling, $91.17\% \rightarrow 67.76\%$ for the single-prototype variant, and $92.35\% \rightarrow 67.52\%$ for the gradient multi-prototype variant. The prototype-based methods remain close under mild and moderate shifts, but a clearer separation emerges under the strongest shift: at $\gamma = 2$, ProtoRoute exceeds the single-prototype and gradient multi-prototype variants by 1.99 and 2.23 F1 points, respectively. Its relative retention is also highest at this severity (0.753), compared with approximately 0.743 and 0.731 for the single- and gradient multi-prototype variants. These results suggest that ordinary class prototypes already provide a useful reference under moderate subject variation, while the proposed persistent multi-prototype evolution provides additional robustness when the subject shift becomes severe.

Together, the controlled studies identify complementary operating regimes of the persistent pathway: sparse token aggregation is most useful when discriminative evidence is localized, multiple persistent prototypes become important as within-class structure grows more heterogeneous, and class-level prototype evidence provides a stable reference under increasing subject shift.

Synthetic Generator Details. The benchmark contains four classes, eight channels, and sequences of length $T = 256$, with disjoint training, validation, and test subjects. Each sample consists of colored background noise, a localized class- and mode-dependent discriminative event, wrong-class distractors outside the evidence interval, and subject-specific nuisance transformations including amplitude variation, baseline and slope drift, temporal displacement, phase distortion, and

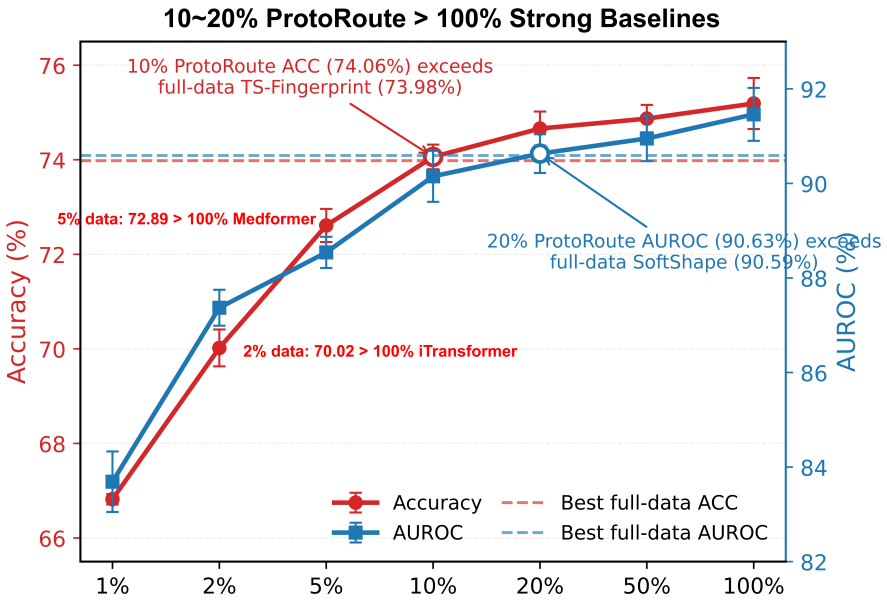

Figure 3: **Training-data efficiency on PTB-XL.**

additional noise. The generator records the exact evidence mask, latent mode identity, and subject identity for every sample.

Evidence sparsity: Study A varies only the support ratio ρ of the true discriminative event. To reduce the confound between evidence sparsity and total signal energy, its amplitude is partially compensated as:

$$a_{\text{eff}}(\rho) = a_0 \,\text{clip}\left[\left(\frac{w_{0.1}}{w_\rho}\right)^{0.35}, 0.55, 2.15\right], \tag{27}$$

while distractor support is sampled independently of ρ.

Within-class heterogeneity: Study B uses a fixed bank of eight canonical modes and activates nested subsets,

$$\mathcal{M}_1 = \{0\}, \quad \mathcal{M}_2 = \{0, 4\}, \quad \mathcal{M}_4 = \{0, 2, 4, 6\}, \quad \mathcal{M}_8 = \{0, \ldots, 7\}, \tag{28}$$

so changing $M^\star$ modifies only the number of active modes without redrawing their underlying geometry.

Subject shift: Study C trains and validates at $\gamma = 0$ and evaluates the same checkpoint at $\gamma \in \{0, 0.5, 1, 1.5, 2\}$. Increasing γ jointly scales predefined subject-level amplitude, drift, temporal, phase, and noise perturbations, thereby isolating test-time distribution shift from changes in optimization or sample composition.

C.2 Training-Data Efficiency

Large-scale physiological datasets often contain substantial redundancy, raising the question of whether strong generalization requires using the entire training set. We evaluate ProtoRoute on the PTB-XL using only $\{1, 5, 10, 20, 50, 100\}\%$ of the training samples, while keeping the subject-independent validation and test sets unchanged. As shown in Figure 3, ProtoRoute retains strong performance even in the low-data regime. Remarkably, with only 10-20% of the training samples, it already matches or surpasses the best **full-data baseline**. Performance saturates rapidly thereafter and reduces end-to-end training time from **3 hours** to **20 minutes** under the same hardware and training protocol. This result is consistent with the role of the persistent dictionary: rather than repeatedly encoding redundant sample-specific variations, the model accumulates reusable class-conditioned evidence across samples, indicating that ProtoRoute can extract transferable class evidence from substantially fewer training observations.

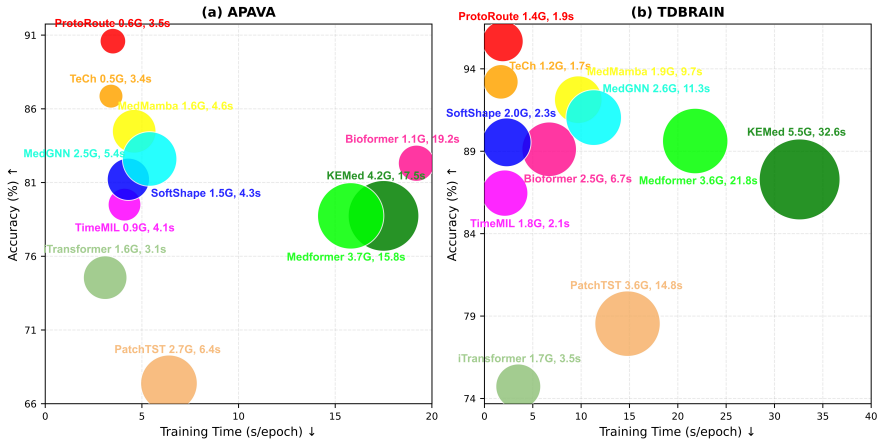

Figure 4: **Accuracy–efficiency trade-off.** The x-axis denotes training time per epoch, the y-axis denotes Accuracy, and bubble size represents peak GPU memory. ProtoRoute consistently occupies the high-accuracy, low-cost region, achieving the highest Accuracy with modest training time and memory overhead.

C.3 Efficiency Analysis

We follow the efficiency evaluation setting of MedMamba and compare against the efficiency measurements reported therein. We profile ProtoRoute under the same hardware setting and report its training time per epoch and peak GPU memory. Multi-GPU training is implemented with DistributedDataParallel; for ProtoRoute, token-level evidence is aggregated across workers before a single synchronized epoch-level memory update. The reported training time includes this synchronization and memory-maintenance overhead.

Figure 4 compare accuracy against training time per epoch and peak GPU memory on APAVA and TDBRAIN, respectively. ProtoRoute consistently lies in the high-accuracy, low-cost region. Compared with MedMamba, ProtoRoute reduces training time by about **24%/80%** and peak memory by **63%/26%** on APAVA/TDBRAIN, respectively, while achieving higher Accuracy. Though TeCh is marginally cheaper, ProtoRoute provides a substantially better accuracy–efficiency trade-off with only a small additional cost. These results indicate that persistent evidence learning introduces little computational burden: the non-parametric dictionary evolves outside backpropagation, while the prototype state itself is small relative to the parametric encoder.

C.4 Robustness Analysis

Robustness to Non-Stationary Drift. We inject a per-channel linear baseline drift into each test sample as $X'_{t,c} = X_{t,c} + s\, a_c(\frac{t}{T} - \frac{1}{2})$, where $s \in \{0, 0.25, 0.5, 0.75, 1.0\}$ and $a_c \sim \mathcal{U}(-\sigma_c, \sigma_c)$ is sampled independently for each channel, with σ_c the within-sample standard deviation of channel c. As shown in Figure 5, full ProtoRoute is remarkably stable under increasing baseline drift: from $s = 0$ to $s = 1$, F1 decreases by only **0.46** percentage points (89.96→89.50), while AUROC decreases by merely **0.08** points (96.94→96.86). In contrast, removing persistent evidence routing leads to substantially larger degradation, with F1 dropping by **2.30** points (86.08→83.78) and AUROC by **1.55** points (95.17→93.62).

The effect becomes more pronounced as the distribution shift strengthens. The F1 gap between full ProtoRoute and its w/o-PER counterpart increases from 3.88 points at $s = 0$ to 5.72 points at $s = 1$, while the corresponding AUROC gap increases from 1.77 to 3.24 points. This widening separation indicates that the persistent pathway provides increasing benefit as low-frequency baseline corruption becomes more severe. Notably, the w/o-PER model still degrades more gracefully than MedMamba and MedGNN, suggesting that robustness is not attributable to the persistent pathway

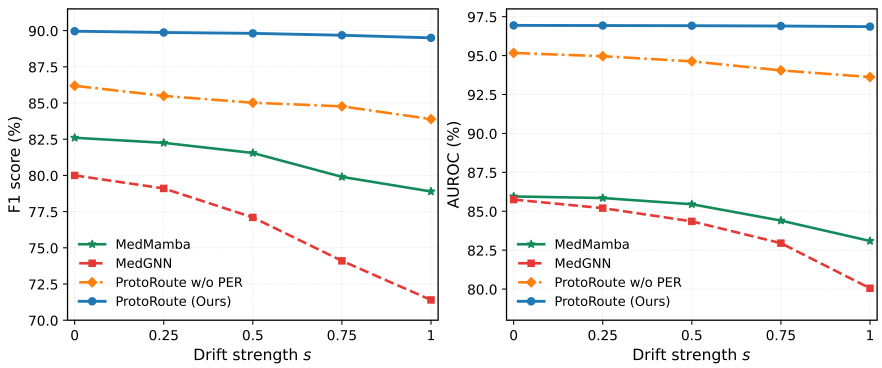

Figure 5: **Robustness to baseline drift on APAVA under the SI setting.** We inject a controlled per-channel linear baseline drift with strengths $s \in \{0, 0.25, 0.5, 0.75, 1.0\}$ at test time. Left: F1 score. Right: AUROC. In addition to MedMamba and MedGNN, we report ProtoRoute without persistent evidence routing (w/o PER) to isolate the contribution of the persistent pathway. Full ProtoRoute remains nearly invariant as drift increases, whereas removing PER produces a noticeably steeper degradation. The widening gap between the full and w/o-PER variants under stronger drift supports the role of persistent class evidence in improving robustness to low-frequency distribution shifts.

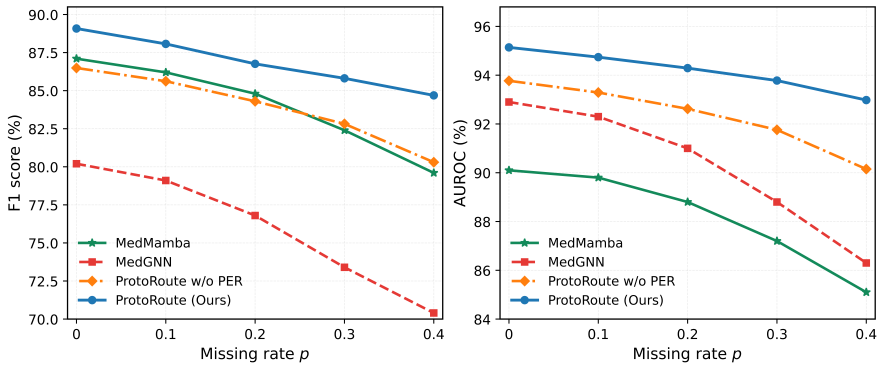

Figure 6: **Robustness to missing channels on PTB under the subject-independent setting.** For each test sample, we randomly mask $\lfloor pC \rfloor$ channels without replacement and zero their complete trajectories, with missing rates $p \in \{0, 0.1, 0.2, 0.3, 0.4\}$. We compare ProtoRoute with MedMamba, MedGNN, and ProtoRoute without persistent evidence routing (w/o PER) to isolate the contribution of the persistent pathway. Left: F1 score; right: AUROC. Full ProtoRoute consistently achieves the strongest performance and exhibits the most graceful degradation as channel loss increases, while removing PER leads to noticeably steeper performance decay. This widening gap between the full and w/o-PER variants indicates that persistent class evidence improves robustness to incomplete sensor availability.

alone. Rather, the parametric backbone already provides a degree of resistance to baseline variation, while persistent evidence routing contributes an additional and substantial stabilizing effect. These results support the interpretation that slowly evolving class evidence provides a stable cross-sample reference when the instantaneous input representation is perturbed by non-stationary baseline shifts.

Robustness to Missing Channels. We evaluate sensor-dropout robustness by randomly masking channels at test time. For each test sample, we uniformly sample $\lfloor pC \rfloor$ channels without replacement and zero their trajectories ($X'_{:,c} = 0$), where $p \in \{0, 0.1, 0.2, 0.3, 0.4\}$ denotes the missing rate. As shown in Figure 6, all methods degrade as channel availability decreases, but full ProtoRoute remains consistently more stable. From $p = 0$ to $p = 0.4$, ProtoRoute drops by only **4.39** F1 points (89.08→84.69) and **2.16** AUROC points (95.14→92.98), indicating strong robustness under substantial sensor loss.

Adding the w/o-PER variant further clarifies the source of this robustness. When persistent evidence routing is removed, performance deteriorates more noticeably, with F1 decreasing by **6.19** points (86.49→80.30) and AUROC by **3.62** points (93.77→90.15). Moreover, the gap between full ProtoRoute and its w/o-PER counterpart widens as the missing rate increases: the F1 gap grows from **2.59** points at $p = 0$ to **4.39** points at $p = 0.4$, while the AUROC gap grows from **1.37** to **2.83** points.

These results suggest that robustness to channel loss is not due solely to the backbone architecture. Rather, persistent evidence routing provides an additional stabilizing effect by preserving reusable class-conditioned evidence even when part of the sensor set is unavailable

C.5 Detailed Ablation Analysis

C.5.1 Architecture Components

Table 2 evaluates four groups of controlled ablations.

Architecture Components. **w/o PER** removes persistent evidence retrieval and prototype-derived class evidence, leaving only the parametric pathway; it produces the most consistent degradation across datasets, supporting the complementary role of persistent cross-sample class evidence. **w/o NFR** disables near/far token refinement, while **w/o MVT** retains only the temporal view and removes the complementary frequency and channel tokens. Finally, **K-means Initialization** replaces the default post-warm-up bootstrap initialization with class-wise K-means while retaining subsequent LAPE updates. The overall results indicate that PER provides the dominant architectural gain, while multi-view modeling, NFR, and spectral information provide complementary improvements; K-means remains competitive but does not explain the benefit of persistent evidence learning.

Hierarchical Routing and Sparse Evidence Aggregation. This isolates the two levels of sparsification in PER. **Dense Prototype** (r=M) aggregates all prototype slots of a class instead of the Top-r most relevant ones, whereas **Dense Tokens** (q=L) averages class evidence over all tokens instead of the Top-q tokens. **Flat Routing** removes the hierarchical sparse aggregation structure. **Proto-logit-only** retains only explicit prototype-derived logits, while **Retrieval-only** retains only the feature-level retrieval residual. **Single Prototype** restricts each class to one prototype, and **Pooled Update** updates the dictionary using pooled sample representations. All variants are generally inferior to the full design in F1, indicating that multi-prototype modeling, sparse evidence aggregation, and token-level memory updates are jointly useful. The two one-sided routing variants further suggest that retrieval and prototype logits provide complementary information.

LAPE Memory Maintenance. We further ablate the decisions involved in persistent dictionary maintenance. **Fixed Budget** uses a constant replacement ratio instead of the loss-adaptive class-wise budget; **No Admission** removes reliability, novelty, and inter-class compatibility constraints; **No Diversity** removes the diversity bonus in candidate selection; **Random Replacement** randomly evicts existing slots; **Usage-only Replacement** performs eviction using usage alone; and **EMA Update** replaces hard slot replacement with exponential moving-average updates. Random replacement causes the most pronounced and consistent degradation in this group, showing that memory quality depends not only on admitting new evidence but also on selecting which existing evidence should be removed. The remaining variants yield smaller but systematic losses, supporting the combined use of adaptive budgeting, candidate filtering, diversity, and structured eviction.

Prototype Regularization. We finally remove the prototype contrastive objective $\mathcal{L}_{\text{proto}}$, the sample-to-prototype margin objective $\mathcal{L}_{\text{margin}}$, or both. Each removal leads to a modest degradation, with the joint removal being generally more detrimental than removing either term alone. These results suggest that the two objectives provide complementary regularization, while their relatively smaller effect compared with removing PER or altering the routing mechanism indicates that the main gains do not arise solely from auxiliary losses.

Table 8: **Persistent-state dynamics under different update regimes.** Path drift characterizes prototype-state displacement, cumulative turnover summarizes normalized slot replacement during training, and validation volatility measures consecutive-epoch fluctuations in validation F1 after warm-up. Drift and turnover are diagnostic measures of state plasticity, not quantities that should necessarily be minimized. "–" denotes not applicable.

	APAVA			BCI-2a			PTB		
Variant	**Drift**	**Turnover**	**Vol. ↓**	**Drift**	**Turnover**	**Vol. ↓**	**Drift**	**Turnover**	**Vol. ↓**
Parametric only	–	–	0.013	–	–	0.017	–	–	0.026
Gradient prototypes	–	–	0.013	–	–	0.018	–	–	0.027
Batch-wise non-parametric	0.048	0.097	0.016	0.059	0.112	0.019	0.099	0.206	0.027
Epoch-wise LAPE	0.028	0.055	0.015	0.042	0.076	0.016	0.051	0.083	0.024
5-epoch LAPE	0.016	0.018	0.011	0.008	0.014	0.016	0.019	0.018	0.025
Frozen dictionary	0.000	0.000	0.009	0.000	0.000	0.015	0.000	0.000	0.024

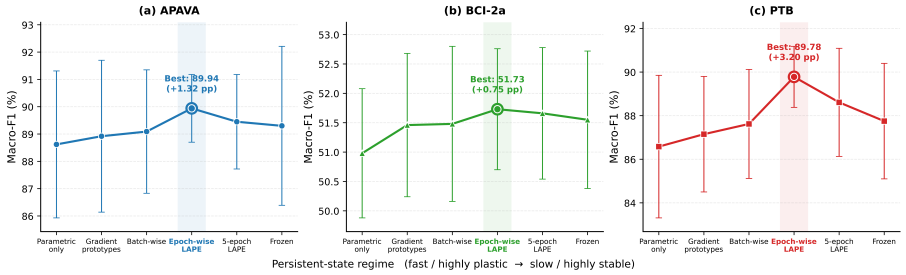

Figure 7: **Effect of persistent-state update regime on macro-F1.** Results are mean±standard deviation over five seeds. Across APAVA, BCI-2a, and PTB, epoch-wise LAPE consistently attains the highest mean F1, revealing an intermediate-timescale optimum: gradient- and batch-wise updates are overly sensitive to transient evidence, whereas less frequent or frozen updates sacrifice adaptivity.

C.5.2 Persistent-State Dynamics

Dynamics metrics. To complement final predictive performance, we quantify how the persistent state evolves during training using three dynamics measures.

Dictionary drift: Dictionary drift measures the epoch-to-epoch displacement of the prototype state:

$$D_e = \frac{1}{KM} \sum_{k=1}^{K} \sum_{m=1}^{M} \left[1 - \cos\left(b_{k,m}^{(e)}, b_{k,m}^{(e-1)} \right) \right]. \tag{29}$$

Larger values indicate greater movement of the persistent class state.

Slot turnover: For non-parametric variants, we define slot turnover as the fraction of dictionary slots replaced during epoch e:

$$T_e = \frac{|\mathcal{R}_e|}{KM}, \tag{30}$$

where $\mathcal{R}_e$ denotes the set of replaced slots.

Validation volatility: We measure predictive instability after the warm-up period using the average absolute change in validation F1 between consecutive epochs:

$$V = \frac{1}{E' - 1} \sum_{e=2}^{E'} \left| \mathrm{F1}_e^{\mathrm{val}} - \mathrm{F1}_{e-1}^{\mathrm{val}} \right|. \tag{31}$$

Lower volatility indicates more stable validation behavior.

Results and Analysis. Figure 7 reveals a consistent intermediate-timescale optimum across APAVA, BCI-2a, and PTB. Epoch-wise LAPE obtains the highest mean F1 on all three datasets, improving over the parametric-only model by 1.32, 0.75, and 3.20 percentage points, respectively.

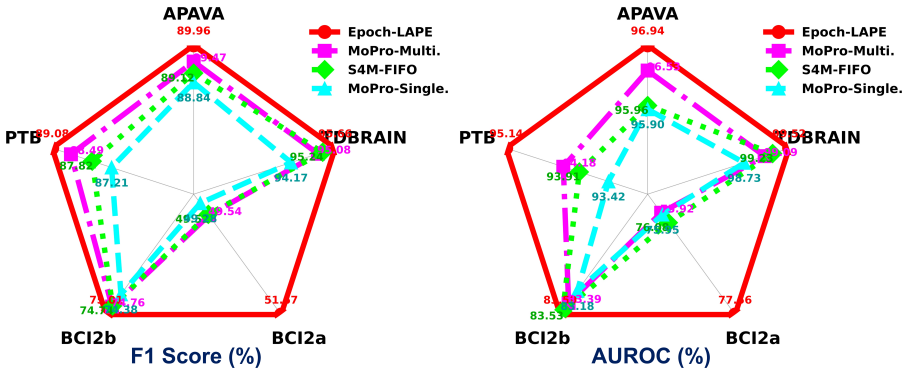

Figure 8: **Comparison of alternative persistent-state evolution strategies.** We compare epoch-wise LAPE with a class-wise single-prototype momentum update (*MoPro-Single*), a capacity-matched multi-slot online momentum update (*MoPro-Multi*), and an S4M-inspired online FIFO bank (*S4M-FIFO*) on five subject-independent benchmarks. Left: macro-F1; right: AUROC. Epoch-wise LAPE achieves the highest mean F1 and AUROC on all five datasets. The multi-slot momentum and FIFO controls retain the same token-level persistent-evidence representation and PER reading mechanism, isolating the effect of how the persistent state is evolved. The results show that neither momentum smoothing, multi-slot capacity, nor online dynamic-bank maintenance alone accounts for the benefit of the proposed class-adaptive epoch-level evolution.

It also outperforms batch-wise non-parametric updating by 0.85, 0.25, and 2.16 points. Averaged across datasets, epoch-wise LAPE improves F1 by 1.76 points over parametric-only learning and by 1.09 points over batch-wise updating.

The state dynamics in Table 8 explain this pattern. Batch-wise updating produces the largest path drift and turnover on every dataset, together with the highest or tied-highest validation volatility. This indicates that updating the dictionary from each mini-batch makes the persistent state overly sensitive to transient evidence. In contrast, five-epoch updates and frozen dictionaries substantially reduce state movement and replacement, but their improved stability is accompanied by lower predictive performance.

Epoch-wise LAPE maintains an intermediate degree of plasticity: it evolves sufficiently often to track the changing encoder representation, while aggregating evidence over a longer horizon than individual mini-batches. This balance is particularly evident on PTB, where epoch-wise LAPE exceeds batch-wise updating and the frozen dictionary by 2.16 and 2.03 F1 points, respectively. The smaller separation between epoch-wise and five-epoch LAPE on BCI-2a suggests that the precise optimum may depend on dataset dynamics; nevertheless, the same mean-performance peak at the epoch timescale is observed across all three benchmarks. Overall, these results support the central dual-timescale hypothesis: persistent class evidence should evolve ***slowly, but remain adaptive***.

C.5.3 Alternative Persistent-State Evolution Strategies

The preceding analysis studies *when* persistent class evidence should be updated. We next examine a complementary question: whether the benefit of LAPE can be explained by existing prototype-evolution principles. Prior work: PCL(Li et al., 2020) alternates representation learning with clustering-based prototype estimation, MoPro(Li et al., 2021) maintains class-wise momentum prototypes, and S4M(Peng et al., 2025) employs a dynamically written prototype bank for time-series modeling. So, we explore whether the proposed *class-adaptive, epoch-level evolution of token-level class evidence* provides additional benefit beyond these alternatives.

To isolate the evolution mechanism, we construct three prior-inspired controls. **MoPro-Single** maintains one normalized momentum prototype per class. **MoPro-Multi** extends the same online momentum rule to M class-specific slots, thereby controlling for multi-prototype capacity. **S4M-FIFO** replaces LAPE with an S4M-inspired online similarity-based admission and FIFO maintenance strategy. For the multi-slot controls, the parametric backbone, token representation space, dictionary capacity, and PER read mechanism are kept unchanged; the principal difference is how the persis-

tent state is evolved. In contrast, epoch-wise LAPE keeps the dictionary fixed throughout mini-batch optimization, accumulates candidate evidence over the epoch, and performs a single structured state transition using reliability-aware admission, class-dependent plasticity, diversity-aware selection, and usage–redundancy–conflict-based replacement.

As shown in Figure 8, epoch-wise LAPE achieves the highest mean F1 and AUROC in all ten dataset–metric comparisons. Compared with the capacity-matched MoPro-Multi control, LAPE improves F1 by 0.49, 0.58, 2.13, 0.25, and 0.59 percentage points on APAVA, TDBRAIN, BCI2a, BCI2b, and PTB, respectively, corresponding to an average gain of 0.81 points. Compared with S4M-FIFO, the corresponding F1 improvements are 0.84, 0.42, 2.15, 0.27, and 1.26 points, averaging 0.99 points. The separation is particularly pronounced on BCI2a, where MoPro-Multi and S4M-FIFO obtain F1 scores of 49.54 and 49.52, respectively, compared with 51.67 for LAPE. Meanwhile, MoPro-Multi generally improves over MoPro-Single, supporting the benefit of retaining multiple within-class evidence modes rather than compressing each class into a single representative.

These controlled comparisons show that the advantage of (*i*) ProtoRoute cannot be explained solely by maintaining an external prototype state, (*ii*) increasing the number of prototype slots, or (*iii*) applying momentum-based slow updates. Likewise, an online dynamically maintained FIFO bank does not recover the performance of LAPE. These results sharpen the role of the proposed dual-timescale formulation: persistent class evidence is represented as a multi-slot token-level state, kept fixed over the fast mini-batch optimization horizon, and evolved from evidence accumulated over a longer horizon through a structured, class-adaptive state transition.

C.5.4 Backbone Portability of Persistent Evidence Learning

We examine whether persistent evidence learning is specific to the parametric backbone used by ProtoRoute or remains beneficial across different representation architectures. For each backbone, we preserve its native tokenization, encoder, and prediction head, and attach the same persistent routing and LAPE mechanism *only* to its final token representation. We therefore compare each backbone against its paired persistent counterpart, measuring $\Delta_{\text{persistent}} = \text{F1}_{\text{backbone+persistent}} - \text{F1}_{\text{backbone}}$.

The results in Table 9 demonstrate that persistent evidence learning transfers consistently across substantially different backbone architectures. Across the 12 backbone–dataset pairs, adding the persistent pathway improves F1 in all 12 cases, with an average gain of $+4.63$ pp, while Accuracy increases in all 12 comparisons. The gains are particularly pronounced for generic Transformer backbones. PatchTST improves by $+20.34$, $+12.67$, and $+4.83$ pp on APAVA, TDBRAIN, and PTB, respectively, while iTransformer gains $+2.21$, $+6.79$, and $+1.21$ pp on the same datasets. Importantly, the benefit remains evident on stronger medical time-series architectures: Medformer gains $+2.58$, $+1.56$, and $+1.23$ pp, and TeCh yields changes of $+0.21$, $+0.70$, and $+1.18$ pp, respectively. These results show that persistent class evidence can be integrated into diverse representation spaces without redesigning the native tokenizer, encoder, or prediction head. The large improvements on generic backbones, together with the continued gains on already strong domain-specific models, support persistent evidence learning as a broadly applicable ***complementary*** mechanism for strengthening class-discriminative representations across architectures.

C.6 Subject-Level Cross-Validation

To further reduce the potential bias introduced by a single fixed subject-independent split, we conduct subject-level cross-validation across all seven benchmarks. For the five clinical-diagnosis datasets—APAVA, ADFTD, TDBRAIN, PTB, and PTB-XL—we follow the subject-level five-fold cross-validation protocol of TeCh (ICLR'2026) (Guoqi et al., 2026), where all windows from the same subject are kept within a single split and the held-out test subjects vary across folds. For BCI-2a and BCI-2b, each of which contains nine subjects, we adopt nine-fold leave-one-subject-out (LOSO) cross-validation, such that each subject serves as the unseen test subject exactly once. In all cases, the same subject partitions or held-out subjects are used for all compared methods. We include three strong recent baselines, Medformer, MedMamba, and TeCh, while each method retains its original preprocessing and training pipeline.

As shown in Figure 9, ProtoRoute achieves the highest mean Accuracy and macro-F1 in all 14 dataset–metric comparisons. On the five clinical-diagnosis benchmarks, ProtoRoute improves Ac-

Table 9: **Backbone portability of persistent evidence learning.** All metrics are shown in percentage (%), and ΔF1 denotes the absolute F1 change in percentage points (pp) relative to the paired base backbone. The better value within each Base/+Persistent pair is highlighted in bold.

Dataset	Backbone	Accuracy ↑		F1 ↑		ΔF1	AUROC ↑	
		Base	+Persistent	Base	+Persistent		Base	+Persistent
APAVA	PatchTST	67.38±1.83	**77.74±1.94**	56.81±3.60	**77.15±1.96**	**+20.34**	65.20±4.30	**87.52±1.36**
	iTransformer	74.55±1.66	**76.85±1.42**	72.30±1.79	**74.51±1.57**	**+2.21**	85.59±1.55	**86.31±1.40**
	TeCh	86.86±1.09	**87.00±0.85**	86.30±1.06	**86.51±1.09**	**+0.21**	94.02±0.52	**94.50±0.03**
	Medformer	78.74±0.64	**80.99±0.02**	76.31±0.71	**78.89±0.02**	**+2.58**	83.20±0.91	**89.32±0.03**
TDBRAIN	PatchTST	79.25±3.79	**91.92±2.23**	79.20±3.77	**91.87±2.27**	**+12.67**	87.95±4.96	**98.93±0.38**
	iTransformer	74.73±1.05	**81.56±1.64**	74.72±1.05	**81.51±1.68**	**+6.79**	83.38±1.14	**89.28±1.07**
	TeCh	93.21±0.61	**93.94±0.76**	93.20±0.61	**93.90±0.55**	**+0.70**	98.68±0.19	**98.74±0.06**
	Medformer	89.62±0.81	**91.27±0.45**	89.62±0.81	**91.18±0.47**	**+1.56**	96.41±0.33	**96.85±0.29**
PTB	PatchTST	74.74±1.62	**77.33±1.85**	64.36±3.38	**69.19±2.62**	**+4.83**	88.79±0.91	**90.05±2.78**
	iTransformer	83.89±0.71	**85.04±1.14**	79.06±1.06	**80.27±1.82**	**+1.21**	91.18±1.16	**93.64±1.29**
	TeCh	85.96±2.52	**87.26±1.41**	81.97±4.07	**83.15±2.11**	**+1.18**	94.57±0.70	**95.03±0.71**
	Medformer	83.50±2.01	**84.67±0.45**	79.18±3.31	**80.41±0.47**	**+1.23**	**92.81±1.48**	92.75±2.93

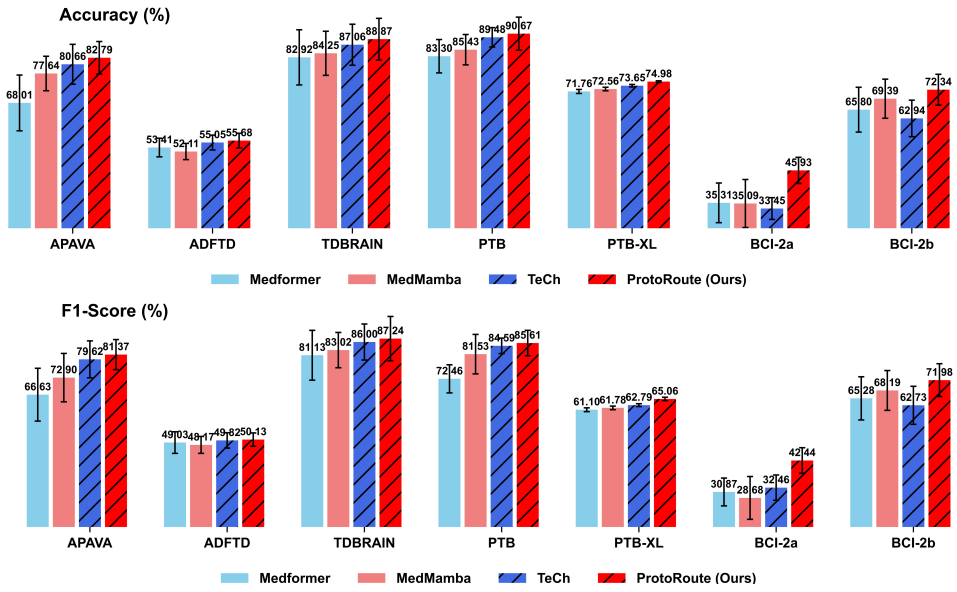

Figure 9: **Subject-level cross-validation across seven benchmarks.** Bars denote the mean performance across folds (or held-out subjects for LOSO), and error bars indicate the corresponding standard deviation.

curacy over the strongest competing method by 2.13, 0.63, 1.81, 1.19, and 1.33 percentage points on APAVA, ADFTD, TDBRAIN, PTB, and PTB-XL, respectively. The corresponding macro-F1 improvements are 1.75, 0.31, 1.24, 1.02, and 2.27 percentage points. This pattern is also observed under the nine-fold LOSO evaluation on the two neural-decoding benchmarks. On BCI-2a, ProtoRoute improves mean Accuracy and macro-F1 by 10.62 and 9.98 percentage points, respectively, while the corresponding gains on BCI-2b are 2.95 and 3.79 points.

These results provide additional evidence that the gains of ProtoRoute are not tied to a particular fixed subject partition. For the five clinical-diagnosis benchmarks, the held-out test subjects vary across folds while remaining strictly subject-disjoint from the training data. For BCI-2a and BCI-2b, nine-fold LOSO further evaluates each of the nine subjects exactly once as an unseen test subject. ProtoRoute retains the highest mean performance across these heterogeneous held-out subject cohorts, indicating that its subject-independent gains persist under varying subject partitions. Combined with the controlled ablations of the persistent pathway, these cross-validation results provide further empirical support for the role of persistent class-evidence learning in cross-subject generalization.

Table 10: **Subject-dependent classification results on the APAVA dataset.** Results are reported as mean ± standard deviation over five random seeds. The best result is highlighted in **red**, and the second-best in **blue**. All values are percentages, and the percentage symbol is omitted.

Dataset	Metric	Ours ProtoRoute	ICML'26 BioFormer	ICLR'26 TeCh	ICML'26 MedMamba	WWW'25 MedGNN	Neurips'24 Medformer	ICLR'24 iTransformer	ICLR'23 PatchTST
APAVA	Accuracy	**99.95**±0.07	**99.50**±0.14	99.07±0.13	98.66±0.16	98.15±0.25	98.26±0.23	93.20±1.06	97.55±0.26
	Precision	**99.95**±0.08	**99.52**±0.14	99.07±0.14	98.66±0.17	98.16±0.23	98.34±0.28	92.76±1.08	97.61±0.28
	Recall	**99.95**±0.06	**99.43**±0.16	99.09±0.12	98.68±0.15	98.18±0.26	98.02±0.21	93.18±1.18	97.28±0.29
	F1	**99.95**±0.07	**99.47**±0.15	99.08±0.13	98.67±0.16	98.16±0.25	98.18±0.24	92.93±1.11	97.44±0.27
	AUROC	**100.00**±0.00	**99.99**±0.01	99.91±0.05	99.88±0.02	99.80±0.05	99.81±0.03	98.11±0.42	99.67±0.04
	AUPRC	**100.00**±0.00	**99.99**±0.01	99.90±0.04	99.86±0.03	99.78±0.05	99.78±0.05	97.94±0.52	99.63±0.05

C.7 Subject-Dependent (SD) Evaluation

Although ProtoRoute is primarily designed for subject-independent generalization, we additionally evaluate it under the subject-dependent setting to examine whether persistent evidence learning compromises performance when subject overlap is permitted. For a stringent comparison, we include the strongest recent baselines with reported results under the same subject-dependent APAVA protocol. As shown in Table 10, ProtoRoute consistently achieves the best performance across all reported metrics, indicating that the proposed dual-timescale design does not rely exclusively on cross-subject distribution shifts.

C.8 Interpretability Analysis: Mechanistic Visualization of Evidence–Prototype Routing

Figure 10 provides a representative visualization of how sparse token-level evidence is routed to persistent prototype slots in PTB. For this correctly classified ECG example, the four selected Top-q tokens are evenly split between the frequency and channel views, with two tokens from each and no temporal token selected. Although $q = 4$ and $r = 2$ permit up to eight token–slot associations, the selected tokens collectively activate only five distinct prototype slots. This illustrates that different pieces of discriminative evidence can share persistent prototype support while still being associated with multiple compatible within-class evidence modes.

The activated prototype slots can further be traced back to similar training evidence in their corresponding native domains. Frequency-view prototypes $P35$ and $P23$ retrieve recurring spectral patterns from multiple training subjects, while the channel-view prototype $P6$ is associated with recurring channel-domain waveform evidence. Their Top-20 nearest-neighbor sets span 5, 7, and 5 of the 118 training subjects for $P35$, $P23$, and $P6$, respectively. These neighborhoods are therefore better interpreted as repeated multi-subject support rather than broad coverage of the training population. Importantly, the retrieved neighbors show that the activated prototype slots are not merely isolated latent vectors, but are associated with recurring class-consistent evidence observed across multiple training subjects.

This example illustrates the intended inference mechanism of ProtoRoute: sparse Top-q evidence selects a limited but non-singleton set of class-specific prototype slots, while the activated slots can be linked back to recurring training evidence. The visualization therefore provides a direct view of how the persistent dictionary participates in prediction by connecting sample-specific evidence with reusable class-conditioned prototype support.

C.9 Sensitivity Analysis

Sensitivity to sparse routing hyperparameters. As illustrated in Figure 11, strong performance is maintained over a broad range of sparse configurations rather than at a single isolated setting, indicating that ProtoRoute is not sensitive to a narrow choice of r and q across all datasets. In addition, Dense aggregation ($r = M$ or $q = L$, denoted by *all*) is generally lowest, supporting the role of sparse prototype and token selection in concentrating class evidence.

Sensitivity to prototype capacity. We further examine the number of persistent prototype slots per class, $M \in \{1, 8, 16, 32, 64, 128\}$, while fixing $q = 4$ and $r = 2$ ($r = 1$ when $M = 1$). Figure 12 shows that performance is not monotonic in dictionary size. Instead, the strongest region

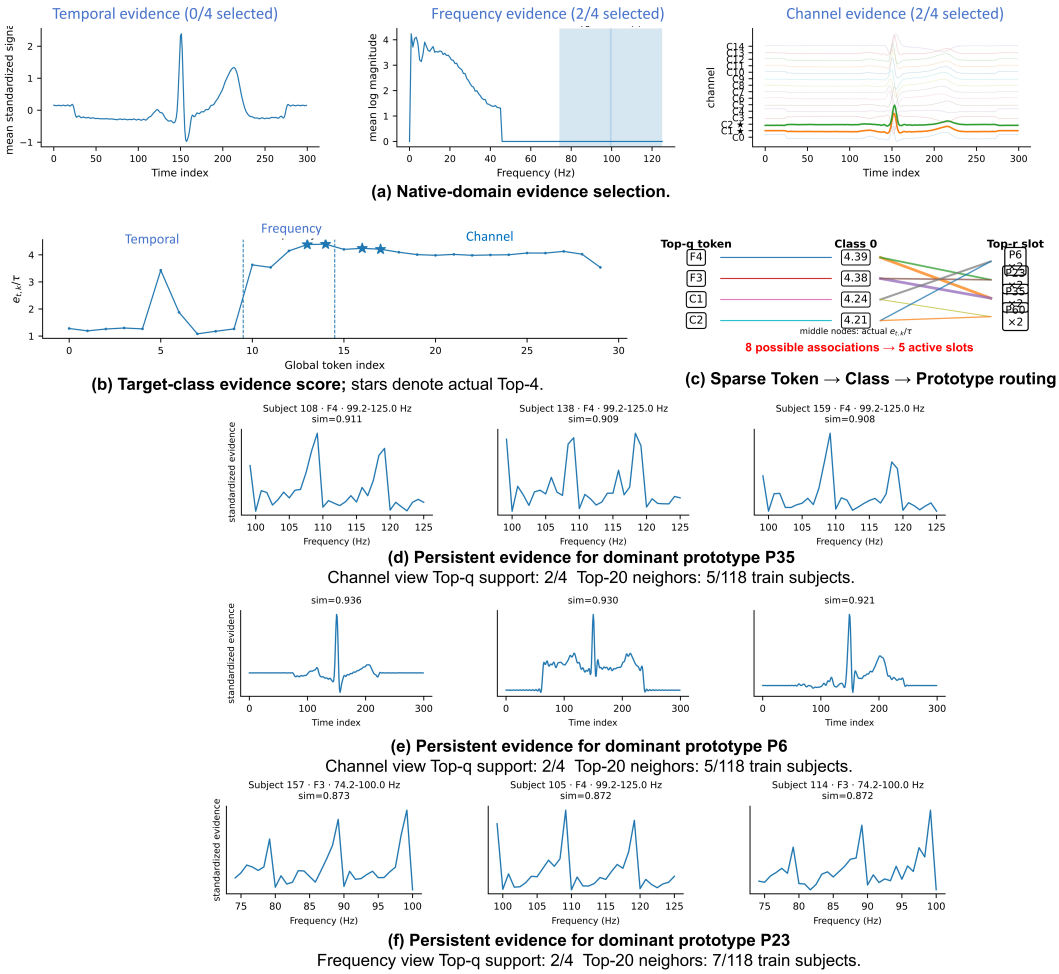

Figure 10: **Mechanistic visualization of evidence–prototype routing on PTB. (a)** Native-domain evidence selection. In contrast to the frequency-only BCI2a example, the four Top-q tokens are divided between the frequency and channel views (F:2, C:2), while no temporal token is selected. **(b)** Target-class evidence scores $e_{i,k}/\tau$ over the concatenated temporal, frequency, and channel token sequence; stars mark the selected Top-q evidence. **(c)** Sparse Token→Class→Prototype routing. Although $q = 4$ and $r = 2$ again permit eight token–slot associations, only five distinct prototype slots are activated, showing that multiple selected tokens share persistent class evidence rather than matching independent prototypes. **(d)** Same-view nearest frequency-domain evidence for prototype $P35$, which is supported by two of the four Top-q tokens; its Top-20 neighborhood spans 5 of 118 training subjects. **(e)** Same-view nearest channel-domain evidence for prototype $P6$, also supported by two Top-q tokens; its Top-20 neighborhood spans 5 of 118 training subjects. **(f)** Same-view nearest frequency-domain evidence for prototype $P23$, whose Top-20 neighborhood spans 7 of 118 training subjects. These examples demonstrate that the persistent dictionary can retrieve recurring evidence from multiple training subjects across different native evidence domains.

generally occurs at moderate capacities ($M = 32$–64). Importantly, increasing the dictionary to $M = 128$ does not improve performance on any of the four datasets. These results suggest that multiple persistent slots are useful for capturing heterogeneous class evidence, while excessive capacity provides diminishing returns and may introduce redundant or weakly supported slots. We therefore use $M = 64$ as a single global setting in the main experiments rather than tuning the memory capacity independently for each dataset. Its performance remains within the high-performing capacity regime across all four datasets, while avoiding dataset-specific hyperparameter selection.

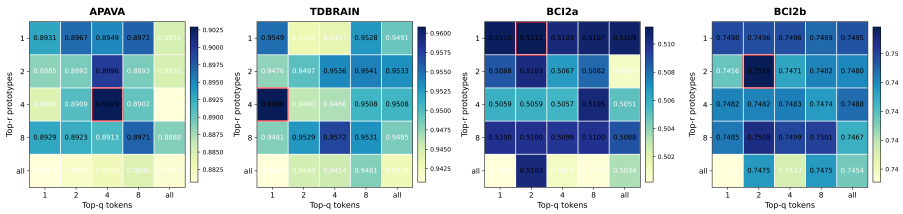

Figure 11: **Sensitivity of ProtoRoute to the sparse routing hyperparameters r and q.**

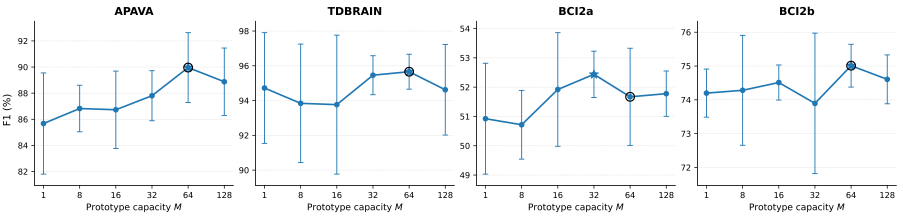

Figure 12: **Sensitivity to the persistent prototype capacity M.**

Warmup and persistent-evidence strength. We further examine the warmup duration E_{warm} and the maximum persistent-evidence strength $\lambda_{\max}$ in Figure 13. Across all datasets, a moderate warmup of $E_{\text{warm}} = 10$ consistently yields the strongest F1. Activating the persistent pathway immediately is inferior, particularly on APAVA and TDBRAIN, while delaying it to 20 epochs also degrades performance. This pattern is consistent with the dictionary becoming useful only after the parametric representation has acquired a sufficiently meaningful geometry.

Performance is comparatively stable across moderate values of $\lambda_{\max}$, but $\lambda_{\max} = 0.5$ attains the best F1 on all four datasets. A weaker persistent contribution can under-utilize cross-sample evidence, whereas increasing the strength to 1.0 provides no further benefit. Together, these results support the default configuration $E_{\text{warm}} = 10$ and $\lambda_{\max} = 0.5$ without indicating strong sensitivity to a narrowly tuned operating point.

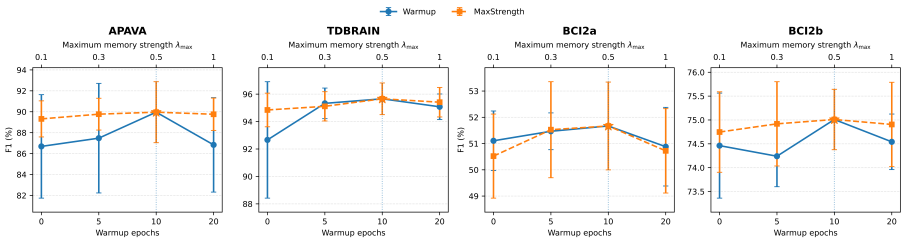

Figure 13: **Sensitivity to warmup duration and persistent-evidence strength.**