SUPPORT-INDUCED DIAGNOSTIC PROBING FOR MULTICHANNEL MEDICAL TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-channel medical time-series classification requires combining diagnostic context (usually given by labelled examples) with the information contained in a new recording. We propose Support-Induced Diagnostic Probing (SIDP), a class-conditioned retrieval architecture that converts positive-vs-negative support differences into diagnostic probes. Cross-channel attention first transforms a recording into a fixed-size set of latent representations. Support-generated probes then retrieve and score diagnosis-specific information from these representations. Once trained on a dataset, SIDP classifies new recordings from a small labelled support context without retraining or updating its parameters. We evaluate on PTB-XL, a large public ECG benchmark, where the task is to predict five diagnostic categories that can co-occur within a recording. Using the standard patient-disjoint test split, SIDP achieves **0.8450** macro-AUROC and **0.6215** macro-F1 across 2,158 test records from 1,877 patients. Our method improves macro-F1 over pooled-descriptor MLP and support-conditioned MLP baselines by **7.50** and **9.97** percentage points, respectively, with bootstrap confidence intervals that exclude zero. We further perform ablation experiments on the PTB dataset to isolate architectural contributions when identical channel descriptors are provided. Here, SIDP reaches **0.8603** window macro-F1 and **0.9661** patient macro-F1. Cross-channel attention improves window macro-F1 by **14.13** points over a pooling baseline with the same inputs, and support-generated probes improve macro-F1 by **2.14** points over fixed learned queries. An analysis of the attention weights suggests that both attention stages adapt to each recording. These results suggest that support-induced probing is an effective way to retrieve diagnostic information from multi-channel recordings. The source code for our architecture is available at https://anonymous.4open.science/r/sidp-85DD.

1 INTRODUCTION

Medical time-series classification involves two main challenges. The first is representing a multi-channel recording. The second is deciding how labelled examples should guide its interpretation. In addition, EEG and ECG datasets differ in channel count, sampling rate, diagnostic labels, and patient composition. Medical transformers (Wang et al., 2024a), reusable encoders (Wang et al., 2025), and foundational model re-purposing (Huang et al., 2026) partially addresses the first challenge. In this work, we focus on the second. In particular, we ask how a small set of labelled examples (the support set) should determine what information a classifier extracts from a new recording (the query).

Matching Networks (Vinyals et al., 2016) compare queries with individual support examples, while Prototypical Networks (Snell et al., 2017) summarize each class by the mean embedding of its support examples. Set attention (Lee et al., 2019) and conditional computation (Perez et al., 2018) provide learned pooling and context-dependent transformations. We propose Support-Induced Diagnostic Probing (SIDP), which separates the class description from the content of the query. First, cross-channel attention maps each recording to a fixed set of latent slots. The average positive and negative support representations then define a diagnostic contrast for each class. A shared network maps this contrast to an offset that is added to a set of learned probe seeds, producing class-specific probes. These probes attend to the query's latent slots to retrieve information. A shared scorer compares each probe with the retrieved information. Unlike centroid-based methods, the query is not

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.

pooled into a single vector before this class-specific retrieval. SIDP therefore extends centroid-based decisions and fixed class queries with support-dependent evidence extraction.

We evaluate SIDP on EEG (ADFTD (Miltiadous et al., 2023)) and ECG (PTB (Bousseljot, 2004) and PTB-XL (Wagner et al., 2020)) datasets. We train SIDP once per dataset. During inference, SIDP builds its class probes from the support set without any parameter updates. With per-channel signal descriptors as input, SIDP achieves **0.8603** window macro-F1 on PTB, and **0.9661** macro-F1 when predictions are aggregated per patient. Ablations on the PTB dataset, where every method receives the same descriptors, show that cross-channel attention, probe retrieval, and support-generated probes each improve over their controls. On PTB-XL, a recording can have several diagnoses at once (multi-label diagnosis). SIDP handles this by building a separate positive-vs-negative contrast for each diagnosis. It achieves **0.8450** macro-AUROC and **0.6215** macro-F1 on the standard patient-disjoint test split.

Our contributions are:

1. **Support contrasts as diagnostic retrieval queries.** We convert positive-vs-negative support summaries into probes that retrieve information from the query's latent slots, and score what they retrieve with a shared scorer. This same architecture handles single-label diagnosis (one class vs the rest) and multi-label diagnosis (each label present vs absent).
2. **Cross-channel attention (with each component supported by ablations).** Cross-channel attention maps recordings with different channel counts to a fixed set of latent slots, while preserving per-channel descriptors. We characterize its invariance properties and attention cost. On PTB, cross-channel attention improves macro-F1 by **14.13** points over a pooling baseline with the same inputs and projections. Probe retrieval improves it by **2.12** points over probe pooling, and support-generated probes by **2.14** points over fixed learned queries. All three gains have bootstrap confidence intervals that exclude zero.
3. **Evaluation on PTB-XL and additional controls.** On PTB-XL, SIDP improves macro-F1 over pooled-descriptor MLP and support-conditioned MLP baselines by **7.50** and **9.97** points, respectively. Additional PTB experiments support the use of channel metadata and learned-mean support summaries. We also report patient-level bootstrap confidence intervals, full baseline tables, and cross-dataset studies of input representation and normalization.

2 Related work

Representations and multichannel structure. Forecasting models such as TimesFM (Das et al., 2024) and Time-LLM (Jin et al., 2024) address transferable temporal prediction, while UniTS (Gao et al., 2024) includes classification in a multi-task model. PatchTST (Nie et al., 2023) splits each series into temporal patches. iTransformer (Liu et al., 2024) uses each variate as a token, and Crossformer (Zhang & Yan, 2023) models temporal and cross-channel interactions separately. Leddam (Yu et al., 2024) combines decomposition with attention within and across series. For medical signals, EEGNet (Lawhern et al., 2018) is a compact convolutional network, Medformer (Wang et al., 2024a) uses patches at several time scales, and TeCh (Yu et al., 2026) aggregates channels into core tokens and redistributes them. SIDP links cross-channel attention to diagnostic retrieval that depends on the support set. Our ablation experiments test this link with pooling controls and by checking whether the attention weights change with the recording. Published scores of these models use other protocols, so we do not compare with them directly.

Repurposing and medical foundation models. FORMED (Huang et al., 2026) combines a frozen forecasting backbone with channel embeddings, learned label queries, and shared decoding attention. It is related to query-based decoders such as DETR (Carion et al., 2020). In SIDP, the class probes are computed from positive and negative support summaries. Our fixed-query ablation measures the value of this step, with the same PTB inputs and training budget. SIDP is trained on each dataset, so our evaluation differs from transfer across cohorts. CBraMod (Wang et al., 2025), AnyECG (Wang et al., 2024b), and MIRA (Li et al., 2025) provide reusable representations for EEG, ECG, and medical time series.

Support-conditioned prediction. Matching Networks (Vinyals et al., 2016) and Prototypical Networks (Snell et al., 2017) introduce episodic prediction from a support set. BD-CSPN (Liu et al., 2020) corrects the bias of prototypes by also using the unlabelled queries. Set Transformer (Lee et al., 2019) provides attention-based pooling of sets with inducing points. Dual-prototype methods (Jiang et al., 2024) combine class-specific and query-specific attention. FiLM (Perez et al., 2018) and HyperNetworks (Ha et al., 2017) condition a network's computation on context. SIDP turns a class-vs-rest support contrast into probes that retrieve values from the query's latent slots. Our positive-only, fixed-query, and probe-pooling controls test different parts of this construction.

In-context time-series classification. TabPFN (Hollmann et al., 2023) learns a classification procedure from synthetic tasks. ROCKET (Dempster et al., 2020) provides random convolutional features, and RocketPFN (O'Rourke et al., 2026) combines them with in-context tabular prediction. TIC-FM (Fang et al., 2026), TiCT (Yeh et al., 2025), and TimEE (Küken et al., 2026) use in-context time-series classifiers, and TimesFM-ICF (Das et al., 2025) uses in-context examples for forecasting. Mantis (Feofanov et al., 2026a) and MantisV2 (Feofanov et al., 2026b) focus on representations and readouts for classification. In contrast to these models, SIDP is trained with full supervision on each dataset before it predicts from a support set. Therefore, the support set is a small part of the labels that SIDP uses. Appendix A compares SIDP with the closest designs.

3 Support-induced diagnostic probing

3.1 Notation and setting

We first define the single-label construction, and then extend it to multi-label records (Figure 1). Let $X \in \mathbb{R}^{T\times C}$ be a recording window with T time steps and C channels, class $y \in \{1,\dots,J\}$, and patient identifier p. We split the training, validation, and evaluation sets by patient, so no patient appears in more than one set. This provides a more realistic assessment of generalization to unseen patients. Within a dataset, all sets use the same classes. At inference, the support set is $\mathcal{S} = \bigcup_{k=1}^{J} \mathcal{S}_k$. For each class, it contains K windows from each of P training patients. Single-label evaluation uses $P = 5$ and $K = 4$, giving $20J$ support windows. Each query window comes from a patient outside the training and validation sets.

We distinguish *fitting* from *inference*. Feature transforms, ridge coefficients, and SIDP parameters are fitted on the training split, and validation selects checkpoints. During inference, these parameters are fixed, and the support examples only build temporary class probes. The architecture and feature rules are the same for all datasets, but the fitted weights are not shared across datasets. Therefore, "no parameter updates" refers only to inference. SIDP is still trained once per dataset.

3.2 Spatial normalization and input representations

At each time step, we subtract the mean across channels and divide by the remaining spread across channels:

$$m_t = C^{-1}\sum_c X_{tc}, \qquad R_{tc} = X_{tc} - m_t, \qquad v_t = C^{-1}\sum_c R_{tc}^2, \tag{1}$$

$$\epsilon(X) = \max\left(10^{-6}\sqrt{T^{-1}\sum_t v_t}, 10^{-12}\right), \qquad \widetilde{X}_{tc} = \frac{R_{tc}}{\max(\sqrt{v_t}, \epsilon(X))}. \tag{2}$$

This keeps the signed differences between channels, while removing their common component and normalizing the remaining scale.

From each channel of $\widetilde{X}$, we compute 29 window-level descriptors covering location, dispersion, distribution shape, temporal variation, and spectrum (Appendix B). Concatenating all channels gives $f(X) \in \mathbb{R}^{29C}$. A transform fitted on the training split applies a signed log to a feature when this sufficiently reduces its skewness. It also removes constant features and scales each feature by its training median and interquartile range. We denote the resulting descriptor vector by $\phi(X)$. A class-balanced ridge classifier with penalty $\alpha = 100$ then produces class margins $r(X)$. Its weighted

least-squares objective is

$$\min_{B,b} \sum_{i \in \mathcal{D}_{\text{tr}}} w_{y_i} \|t(y_i) - B^\top \phi(X_i) - b\|_2^2 + 100\|B\|_F^2, \qquad w_k = \frac{n}{Jn_k}, \tag{3}$$

where $t(y)$ is the one-vs-rest target vector for class y (a single signed target for binary classification). For PTB, the scalar margin r is expanded to $[-r, r]$. Multi-class margins keep the class order. A second transform fitted on the training split produces $z(X) \in \mathbb{R}^J$. We then construct

$$H_{c,1,:}(X) = z(X), \qquad H(X) \in \mathbb{R}^{C \times 1 \times J}. \tag{4}$$

This *margin input* provides the same global margin vector to every channel. We use this input for our single-label results on ADFTD and PTB, where the same pipeline and settings are applied to both datasets. Because every channel receives the same vector, the cross-channel attention weights do not change across recordings with this input (Appendix J.1).

Per-channel descriptor input. To give channel-specific information directly to the classifier, our PTB experiments also use a *per-channel input*. It replaces Eq. (4) with

$$H^{\text{ch}}_{c,1,:}(X) = f_c(\widetilde{X}) \in \mathbb{R}^{29}, \qquad H^{\text{ch}}(X) \in \mathbb{R}^{15 \times 1 \times 29}, \tag{5}$$

where f_c is the per-channel descriptor map. This input skips the ridge classifier, so each channel keeps its own descriptors up to cross-channel attention. The latentizer is retrained for input width 29, while the attention, probe, and scoring design stays the same.

3.3 Cross-channel latent representation

The temporal latentizer maps each channel's L input tokens to learned slots $Q_t \in \mathbb{R}^{R_t \times d}$ by attention:

$$Z_c = \text{softmax}\left(\frac{Q_t (H_c W_K)^\top}{\sqrt{d}}\right) H_c W_V \in \mathbb{R}^{R_t \times d}. \tag{6}$$

In our experiments, we set $d = 64$, $R_t = 4$, and $R_c = 8$. The single-label inputs use $L = 1$, with width J for margins and 29 for per-channel descriptors. The multi-label input uses $L = 4$ and a width of 50. Channel metadata $m_c \in \mathbb{R}^{14}$, derived from the channel names, is encoded by a shared MLP and added to the temporal slots. We then apply cross-channel attention, which we call *canonical attention* in the rest of the paper. Its learned queries $A \in \mathbb{R}^{R_c \times d}$ attend to the flattened set of channel-slot vectors:

$$U_{cr} = Z_{cr} + g_m(m_c), \qquad P(X) = \text{LN}\Big(\text{MHA}(A, U^\flat, U^\flat)\Big) \in \mathbb{R}^{8 \times 64}, \tag{7}$$

where $U^\flat \in \mathbb{R}^{4C \times 64}$ and LN is an output LayerNorm. Canonical attention has four heads. Each learned query summarizes the metadata-tagged channel slots into one of the eight latent slots of $P(X)$. This gives a fixed-size representation for any channel count and keeps several slots for diagnostic retrieval. $P(X)$ represents one input window or one multi-label record. Appendix G gives the implementation details.

3.4 From support contrasts to query retrieval

For class k, we average the latent slots of its support windows. We separately average the slots of all other support windows, $\mathcal{S}_{\neg k} = \mathcal{S} \setminus \mathcal{S}_k$:

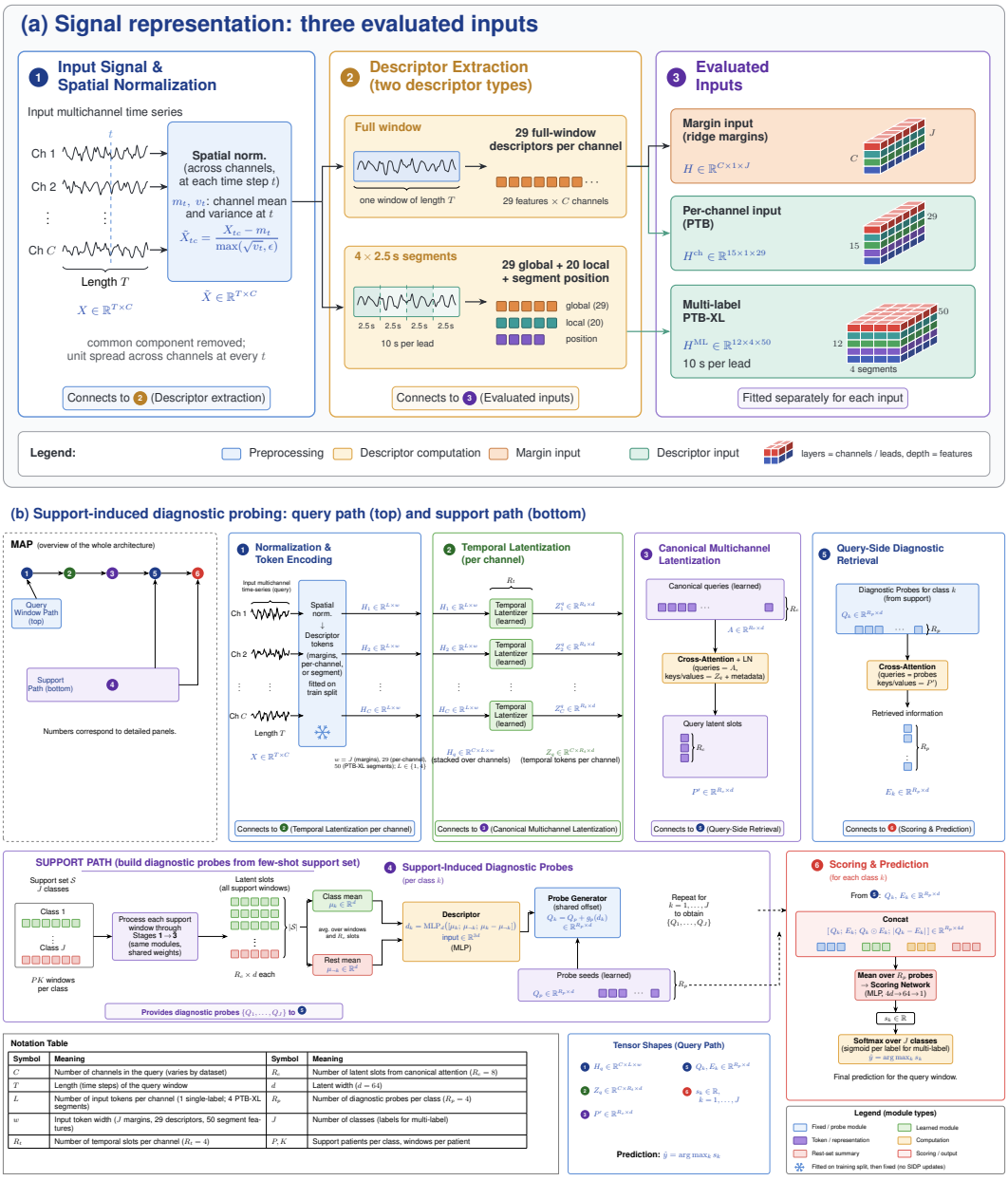
$$\mu_k = \frac{\sum_{X_i \in \mathcal{S}_k} \sum_{r=1}^{R_c} P(X_i)_r}{|\mathcal{S}_k| R_c}, \qquad \mu_{\neg k} = \frac{\sum_{X_i \in \mathcal{S}_{\neg k}} \sum_{r=1}^{R_c} P(X_i)_r}{|\mathcal{S}_{\neg k}| R_c}. \tag{8}$$

The single-label tasks have at least two classes, so $\mathcal{S}_{\neg k}$ is never empty. The support descriptor d_k and the $R_p = 4$ class probes are

$$d_k = \text{MLP}_d(\mu_k \,\|\, \mu_{\neg k} \,\|\, (\mu_k - \mu_{\neg k})), \qquad Q_k = Q_p + \mathbf{1}_{R_p} g_p(d_k)^\top \in \mathbb{R}^{4 \times 64}. \tag{9}$$

Q_p is a shared bank of learned probe seeds. Every seed receives the same class-specific offset. We do not generate a separate weight matrix for each class. For the query slots $P' = P(X')$, the probes retrieve class-specific information by attention:

$$E_k = \text{softmax}\left(\frac{(Q_k W_Q^p)(P' W_K^p)^\top}{\sqrt{d}}\right) P' W_V^p \in \mathbb{R}^{4 \times 64}. \tag{10}$$



Symbol	Meaning	Symbol	Meaning
C	Number of channels in the query (varies by dataset)	R_c	Number of latent slots from canonical attention ($R_c = 8$)
T	Length (time steps) of the query window	d	Latent width ($d = 64$)
L	Number of input tokens per channel (1 single-label; 4 PTB-XL segments)	R_p	Number of diagnostic probes per class ($R_p = 4$)
w	Input token width (J margins, 29 descriptors, 50 segment features)	J	Number of classes (labels for multi-label)
R_t	Number of temporal slots per channel ($R_t = 4$)	P, K	Support patients per class, windows per patient

Figure 1: Overview of SIDP. (a) Inputs. Each window is normalized across channels at every time step (Eqs. (1)–(2)). Full-window descriptors give the shared single-label margin input H (Eq. (4)) or the per-channel input H^{ch} (Eq. (5)). Four 2.5-second segments per lead give the multi-label PTB-XL input H^{ML}. Each input is fitted separately. (b) Probing. Query and support windows share the temporal latentizer and canonical attention, which produce R_c latent slots (Eqs. (6)–(7)). For each class, the support slots are averaged over windows and slots into a class mean and a rest mean in $\mathbb{R}^d$, which set the probes Q_k (Eqs. (8)–(9)). The probes retrieve E_k from the query slots. A shared scorer uses Q_k, E_k, their element-wise product, and their absolute difference, averaged over probes (Eqs. (10)–(12)). Single-label predictions use a softmax. For multi-label PTB-XL, label-positive and label-negative means replace the class and rest means, and each label receives an independent sigmoid. Descriptor transforms and the ridge classifier are fitted before SIDP training, and all parameters are fixed during inference.

The shared scorer then compares each probe with what it retrieves:

$$h_k = \frac{1}{R_p} \sum_{r=1}^{R_p} [Q_{kr} \,\|\, E_{kr} \,\|\, (Q_{kr} \odot E_{kr}) \,\|\, |Q_{kr} - E_{kr}|] \,, \tag{11}$$

$$s_k = \mathrm{MLP}_s(h_k), \qquad p(y = k \mid X', \mathcal{S}) = \frac{\exp(s_k/\tau)}{\sum_{j=1}^{J} \exp(s_j/\tau)}. \tag{12}$$

The scorer has widths $256 \to 64 \to 1$. The temperature τ is learned, positive, and clamped to $[0.05, 20]$. We predict the class with the largest score. The query slots are not pooled before this class-specific retrieval. Different diagnoses can therefore read different combinations of query slots. The product and absolute-difference terms tell the scorer where each probe and its retrieved vector agree and differ. The scorer thus learns how to compare each probe with what it retrieves. The support averages are taken over learned latent slots, so they can capture nonlinear properties of the signal.

Multi-label records and segment tokens. For the standard PTB-XL split, we divide each ten-second record into four 2.5-second segments per lead. Each segment token combines 29 global lead descriptors, 20 local non-spectral descriptors, and the position of the segment center, giving $H^{\mathrm{ML}} \in \mathbb{R}^{12\times4\times50}$. This input skips the ridge classifier and keeps the same internal SIDP widths. For label k, let $\mathcal{S}_k^+ = \{i : y_{ik} = 1\}$ and $\mathcal{S}_k^- = \{i : y_{ik} = 0\}$. Their averaged embeddings replace μ_k and $\mu_{\neg k}$ in Eq. (9). Each label receives an independent probability $p_k = \sigma(s_k)$, trained with binary cross-entropy:

$$\mathcal{L}_{\mathrm{ML}} = -\mathbb{E}_{(X,y)} \frac{1}{J} \sum_{k=1}^{J} [y_k \log p_k + (1 - y_k) \log(1 - p_k)]. \tag{13}$$

Co-occurring diagnoses can share positive examples. The negative set of a label is defined by its absence. Appendix J.4 gives the token construction, fitting, and support sampler.

3.5 Training

For the margin input, the ridge classifier is fitted before SIDP training and then kept fixed. The per-channel input uses the descriptors directly. We train single-label SIDP on episodes. Each episode samples a support set with $P = 4$ and $K = 4$, and up to 16 query windows from other training patients. We minimize the cross-entropy on the query windows:

$$\mathcal{L}(\Theta) = -\mathbb{E}_{(\mathcal{S},\mathcal{Q})} \frac{1}{|\mathcal{Q}|} \sum_{(X,y)\in\mathcal{Q}} \log p_\Theta(y \mid X, \mathcal{S}). \tag{14}$$

Query patients never appear in their episode's support. However, the fixed ridge classifier is fitted on all training patients, so its margins for training queries are in-sample. Appendix I.4 evaluates ridge classifiers fitted without the query patients.

Invariance properties. For fixed encoded inputs, reordering the support examples does not change any summary, probe, or score. For single-label tasks, two support sets with the same class counts and the same latent class means give identical predictions for a fixed query (Eqs. (8)–(12)). For multi-label tasks, the same holds when each label's positive and negative summaries are equal. Canonical attention is invariant to permuting the channels together with their metadata. However, the ridge classifier used for the margin input acts on ordered descriptors (Appendix C). Appendix D gives the attention cost.

4 Experiments

4.1 Datasets and evaluation

We evaluate SIDP on ADFTD EEG (Miltiadous et al., 2023), PTB ECG (Bousseljot, 2004), and PTB-XL ECG (Wagner et al., 2020) datasets. Table 1 lists the patients and windows in each split. Our main results are multi-label classification on the standard PTB-XL split (Section 4.2) and ablations on PTB with per-channel inputs (Section 4.3). The margin-input experiments use the same pipeline and settings for EEG and ECG, and provide additional controls (Sections 4.4–4.6).

Table 1: Patients and windows (or records) in the training/validation/test splits. No patient appears in more than one split. PTB-XL SL is a single-label task, and PTB-XL ML is the multi-label task on the standard split.

Task	Classes / channels	Patients	Windows or records
ADFTD	3 / 19	52 / 18 / 18	3328 / 1152 / 3996
PTB	2 / 15	120 / 40 / 40	7446 / 2508 / 3870
PTB-XL SL	5 / 12	2402 / 799 / 799	16814 / 5593 / 3995
PTB-XL ML	5 / 12	14823 / 1917 / 1877	17084 / 2146 / 2158

Table 2: Multi-label PTB-XL results on the test split (record level). Bold marks the best result.

Method	Input	AUROC	AUPRC	Macro-F1
Linear	Pooled descriptors	.7351	.4897	.5072
MLP	Pooled descriptors	.7791	.5483	.5465
Support MLP	Pooled descriptors	.7574	.5057	.5218
Prototype	Pooled descriptors	.5327	.2894	.3639
SIDP	Segment descriptors	**.8450**	**.6562**	**.6215**

Single-label evaluation. For each single-label task, we report window macro-F1 averaged over three training seeds and 32 support sets. We compute macro-F1 for each run before averaging (Appendix E). If a class has no true and no predicted windows in a run, its F1 term is set to zero. All compared methods receive the same inputs and use the same training schedule and checkpoint selection. The MLP baselines flatten the channel tensor. We did not tune the baselines separately (Appendix J). Appendix B gives the sampling and normalization details.

Confidence intervals. We compute paired 95% confidence intervals from 2,000 bootstrap samples of test patients. For single-label tasks, we sample patients within each class. For multi-label PTB-XL, each record keeps all of its diagnoses. The intervals describe variation over test patients for fixed trained models and support sets. Unless stated otherwise, we do not correct them for multiple comparisons. The single-label test patients were also evaluated during model development (Appendix H).

4.2 Multi-label PTB-XL classification

We use PTB-XL v1.0.3 at 100 Hz, with complete ten-second records and the standard patient-disjoint split recommended by the dataset authors (Wagner et al., 2020). Each record has one or more of five diagnostic superclasses (NORM, MI, STTC, CD, and HYP). SIDP receives the $12 \times 4 \times 50$ segment tokens from Section 3.4. All the compared baselines receive the same descriptors averaged over leads and segments. Therefore, these comparisons measure the combined effect of keeping the lead and segment structure, and of the SIDP classifier. Each label's decision threshold is chosen on the validation set and then fixed for testing. Scores are averaged over three seeds and 32 support sets, each with 49–50 training patients. AUPRC denotes average precision. Appendix J.4 gives the support sets, per-class results, compute, and confidence intervals.

SIDP improves on classifiers that use pooled descriptors. SIDP achieves **0.8450** macro-AUROC, **0.6562** average precision, and **0.6215** macro-F1 (95% CI [0.6060, 0.6359]). Its macro-F1 is **7.50** points [6.33, 8.74] higher than the MLP, and **9.97** points [8.82, 11.10] higher than the support MLP. It is also 11.43 and 25.76 points higher than the linear and prototype baselines, and both intervals exclude zero. Section 4.3 isolates the contribution of each component on the PTB dataset.

4.3 Ablations

For ablation experiments, every method receives the same $15 \times 1 \times 29$ per-channel PTB descriptors. SIDP achieves **0.8603** window macro-F1, the highest mean among the seven methods. We compare SIDP with three controls, each of which changes one component. Canonical pooling replaces cross-channel attention with mean pooling, and keeps the value and output projections. Probe pooling

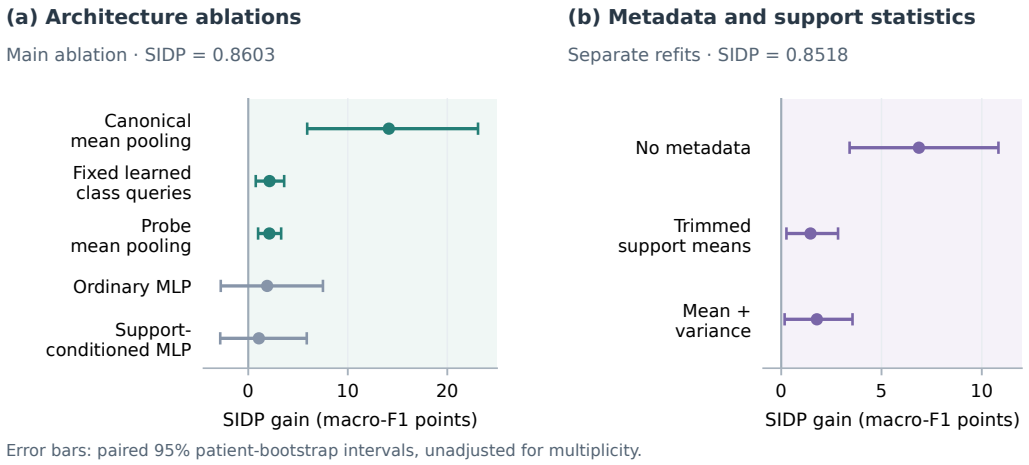


Figure 2: Ablations on the PTB dataset. Points show the macro-F1 difference between SIDP and each control, and bars show paired 95% bootstrap confidence intervals over test patients. (a) Cross-channel attention and probe controls with per-channel descriptors. (b) Channel metadata and support summaries, from separate refits. The two panels use different trained models. Appendix I reports all scores.

replaces probe retrieval with mean pooling, and keeps the value projection. Fixed learned queries replace the support-generated probes. Therefore, each control keeps the same feature transforms and changes only how information is selected.

Both retrieval stages contribute. Cross-channel (canonical) attention improves macro-F1 by **14.13** points [5.92, 23.09] over canonical pooling. Probe retrieval improves it by **2.12** points [0.99, 3.31] over probe pooling, and support-generated probes by **2.14** points [0.75, 3.61] over fixed learned queries (Figure 2a). These results suggest that both attention stages and the support-generated probes contribute. The MLP and support MLP reach 0.8413 and 0.8496, respectively. SIDP is higher by 1.90 and 1.07 points, but these two intervals include zero (Table 10).

Patient-level results. We average the window probabilities of each patient, an aggregation rule chosen on the validation set. SIDP reaches **0.9661** patient macro-F1 and **0.9609** balanced accuracy on the 40 PTB test patients. This is **5.74** points [0.25, 14.01] higher than the support MLP.

Recording-dependent retrieval. To verify that both attention stages adapt to each recording, we fix the support set and pass one window from each validation patient through the trained models. For each attention layer, we record the largest change of any attention weight across recordings. On PTB, this change is 0.2361–0.2441 for canonical attention and 0.2555–0.9990 for probe attention across seeds. With per-channel inputs, the attention weights also change across recordings on ADFTD and single-label PTB-XL (Appendix J.1).

4.4 Support contrasts, summaries, and channel metadata

We measure how SIDP uses the support set on the single-label PTB-XL task, where all methods receive the same ridge margins (Appendix I.2). This task has different patients, targets, and models from the multi-label task in Section 4.2. SIDP reaches **0.4258** macro-F1, which is **2.04** points [1.21, 2.88] higher than the support MLP, after correcting for six comparisons. Building probes from both positive and negative support adds **0.43** points [0.06, 0.81] over probes built from positive support only. However, the MLP without support scores 0.96 points higher than SIDP on this task [0.30, 1.63]. Therefore, we use this task only to compare ways of using the support set.

In separate PTB refits with per-channel inputs, SIDP reaches **0.8518** macro-F1. Averaging the learned support representations improves macro-F1 by **1.46** points [0.26, 2.83] over trimmed means,

Table 3: Macro-F1 of the reference methods, and of SIDP and an MLP with the margin input. The reference columns give the dataset and the input used by the reference methods: CBraMod embeddings (Wang et al., 2025) or spectral-statistic features. The FORMED and in-context attention entries are simplified stand-ins and do not reproduce the complete published systems. A separate PTB set-pooling result (.6977) gives 22 reference entries in total.

Method	ADFTD CBraMod	ADFTD Spectral stats	PTB Spectral stats
Linear probe	.408	.414	.821
ProtoNet (Snell et al., 2017)	.348	.332	.749
BD-CSPN (Liu et al., 2020)	.298	.316	.719
FORMED adapter stand-in	.377	.420	.754
FORMED-shaped decoder	.272	.370	.719
In-context attention stand-in	.305	.270	.766
TabPFN (Hollmann et al., 2023)	.335	.325	.785
	ADFTD, margin input		PTB, margin input
MLP	.5080		.8379
SIDP	.5043		.8410

and by **1.78** points [0.17, 3.55] over adding the support variance. Removing the channel metadata lowers macro-F1 by **6.87** points [3.41, 10.83]. These results suggest that simple support averages and channel metadata both help in this setting. The metadata are derived from the channel names (Appendix I.5).

4.5 Normalization statistics

By default, ADFTD and PTB are normalized with unlabelled statistics from each patient's recording. To check whether our results depend on these statistics, we refit all models with statistics from the current window only, or from the training set. These refits use a separate set of one-second windows. On PTB, SIDP reaches **0.8748** with window statistics and **0.8310** with training-set statistics, compared to 0.8299 with patient statistics. On ADFTD, the corresponding scores are **0.4861**, **0.4887**, and 0.5014. All four differences have confidence intervals that include zero. Therefore, we conclude that there is no clear drop when the patient statistics are removed (Appendix J.3).

4.6 Comparison with reference methods

With the margin input, SIDP reaches **0.5043** macro-F1 on ADFTD and **0.8410** on PTB. Table 3 compares these results with 21 reference results. The reference methods use other inputs and protocols, so these comparisons are unpaired. SIDP scores above all 22 reference entries, by 8.43 points on ADFTD and 2.00 points on PTB over the best one. However, an MLP on the same margin input also scores above all of them. Therefore, this gain mainly reflects the margin input. In addition, we selected the final margin pipeline because it scored above all 22 entries (Appendix H).

5 Discussion and conclusion

SIDP generates probes from positive-vs-negative support contrasts in order to retrieve diagnosis-specific information from the query's latent slots. On the PTB dataset with per-channel descriptors, cross-channel attention, probe retrieval, and support-generated probes each improve over controls. On multi-label PTB-XL, the same architecture handles co-occurring diagnoses and improves over classifiers that use pooled descriptors. Together, these results suggest that letting the support set decide what to retrieve from each recording is a useful design for medical time-series classification.

Our results also have clear limits. First, the component gains are specific to the PTB dataset. With the same per-channel inputs, SIDP scores below the MLP baselines on ADFTD and single-label PTB-XL (Appendix J.1). Second, SIDP is trained separately for each dataset, and we do not evaluate transfer to new datasets or new classes without training. Future work includes applying SIDP to learned or pretrained encoders, transfer across cohorts, and support sets with real annotation noise.

REPRODUCIBILITY STATEMENT

Section 3 describes the inputs and the single-label and multi-label versions of SIDP. Section 4 describes the datasets, patient splits, baselines, and evaluation. Appendices B–E give the descriptors, proofs, training settings, and confidence-interval computation. Appendix G lists the implementation details needed for a full re-implementation. Appendices I–J report additional ablations and cross-dataset results, including methods that score higher than SIDP. Appendix J.4 gives the multi-label PTB-XL tokens, support counts, thresholds, and compute. We publicly release the source code of our method at https://anonymous.4open.science/r/sidp-85DD.

ETHICS STATEMENT

This work concerns the classification of public EEG and ECG datasets. Window-level benchmark results do not establish prospective diagnostic validity. Small class-specific patient counts and offline recording normalization restricts clinical interpretation. Attention-derived latent evidence is not a validated medical explanation. Dataset access terms and the original studies' consent and ethics conditions remain applicable. This manuscript makes no claim of a new ethics approval or clinical deployment validation.

AI USE STATEMENT

We report the use of generative LLMs (OpenAI ChatGPT and Anthropic Claude) in this work for literature discovery, code implementation, and help with writing. We (the authors) have reviewed all AI-assisted content and take full responsibility for the paper.

REFERENCES

Ralf-Dieter Bousseljot. PTB Diagnostic ECG Database. PhysioNet, version 1.0.0, 2004. URL https://physionet.org/content/ptbdb/1.0.0/.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*, 2020. URL https://arxiv.org/abs/2005.12872.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2310.10688.

Abhimanyu Das, Matthew Faw, Rajat Sen, and Yichen Zhou. In-Context Fine-Tuning for Time-Series Foundation Models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2410.24087.

Angus Dempster, François Petitjean, and Geoffrey I. Webb. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34:1454–1495, 2020. doi: 10.1007/s10618-020-00701-z. URL https://arxiv.org/abs/1910.13051.

Juntao Fang, Shifeng Xie, Shengbin Nie, Yuhui Ling, Yuming Liu, Zijian Li, Keli Zhang, Lujia Pan, Themis Palpanas, and Ruichu Cai. Rethinking Zero-Shot Time Series Classification: From Task-specific Classifiers to In-Context Inference. arXiv:2602.00620, 2026. URL https://arxiv.org/abs/2602.00620.

Vasilii Feofanov, Songkang Wen, Shifeng Xie, Simon Roschmann, Marius Alonso, Hongbo Guo, Romain Ilbert, Malik Tiomoko, Quentin Bouniot, Zeynep Akata, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight Foundation Model for Time Series Classification. arXiv:2502.15637, 2026a. URL https://arxiv.org/abs/2502.15637.

Vasilii Feofanov, Songkang Wen, Jianfeng Zhang, Lujia Pan, and Ievgen Redko. MantisV2: Closing the Zero-Shot Gap in Time Series Classification with Synthetic Data and Test-Time Strategies. arXiv:2602.17868, 2026b. URL https://arxiv.org/abs/2602.17868.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: A Unified Multi-Task Time Series Model. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2403.00131.

David Ha, Andrew Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.09106.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2207.01848.

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*, 2026. URL https://arxiv.org/abs/2410.03794.

Lei Jiang, Yongzhao Zhan, Zhen Jiang, and Na Tang. A dual-prototype network combining query-specific and class-specific attentive learning for few-shot action recognition. *Neurocomputing*, 598:127819, 2024. doi: 10.1016/j.neucom.2024.127819. URL https://www.sciencedirect.com/science/article/pii/S0925231224005903.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.01728.

Jaris Küken, Shi Bin Hoo, Martin Mráz, Frank Hutter, and Lennart Purucker. TimEE: End-to-end Time Series Classification via In-Context Learning. arXiv:2607.07500, 2026. URL https://arxiv.org/abs/2607.07500.

Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. *Journal of Neural Engineering*, 15(5):056013, 2018. doi: 10.1088/1741-2552/aace8c. URL https://arxiv.org/abs/1611.08024.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *International Conference on Machine Learning*, 2019. URL https://arxiv.org/abs/1810.00825.

Hao Li, Bowen Deng, Chang Xu, Zhiyuan Feng, Viktor Schlegel, Yu-Hao Huang, Yizheng Sun, Jingyuan Sun, Kailai Yang, Yiyao Yu, and Jiang Bian. MIRA: Medical Time Series Foundation Model for Real-World Health Data. arXiv:2506.07584, 2025. URL https://arxiv.org/abs/2506.07584.

Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype Rectification for Few-Shot Learning. In *European Conference on Computer Vision*, 2020. URL https://arxiv.org/abs/1911.10713.

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*, 2024. URL https://arxiv.org/abs/2310.06625.

Andreas Miltiadous, Katerina D. Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G. Tsalikakis, Pantelis Angelidis, Markos G. Tsipouras, Euripidis Glavas, Nikolaos Giannakeas, and Alexandros T. Tzallas. A dataset of scalp eeg recordings of alzheimer's disease, frontotemporal dementia and healthy subjects from routine eeg. *Data*, 8(6):95, 2023. URL https://www.mdpi.com/2306-5729/8/6/95.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730.

Franco Martino O'Rourke, Ana Trisovic, and Dimitris Bertsimas. RocketPFN: Accurate Time Series Classification via In-Context Learning. arXiv:2606.21786, 2026. URL https://arxiv.org/abs/2606.21786.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://arxiv.org/abs/1709.07871.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1703.05175.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, 2016. URL https://arxiv.org/abs/1606.04080.

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*, 7:154, 2020. doi: 10.1038/s41597-020-0495-6. URL https://www.nature.com/articles/s41597-020-0495-6.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. CBraMod: A Criss-Cross Brain Foundation Model for EEG Decoding. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.07236.

Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. Medformer: A Multi-Granularity Patching Transformer for Medical Time-Series Classification. In *Advances in Neural Information Processing Systems*, 2024a. URL https://arxiv.org/abs/2405.19363.

Yue Wang, Xu Cao, Yaojun Hu, Haochao Ying, Hongxia Xu, Ruijia Wu, James Matthew Rehg, Jimeng Sun, Jian Wu, and Jintai Chen. AnyECG: Foundational Models for Multitask Cardiac Analysis in Real-World Settings. arXiv:2411.17711, 2024b. URL https://arxiv.org/abs/2411.17711.

Chin-Chia Michael Yeh, Uday Singh Saini, Junpeng Wang, Xin Dai, Xiran Fan, Jiarui Sun, Yujie Fan, and Yan Zheng. TiCT: A Synthetically Pre-Trained Foundation Model for Time Series Classification. arXiv:2511.19694, 2025. URL https://arxiv.org/abs/2511.19694.

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I. Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing Multivariate Time Series Forecasting: Learnable Decomposition with Inter-Series Dependencies and Intra-Series Variations Modeling. In *International Conference on Machine Learning*, pp. 57818–57841, 2024. URL https://proceedings.mlr.press/v235/yu24s.html.

Guoqi Yu, Juncheng Wang, Chen Yang, Jing Qin, Angelica I. Aviles-Rivero, and Shujun Wang. Decentralized Attention Fails Centralized Signals: Rethinking Transformers for Medical Time Series. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.18473.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/pdf/1d793d6ba7c00ecfe98128614d58e2493255bd89.pdf.

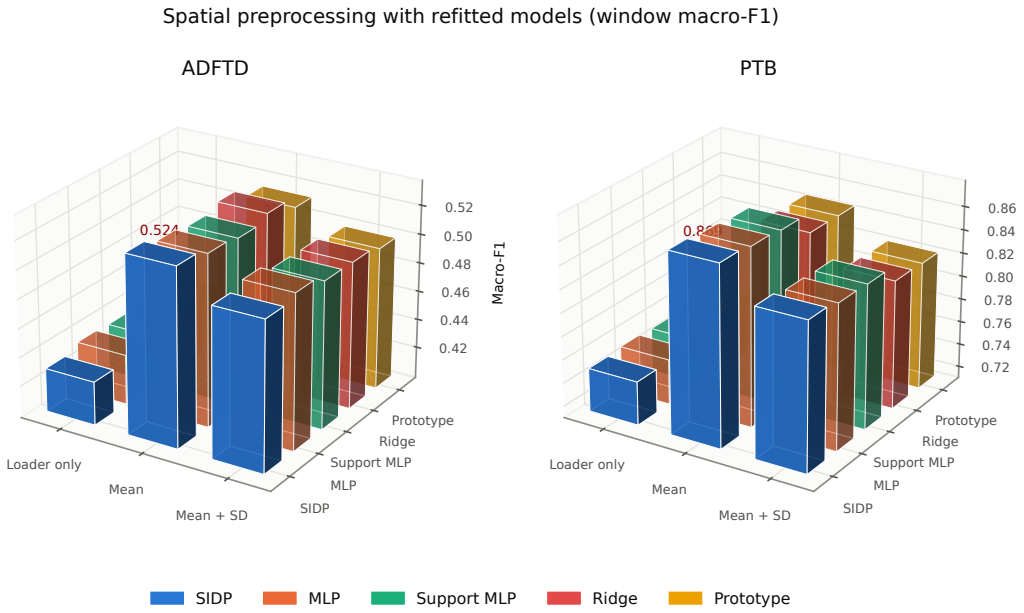


Figure 3: Window macro-F1 on ADFTD and PTB for three spatial preprocessing settings, with all models refitted for each setting (Table 26). The label in each panel marks the highest score.

A Positioning the technical contributions

Our contribution is the combination of the following three design decisions, in order. First, cross-channel (canonical) attention maps each recording to a fixed set of latent slots. Then, class-vs-rest support contrasts produce class probes. Finally, these probes retrieve information from the query's slots. Table 4 compares this design with its closest precedents.

Table 4: Design comparisons with the closest methods.

Approach	Related mechanism	How SIDP differs
ProtoNet (Snell et al., 2017)	One mean support embedding per class	The means generate probes that retrieve information from the query slots.
Matching Networks (Vinyals et al., 2016)	Attention over individual support examples	Retrieval reads the query slots after the support set is summarized.
Set Transformer (Lee et al., 2019)	Learned seeds pool an unordered set	Cross-channel pooling is followed by class-specific diagnostic probing.
FORMED (Huang et al., 2026)	Learned embeddings and label queries	The probes are computed from the current support set.
TeCh (Yu et al., 2026)	Core-token aggregation and redistribution	Several latent slots feed a class-specific retrieval step.
Dual prototypes (Jiang et al., 2024)	Class and query attentive prototypes	Class contrasts generate the probes, and the retrieved values come from the query.
TIC-FM / TimEE (Fang et al., 2026; Küken et al., 2026)	In-context classification	SIDP is a multi-channel probe classifier, and we currently train it on each dataset.

B DESCRIPTORS AND PREPROCESSING

All datasets use the same pipeline after loading. For ADFTD and PTB, the loader first standardizes each channel with statistics from the patient's retained recording. PTB-XL skips this step. We then apply the spatial normalization of Eqs. (1)–(2). The spectral descriptors use a Hann-windowed

Table 5: The 29 descriptors, computed separately for each spatially normalized channel.

Group	Count	Descriptors
Location / dispersion	9	Mean; SD; median; median absolute deviation; IQR; 5th and 95th percentiles; peak-to-peak amplitude; RMS.
Distribution shape	2	Signed skewness; excess kurtosis.
Temporal variation	9	Mean absolute first difference; first-difference SD; Hjorth mobility and complexity; zero-crossing rate; slope per second; linear-trend variance fraction; detrended residual SD; 100 ms autocorrelation.
Spectrum	9	Relative power in 0.5–4, 4–8, 8–13, 13–30, and 30–50 Hz; spectral centroid; entropy; peak frequency; 90% roll-off.

FFT over the whole window. We use the same frequency bands for EEG and ECG, although they have different physiological meanings in the two signals. The trend features use linear regression. The final pipeline has no calendar features, outlier clipping, correlation pruning, or Yeo–Johnson transform.

For feature j, let $a_j = \max(\mathrm{median}_{\mathrm{tr}}\,|f_j|, 10^{-6})$. The candidate transform is

$$g_j(f_j) = \mathrm{sign}(f_j)\log(1 + |f_j|/a_j). \tag{15}$$

We apply it only if the absolute training skewness of the feature is > 1, and the transform reduces it by more than 20% (up to a numerical tolerance). We then center each feature by its training median and scale it by its training IQR. If the IQR is zero, we use the training SD, and then one. We remove features whose training peak-to-peak range is at most 10^{-12}. We use kurtosis only as a descriptor. A second fitted signed-log and robust-scaling transform scales the margin coordinates.

The raw feature widths are 551, 435, and 348 for ADFTD, PTB, and PTB-XL. With the margin input, the SIDP inputs are $19 \times 1 \times 3$, $15 \times 1 \times 2$, and $12 \times 1 \times 5$. The binary PTB margin r is expanded to $[-r, r]$. The per-channel PTB input keeps the $15 \times 1 \times 29$ descriptors of Eq. (5). The exact feature masks and coefficients are part of each fitted encoder.

For single-label PTB-XL, we keep one record per patient and assign the superclass with the highest-confidence diagnostic annotation. Ties are resolved by the loader's iteration order. This 4,000-patient subset is separate from the full multi-label benchmark and its standard split. For PTB, we keep all 200 HC and MI patients. The FORMED report uses 198 patients but does not document which two it excludes. We sample at most $\lfloor 4000/N_{\mathrm{patients}} \rfloor$ query windows per patient, and keep all windows when fewer are available. This gives the counts in Table 1.

C STRUCTURAL PROPERTIES AND PROOFS

The following results are algebraic properties of the operators.

Proposition 1 (Common-offset invariance)**.** *Let $\mathcal{N}$ denote Eqs. (1)–(2). For any scalar time series b_t, $\mathcal{N}(X + b\mathbf{1}_C^\top) = \mathcal{N}(X)$. For a constant $a > 0$, $\mathcal{N}(aX + b\mathbf{1}_C^\top) = \mathcal{N}(X)$ if the relative epsilon term is larger than the absolute floor in both windows.*

Proof. Adding b_t shifts m_t by b_t and leaves the residuals, variances, and epsilon unchanged. Multiplying by a scales the residuals and SD by a. Under the stated condition, $\epsilon(aX) = a\epsilon(X)$, so the numerator and denominator scale equally. Because of the absolute floor, the statement does not hold for arbitrarily small signals. □

Proposition 2 (Canonical set invariance)**.** *Canonical attention is invariant to permuting its key and value rows together, including the attached metadata.*

Proof. For a permutation matrix Π, the row-wise softmax satisfies $\mathrm{softmax}(B\Pi^\top) = \mathrm{softmax}(B)\Pi^\top$. Hence $\mathrm{Attn}(A, \Pi K, \Pi V) = \mathrm{softmax}(AK^\top\Pi^\top/\sqrt{d})\Pi V = \mathrm{Attn}(A, K, V)$. The identity holds for each head and after the output projection. It applies to the set of attention inputs. The descriptor and ridge steps before attention act on ordered channels. □

For single-label tasks, where the support classes are disjoint, the following sufficiency result holds.

Proposition 3 (Support-summary sufficiency)**.** *For a fixed encoder and query, two support sets with the same class counts and the same class-mean embeddings give identical probes and predictions. In particular, reordering the support examples does not change the predictions.*

Proof. Let $n_k = |\mathcal{S}_k|$ and $u_k = \sum_{i\in\mathcal{S}_k}\sum_r P(X_i)_r$. Equal counts and means give equal n_k and u_k. Therefore, both $\mu_k = u_k/(n_k R_c)$ and $\mu_{\neg k} = (\sum_j u_j - u_k)/((\sum_j n_j - n_k)R_c)$ are equal. Equations (9)–(12) are deterministic functions of these summaries and the fixed query. Reordering preserves sums and counts. □

For balanced support, $\mu_{\neg k} = (J-1)^{-1}\sum_{j\neq k}\mu_j$. Therefore, the negative support gives the model the latent means of the competing classes. Averaging after a nonlinear embedding can keep higher-order properties of the raw signal. For example, averaging (x, x^2) keeps the first and second raw moments. This example only illustrates the idea, since SIDP learns its own embedding. The proposition requires equal *latent* means and counts. Appendix I.3 tests whether adding the within-support variance improves the model.

Proposition 4 (Temporal pooling and input scope)**.** *When $L = 1$, each attention-weighted row of Eq. (6) equals the single projected value. With the margin input, and for fixed metadata and support, the query prediction depends on the waveform only through $z(X') \in \mathbb{R}^J$.*

Proof. The softmax over a single key equals one, which gives $Z_{cr} = H_{c,1,:}W_V$ for every temporal slot. Slot-dependent residual terms, if present in the implementation, cannot add information from the waveform. With the margin input, all channel inputs are functions of the same $z(X')$, and the later operations are deterministic functions of these inputs and the metadata. Therefore, equal z and metadata give equal predictions for a fixed support set. The per-channel PTB input depends on the full 15×29 descriptor tensor, and the multi-label input keeps a $12 \times 4 \times 50$ tensor. □

Proposition 4 separates temporal tokenization within a channel from aggregation across channels. The per-channel input gives separate descriptor vectors to canonical attention and bypasses z. With $L = 1$, these channel vectors still differ from each other. Table 11 shows that its canonical and probe attention weights change with the recording, and Table 10 measures their effect on accuracy. These conclusions apply to the per-channel PTB input. We do not interpret the attention weights as clinical explanations.

D Training and inference

In the margin-input experiments, the ridge classifier uses a Cholesky solver, class balancing, and $\alpha = 100$. It stays fixed during SIDP training. The three SIDP seeds share one fitted ridge classifier for each dataset. We train SIDP for 1,800 updates with support sets of $P = 4$ patients and $K = 4$ windows, and up to 16 query windows from other training patients. We use AdamW with a learning rate of 10^{-3}, weight decay of 10^{-4}, cosine decay, and gradient clipping at norm 5. We do not add artificial noise to the support windows. For validation, we draw eight support sets ($P = 5$, $K = 4$) with support seed 20260918. We compare the checkpoints at 600, 1,200, and 1,800 updates, and keep the earlier one in case of a tie. Appendix I describes the per-channel and encoder-supervision variants.

Attention cost. For general L, the temporal and canonical attention products cost $O(CLR_t d + CR_t R_c d)$ per window. Retrieval costs $O(JR_p R_c d)$ per query. Averaging an already encoded support set costs $O(|\mathcal{S}|R_c d)$, and the class and complement sums can be cached. These bounds exclude descriptor extraction, ridge fitting, projections, and MLPs. With fixed slot counts, the attention cost is linear in the number of input tokens. Appendix J.4 reports measured model sizes and fitting times.

Algorithm 1 Single-label SIDP inference after training on a dataset

Require: Fixed encoders and SIDP parameters; labelled support set $\mathcal{S}$; query X'.

for each support window and the query **do**
  Normalize the channels, compute descriptors, and apply the fitted feature transforms.
  Form margin tokens with Eq. (4), or per-channel descriptors with Eq. (5).
  Compute the latent slots $P(X)$ with channel metadata.
end for
Cache the class counts n_k and slot sums u_k; let $u = \sum_k u_k$ and $n = \sum_k n_k$.
for $k = 1, \ldots, J$ **do**
  $\mu_k \leftarrow u_k/(n_k R_c)$; $\mu_{\neg k} \leftarrow (u - u_k)/((n - n_k)R_c)$.
  Build Q_k with Eq. (9), and retrieve E_k with Eq. (10).
  Average the probe/retrieval interactions and compute s_k.
end for
return $\mathrm{softmax}(s/\tau)$ and $\arg\max_k s_k$.

With $R_t = 4$, $R_c = 8$, and $R_p = 4$, the temporal, canonical, and retrieval stages have $4CL$, $32C$, and $32J$ attention entries per query for each attention map. Canonical attention uses four heads, and the projections and MLPs add further cost. For single-label tasks, each complement sum is the total sum minus the class sum. We did not measure inference time or memory for the single-label models.

Table 6: Seed results for the margin input on ADFTD and PTB. The single-label and multi-label PTB-XL results are reported separately. Each seed score averages 32 support sets.

Dataset	Seed 1 / 2 / 3 F1	Selected updates (1 / 2 / 3)	Validation F1
ADFTD	.506371 / .505722 / .500820	1800 / 1800 / 1200	.541198
PTB	.844087 / .839155 / .839661	1200 / 1800 / 1800	.832024

Evaluation support sets use seed 20260921. The $P = 10$, $K = 2$ budget keeps 20 windows per class, but doubles the number of support patients. The $P = 5$ patients are a subset of the $P = 10$ patients, and for shared patients, the $K = 2$ windows are a subset of the $K = 4$ windows. The complete support sets are not nested, because the two budgets trade patients for windows per patient. Both budgets only use patients with at least four windows.

E Metric aggregation and patient bootstrap

Let M_{sbp} be the window confusion matrix of patient p in the run with seed s and support set b. We score each run from $\sum_p M_{sbp}$ and average the 96 run scores. Because F1 is nonlinear, summing all confusion matrices before scoring would give a different estimate. The reported score is

$$\widehat{F} = \frac{1}{96} \sum_{s=1}^{3} \sum_{b=1}^{32} \frac{1}{J} \sum_{k=1}^{J} \frac{2\,\mathrm{TP}_{sbk}}{2\,\mathrm{TP}_{sbk} + \mathrm{FP}_{sbk} + \mathrm{FN}_{sbk}}. \tag{16}$$

In bootstrap replicate r, we resample the original number of patients with replacement within each true class. Let $w_p^{(r)}$ be the number of times patient p is drawn. We compute

$$F^{(r)} = \frac{1}{96} \sum_{s,b} \mathrm{MacroF1}\left(\sum_p w_p^{(r)} M_{sbp} \right). \tag{17}$$

All prediction runs use the same patient counts $w_p^{(r)}$. The interval is given by the 2.5th and 97.5th percentiles of 2,000 replicates (seed 20260920). Paired differences use the same resampled patients for both models. These intervals are conditional on the fitted models, class counts, and support sets, and they do not account for model selection. The 96 runs are repeated evaluations of the same test patients.

We keep 222 query windows per ADFTD patient, 32–100 per PTB patient, and five per PTB-XL patient. Macro-F1 gives each class equal weight. Within a class, patients with more windows contribute more. For PTB, macro-F1 averages the HC and MI classes.

F CLASS RESULTS

Table 7: Per-class results with the margin input, each averaged over the 96 runs. The mean F1 can differ from the harmonic mean of the mean precision and recall. Appendix J.4 gives the per-class results for multi-label PTB-XL.

Dataset	Class	Patients	Windows	Precision	Recall	F1
ADFTD	AD	7	1554	.5110	.5055	.5078
	FTD	5	1110	.2973	.2420	.2666
	CN	6	1332	.6857	.8003	.7385
PTB	HC	10	1000	.7618	.7682	.7645
	MI	30	2870	.9190	.9159	.9174

AD is Alzheimer's disease, FTD is frontotemporal dementia, CN and HC are healthy controls, and MI is myocardial infarction. The mean window accuracy and balanced accuracy are 0.5306 and 0.5159 for ADFTD, and 0.8777 and 0.8420 for PTB. The PTB-XL labels are NORM, MI, STTC, CD, and HYP (normal ECG, myocardial infarction, ST/T change, conduction disturbance, and hypertrophy).

G IMPLEMENTATION AND REPRODUCIBILITY

The implementation uses a latent width of 64, four temporal slots, eight canonical slots, four probes, four canonical attention heads, and 14 metadata fields derived from the channel names. The metadata contain the acquisition order and name-derived indicators, and Appendix I.5 evaluates a correction of the Frank-lead modality flags. These fields are encodings of the channel names, and they do not contain measured electrode positions. A full reimplementation also needs the complete metadata definition, descriptor guards, fitted feature masks and statistics, ridge coefficients, MLP hidden widths and activations, attention residual and bias settings, the temperature parameterization, the exact patient and record indices, the software environment, and the checkpoints. The additional PTB refits in Appendix I.3 ran on CPU. Appendix J.4 reports parameter counts and fitting times for multi-label PTB-XL.

A recomputation from the saved predictions found no score differences across 337 single-label method/setting rows in 57 groups. The multi-label methods were also checked.

H MODEL SELECTION

Table 8: Margin-input results on ADFTD and PTB. The gain is measured over the best reference entry in Table 3, which used different inputs and protocols.

Dataset	Macro-F1	Patient 95% interval	Seed SD	Gain (points)
ADFTD	.5043	[.4271, .5860]	.003035	+8.43
PTB	.8410	[.7466, .9131]	.002713	+2.00

Section 4.6 compares the margin-input results with 22 reference entries (Table 3), and Table 8 gives their intervals. During development, we compared two candidate pipelines. The spatial-SD candidate uses the spatial normalization of Section 3.2 with the full SIDP model. The anchored candidate uses engineered tokens with log and robust scaling, and an anchored SIDP correction. Spatial-SD improves ADFTD by 0.1551 with an interval that excludes zero. On PTB, the interval of the difference includes zero (Table 9). The two candidates also used different amounts of training computation. The anchored candidate had a higher mean validation macro-F1 over ADFTD and PTB (0.7019 vs 0.6866), and scored above 17 of the 22 reference entries. We selected spatial-SD because it scored above all 22. Therefore, this choice used test scores, and the test patients had also been evaluated during development. The experiments in Appendices I–J select checkpoints on validation, but they use the same test patients.

Table 9: The two candidate pipelines during development ($P = 5$, $K = 4$). The anchored pipeline is a different pipeline and not an ablation of spatial-SD. Both sets of checkpoints were fixed before this comparison, and the comparison with the reference entries influenced the final choice. Intervals are paired patient-bootstrap 95% intervals.

Dataset	Spatial-SD	Anchored	Difference	Paired interval
ADFTD	0.5043	0.3492	+0.1551	[+0.0472, +0.2631]
PTB	0.8410	0.8437	−0.0028	[−0.0722, +0.0669]

Changing the support budget from $P = 5$, $K = 4$ to $P = 10$, $K = 2$ keeps 20 support windows per class, but doubles the number of support patients. The margin model then scores 0.5041 on ADFTD and 0.8409 on PTB, close to the $P = 5$, $K = 4$ values.

I Extended experiments on retrieval, support summaries, and metadata

These experiments test SIDP's learned retrieval, how it uses the support set, and the channel metadata. In all of them, we train the models on each dataset and then predict from support sets with fixed parameters. We do not transfer a trained model between datasets.

Shared protocol. All learned models use three seeds, the same training patients, and 1,800 episodic updates. We select the checkpoint at 600, 1,200, or 1,800 updates by validation macro-F1 over eight fixed support sets. Test scores average over the three seeds and 32 shared support sets. Unless stated otherwise, each support set has five patients per class and four windows per patient ($P = 5$, $K = 4$). The training/validation/test patients are 52/18/18 for ADFTD, 120/40/40 for PTB, and 2,402/799/799 for the single-label PTB-XL subset. The test sets contain 3,996, 3,870, and 3,995 windows. The single-label PTB-XL task uses the 4,000-patient subset of Appendix B.

The paired 95% confidence intervals use 2,000 bootstrap samples of patients, stratified by class. Each sample keeps every window of a selected patient, and all compared methods use the same samples. The intervals are conditional on the fitted models and support sets. The three seeds and 32 support sets do not add test patients. Except for the one Bonferroni-corrected interval stated below, all intervals in this appendix are unadjusted. The test patients were also used during model development, and the intervals do not account for this model selection.

The main per-channel ablation, the margin-input experiments, and the additional PTB refits are separate experiments with separately fitted models. For example, SIDP reaches 0.4258 on the single-label PTB-XL task below and 0.6215 on the separate multi-label task. On per-channel PTB, it reaches 0.8603 in the main ablation and 0.8518 in the additional refits. We only compare methods within the same experiment.

I.1 Per-channel descriptors on PTB

After channel-mean subtraction and SD normalization, each PTB channel gives its own 29 descriptors, which form a $15 \times 1 \times 29$ tensor. This input skips the ridge classifier, so SIDP receives the information of each channel directly. Every method receives the same tensor. The two MLPs flatten it, while SIDP keeps its channel structure. We retrain all learned models with the shared schedule. The canonical-pooling control replaces canonical attention with mean pooling, and keeps its value and output projections. The probe-pooling control replaces probe retrieval with mean pooling, and keeps its value projection.

SIDP has the highest mean window score among these methods. Its differences from the two MLPs have intervals that include zero. When we average the window probabilities of each patient, SIDP reaches **0.9661** patient macro-F1 and 0.9609 balanced accuracy. We chose mean-probability aggregation on validation data for all three SIDP seeds. At the patient level, SIDP is 5.74 points [0.25, 14.01] above the support-conditioned MLP.

Table 10: Main PTB ablation with per-channel descriptors (window macro-F1). Differences are SIDP minus each method. The intervals for fixed queries and both pooling controls exclude zero.

Method	Macro-F1	Difference	Paired 95% CI
Prototype classifier	.525756	+.334544	[+.290046, +.377236]
Ordinary MLP	.841309	+.018991	[−.027638, +.075042]
Support-conditioned MLP	.849621	+.010679	[−.028174, +.058779]
Fixed learned class queries	.838929	+.021370	[+.007524, +.036079]
Canonical mean pooling	.719024	+.141275	[+.059213, +.230872]
Probe mean pooling	.839108	+.021192	[+.009874, +.033051]
SIDP, per-channel descriptors	**.860300**	—	—

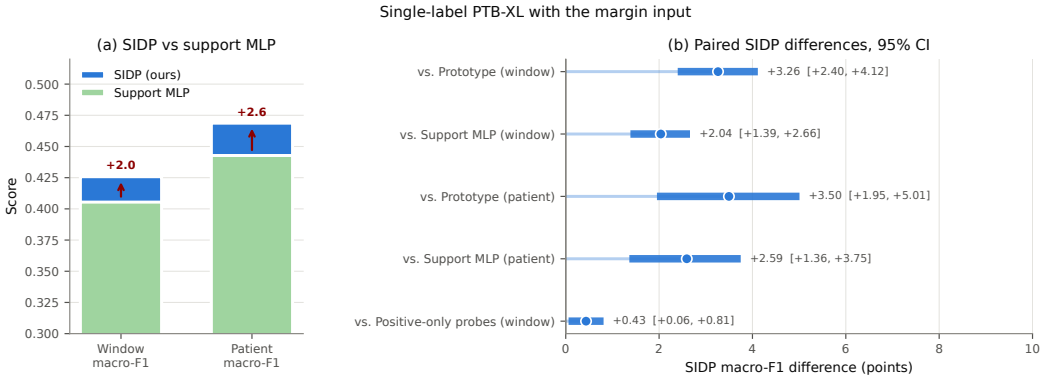


Figure 4: Single-label PTB-XL with the margin input. (a) SIDP and the support MLP (Table 12). (b) Paired SIDP differences from the support-based methods in macro-F1 points, with unadjusted 95% intervals (Table 13 and the positive-only variant).

Attention weights change with the recording. We fix the support set and pass one window from each of the 40 validation patients through each of the three trained models, with dropout disabled and float64 arithmetic. For an attention tensor $A^{(i)}$ from recording i, we report the range

$$\mathcal{R}(A) = \max_j \left(\max_i A_j^{(i)} - \min_i A_j^{(i)} \right), \tag{18}$$

where j indexes an attention weight. The weights of both attention stages change with the recording (Table 11). Recomputing the logits from these weights reproduces the model outputs up to numerical precision. Together with the pooling controls, this suggests that SIDP uses recording-dependent attention and benefits from learned retrieval with this PTB input. Because $L = 1$, the temporal attention has a single key, so only the canonical and probe attention can vary.

Table 11: Attention ranges of the three per-channel PTB models of Table 10. The last column is the maximum absolute error when the logits are recomputed from the attention weights.

Seed	Canonical $\mathcal{R}(A)$	Probe $\mathcal{R}(A)$	Max. logit error
1	.236101	.998976	3.553×10^{-15}
2	.244120	.998805	4.441×10^{-15}
3	.243913	.255484	1.055×10^{-15}

I.2 Support-based methods on single-label PTB-XL

Here, the descriptor and ridge encoder is fixed, and all methods receive the same margins, patients, and queries. The prototype classifier assigns the class of the nearest support centroid (Euclidean distance). The support-conditioned MLP is a shared nonlinear scorer of the query vector and the positive and negative support means. Table 12 reports all methods, including the ordinary MLP, and Figure 4 summarizes the comparisons.

Table 12: Single-label PTB-XL with the margin input. This custom task is separate from the multi-label evaluation. A dash marks a patient-level value that was not computed.

Method	Window macro-F1	Patient macro-F1
Prototype classifier	.393161	.433745
Support-conditioned MLP	.405423	.442773
Ordinary MLP	.435392	—
SIDP	.425784	.468705

Table 13: Paired SIDP differences from the two support-based methods on single-label PTB-XL. Window and patient scores are separate analyses.

Unit	Comparator	Difference	Paired 95% CI
Window	Prototype	+.032623	[+.023988, +.041154]
Window	Support-conditioned MLP	+.020361	[+.013856, +.026621]
Patient	Prototype	+.034959	[+.019536, +.050078]
Patient	Support-conditioned MLP	+.025931	[+.013624, +.037486]

At the window level, SIDP stays above the support-conditioned MLP after a Bonferroni correction over six comparisons (SIDP vs MLP and vs support MLP on each of the three datasets), with interval [+0.012061, +0.028760]. We do not apply this correction to the prototype or patient-level intervals in Table 13. However, the ordinary MLP scores higher than SIDP on this task. In the same corrected family, SIDP minus MLP is -0.009608, with interval $[-0.016272, -0.002960]$ (Table 22).

Negative support improves the probes. We retrain a positive-only variant, which removes the negative-support mean and the positive-minus-negative difference from probe generation. The input, training budget, validation selection, and support sets stay the same. This variant reaches 0.421436 window macro-F1, compared with 0.425784 for the full contrast. The gain is +0.004348, with paired interval [+0.000604, +0.008106]. This suggests that the negative support adds useful information on this task.

I.3 Additional PTB refits: metadata and support statistics

Here, we retrain the per-channel PTB models with the shared three-seed schedule, checkpoint selection on validation only, 32 support sets, and the same query patients. These refits ran on CPU and are separate from the main ablation (0.8603). The no-metadata variant removes the metadata term completely. As alternatives to the mean support summaries, we test coordinate-wise 20% trimmed means, and means with appended $\log(1 + \text{within-support variance})$ terms.

Table 14: Additional PTB refits with per-channel descriptors (window macro-F1). These results belong to the refitted SIDP (0.851765), and are separate from the main ablation (0.8603).

Method	Window macro-F1
Prototype	.525756
Ordinary MLP	.841309
Support-conditioned MLP	.849617
No metadata	.783113
Trimmed support means	.837167
Mean and variance support summaries	.833995
SIDP	**.851765**

Adding the metadata improves SIDP by 6.87 points, with an interval that excludes zero. The differences from the two MLPs have intervals that include zero. The mean summaries score higher than both alternatives, and both intervals exclude zero. These results favor the mean summaries on PTB. Figure 5a shows these differences.

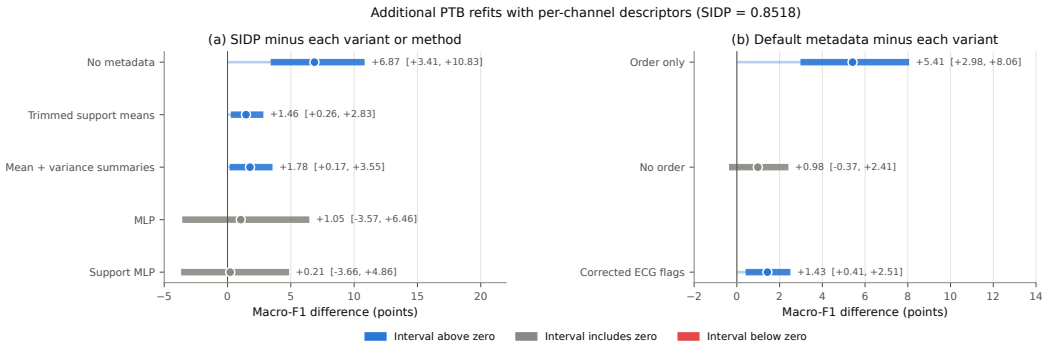


Figure 5: Additional PTB refits with per-channel descriptors, in macro-F1 points with unadjusted paired 95% intervals. (a) SIDP minus each variant or method (Table 15). (b) Default metadata minus each metadata variant (Table 18).

Table 15: Paired differences in the additional PTB refits. Each row gives the first method minus the second. All intervals are unadjusted.

Comparison	Difference	Paired 95% CI
SIDP − no metadata	+.068652	[+.034088, +.108303]
SIDP − MLP	+.010456	[−.035749, +.064630]
SIDP − support MLP	+.002148	[−.036631, +.048622]
Trimmed means − SIDP	−.014599	[−.028330, −.002596]
Trimmed means − MLP	−.004142	[−.050844, +.049472]
Trimmed means − support MLP	−.012450	[−.048836, +.030529]
Mean/variance − SIDP	−.017770	[−.035543, −.001693]
Mean/variance − MLP	−.007313	[−.055146, +.045392]
Mean/variance − support MLP	−.015622	[−.052808, +.027587]

Why averages of learned representations can be enough. SIDP averages learned, nonlinear latent slots of the support windows. Such averages can capture higher-order properties of the signal. For example, the mean of a squared input coordinate gives its second raw moment. The gains over trimmed means and appended variances support our choice of summary on PTB. They do not show that higher-order statistics are never useful. The exact structural property is that support sets with equal class counts and equal latent class means give equal predictions (Appendix C).

I.4 Separating the ridge and SIDP supervision

With the margin input, the ridge classifier is fitted on all training patients, so its margins for the training queries are in-sample. We test two PTB settings in which the ridge classifier never sees the labels of the training query patients. In four-fold patient cross-fitting, the query margins come from a ridge classifier fitted without those patients. In the split setting, one half of the training patients fits the ridge classifier, and the other half trains SIDP. Feature selection and scaling are fitted only on the patients that fit the ridge classifier. In cross-fitting, the following token normalizer is fitted on the pooled out-of-fold training margins. Both settings use three seeds, checkpoints selected on validation, and 32 shared test support sets. The split setting also leaves fewer patients for each fit.

In both settings, SIDP improves on both MLPs, and all four intervals exclude zero. However, all neural methods score lower than with the standard margin input (Table 21), and the direct and thresholded ridge classifiers score higher than SIDP in both settings (Table 16).

I.5 Metadata components and the Frank-lead correction

For each metadata variant, we retrain SIDP on the same per-channel descriptors with three seeds, 1,800 updates, and checkpoint selection on validation only. All variants keep the same metadata-network sizes. The order-only variant keeps only the acquisition order. The no-order variant keeps

Table 16: PTB methods when the ridge classifier is trained without the query patients (window macro-F1). Threshold denotes ridge with a threshold chosen on validation.

Method	Cross-fitted ridge	Separate-patient ridge
Ridge	.823456	**.830644**
Threshold	**.843678**	.830243
Prototype	.822553	.750179
Ordinary MLP	.797346	.751625
Support-conditioned MLP	.801230	.760592
SIDP	.805359	.768521

Table 17: Paired SIDP differences from the two MLPs when the ridge classifier is trained without the query patients. The intervals are unadjusted.

Setting	Comparator	Difference	Paired 95% CI
Cross-fitting	MLP	+.008013	[+.004158, +.012160]
Cross-fitting	Support MLP	+.004128	[+.002260, +.006113]
Separate patients	MLP	+.016896	[+.010401, +.023479]
Separate patients	Support MLP	+.007929	[+.004944, +.010932]

all other default fields. The corrected-ECG variant changes only the modality flags of the Frank leads `vx`, `vy`, and `vz`. Our default metadata mark these three leads as EEG channels. The name-derived odd/even indicators and the acquisition order are encodings of the channel names, and they do not contain measured electrode positions.

Table 18: PTB metadata variants in the additional refits. Differences are each variant minus the default metadata.

Metadata	Window macro-F1	Difference	Paired 95% CI
Default	.851765	—	—
Order only	.797616	−.054149	[−.080617, −.029761]
No order	.841945	−.009820	[−.024125, +.003678]
Corrected ECG flags	.837475	−.014290	[−.025061, −.004098]

The default metadata score 5.41 points higher than the acquisition order alone, which suggests that the other name-derived fields help. Removing the order gives an interval that includes zero. Correcting the Frank-lead flags lowers the score by 1.43 points in this refit (Figure 5b). These results concern our encoding of the channel names, and they do not show that the metadata capture anatomy. All reported results use the default metadata, and we report the corrected version only as a separate variant.

J Additional experiments across datasets and evaluation protocols

This appendix reports additional experiments on real recordings. Their patients, fitted models, and estimators are separate from the experiments in Appendix I. Single-label scores average 96 window macro-F1 values, each computed separately (three seeds and 32 shared support sets). The standard support sets ($P = 5$, $K = 4$) use 15, 10, and 25 training patients and 60, 40, and 100 windows for ADFTD, PTB, and the single-label PTB-XL subset. All models are first trained with full supervision. All tables use real recordings with their original labels.

The single-label schedule uses 1,800 AdamW updates, a learning rate of 10^{-3}, weight decay of 10^{-4}, gradient clipping at norm 5, and validation at 600, 1,200, and 1,800 updates over eight support sets. Within each setting, all methods use the same episode sampling. All methods use the same fixed learning rate and widths, and we did not tune them separately. We also did not log the exact number of query windows that each method saw during training. The MLPs flatten the channel

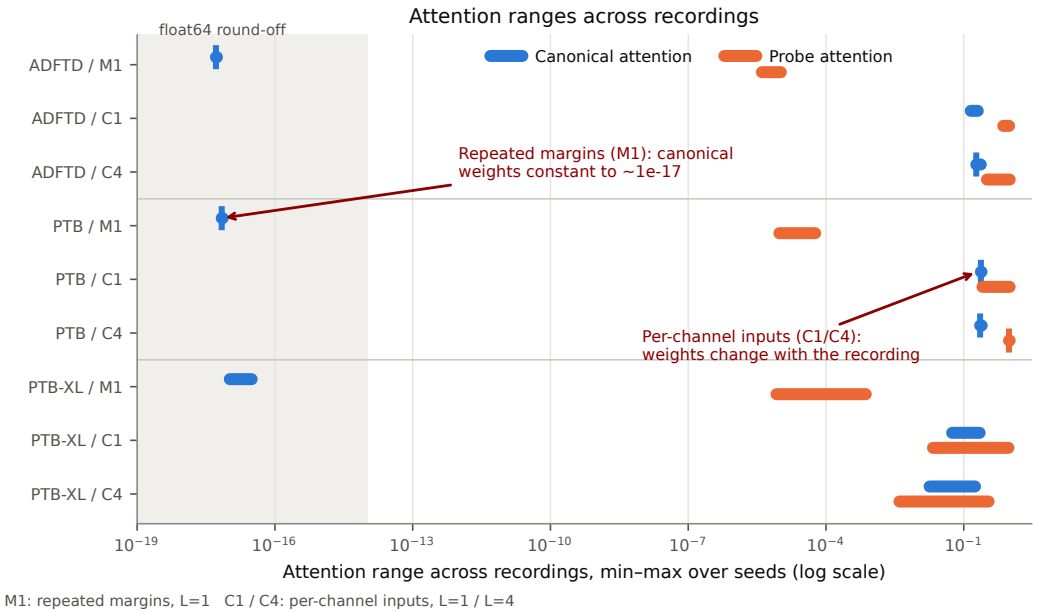


Figure 6: Attention ranges across recordings on a log scale (Table 20). M1: repeated margins. C1/C4: per-channel inputs with $L = 1$ and $L = 4$. The shaded region is below float64 round-off.

tensors in cache order, and the attention models also use the fixed metadata. Appendix J.4 describes the multi-label training.

The single-label confidence intervals use 2,000 patient resamples, stratified by class. Each resample keeps all windows of the selected patients, and all methods and runs use the same resamples. Except for the six-comparison correction stated below, the intervals are unadjusted. They are conditional on the fitted models and support sets, and on test patients that were evaluated during development. Methods that do not use the support set give the same predictions for every support set, and the prototype classifier has no training seed. We repeat their predictions only to align them with the other methods, so they do not count as extra independent fits.

A recomputation from the saved predictions found no score differences across 337 method/setting rows in 57 groups, and the prediction and checkpoint hashes matched.

J.1 Per-channel inputs across three datasets

With $L = 1$, the inputs before the fitted transforms are $19 \times 1 \times 29$, $15 \times 1 \times 29$, and $12 \times 1 \times 29$ for ADFTD, PTB, and PTB-XL. The PTB input includes its three Frank leads. Table 19 reports all seven methods with $L = 1$. The $L = 4$ variants give four temporal tokens per channel, and Table 20 includes their attention ranges. The mean-pooling controls keep the relevant value and output projections. Each control is a separately retrained model.

Table 19: Per-channel inputs with $L = 1$ (window macro-F1). PTB-XL is the single-label subset. All methods in a column receive the same input tensor.

Method	ADFTD	PTB	PTB-XL
SIDP	.413709	.860300	.153950
MLP	.525715	.841309	.424677
Support MLP	.510446	.849621	.362585
Prototype	.381025	.525756	.232871
Fixed learned queries	.410457	.838929	.391068
Canonical mean pooling	.447794	.719024	.145450
Probe mean pooling	.427593	.839108	.132336

On PTB, SIDP improves on the pooling controls and fixed learned queries, and these gains support the component results in Section 4.3. On the other two datasets, the results differ. On ADFTD, SIDP scores below the MLP (0.4137 vs 0.5257) and the support MLP (0.5104). On single-label PTB-XL, SIDP still scores above both pooling controls. However, it scores below the MLP (0.1540 vs 0.4247) and fixed learned queries (0.3911). Therefore, the component gains on PTB do not carry over to these per-channel inputs on ADFTD and PTB-XL.

Attention weights with fixed trained models. We pass one real validation window per patient through each model, with dropout disabled, three trained seeds, and one fixed validation support set. For PTB-XL, we use at most 128 patients. For each attention tensor, we compute the largest range of any weight across recordings. Table 20 gives the minimum and maximum of this range across seeds, and Figure 6 plots them. We also recompute the logits in float64. These ranges measure how much the attention changes with the recording, and they do not measure accuracy.

Table 20: Attention ranges for nine models. M1: repeated margins, $L = 1$. C1/C4: per-channel inputs, $L = 1$ and $L = 4$. Ranges are shown in scientific notation. The logit error is the maximum absolute error when the logits are recomputed.

Dataset / input	Canonical range	Probe range	Logit error
ADFTD / M1	5.2e-18–5.2e-18	4.02e-06–1.04e-05	1.78e-15
ADFTD / C1	.142–.202	.723–.973	2.89e-15
ADFTD / C4	.187–.235	.319–1	1.78e-15
PTB / M1	6.94e-18–6.94e-18	9.61e-06–5.82e-05	1.78e-15
PTB / C1	.236–.244	.255–.999	4.44e-15
PTB / C4	.227–.246	.966–1	3.55e-15
PTB-XL / M1	1.04e-17–3.12e-17	8.33e-06–.000736	1.78e-15
PTB-XL / C1	.0557–.222	.0213–.939	1.33e-15
PTB-XL / C4	.018–.176	.00397–.347	4.44e-16

With repeated margins, the canonical attention weights are constant up to float64 precision. This is consistent with every channel receiving the same signal, which cancels in the softmax. The canonical output includes a LayerNorm, so the probe attention weights and the logits still change slightly with the recording. With per-channel inputs, the weights of both stages change with the recording. With $L = 1$, the model cannot localize information in time, and these ranges do not show clinical localization.

J.2 All margin-input methods

Table 21 compares all methods on the same margin input within each dataset. On binary PTB, this input comes from a single margin r, expanded to $[-r, r]$. The ridge threshold chosen on validation is -0.2063476145. SIDP minus the thresholded ridge is -0.002711, with interval $[-0.013981, +0.010812]$.

Table 21: All twelve methods with the margin input (window macro-F1). A dash marks an entry that was not computed or does not apply. These fits are separate from the other settings.

Method	ADFTD	PTB	PTB-XL
Direct ridge	.502354	.823456	.434228
Thresholded ridge	—	.843678	—
MLP	.508037	.837927	.435392
Support MLP	.502739	.837117	.405423
Prototype	.499025	.822553	.393161
Fixed learned queries	.504185	.838116	.438699
Canonical mean pooling	.504071	.840885	.425579
Probe mean pooling	.504309	.841006	.426006
Positive-only probes	.505141	.838091	.421436
One probe	.504965	.839035	.424931
One canonical slot	.504580	.839070	.428421
SIDP	.504316	.840967	.425784

Table 22: SIDP minus MLP and support MLP (S-MLP) with the margin input. The last column gives the Bonferroni-corrected interval over these six comparisons.

Dataset	Control	Difference	Unadjusted 95% CI	Adjusted interval
ADFTD	MLP	-.003721	[-.007735, +.001032]	[-.008651, +.003392]
ADFTD	S-MLP	+.001577	[-.004944, +.009319]	[-.006731, +.011878]
PTB	MLP	+.003040	[-.004241, +.009278]	[-.006358, +.011990]
PTB	S-MLP	+.003851	[-.004321, +.010993]	[-.006443, +.013798]
PTB-XL	MLP	-.009608	[-.014600, -.004886]	[-.016272, -.002960]
PTB-XL	S-MLP	+.020361	[+.013856, +.026621]	[+.012061, +.028760]

With the margin input, all methods score within 0.9 points of each other on ADFTD. On PTB, all methods except the prototype and the direct ridge classifier score within 0.7 points. On PTB-XL, several methods score higher than SIDP, including fixed learned queries (0.4387), the MLP (0.4354), and the direct ridge classifier (0.4342).

The fixed-query model is a learned-query control, and it does not reproduce DETR. We did not run separately tuned logistic regression, larger MLP searches, or verified implementations of Set Transformer and dual prototypes.

J.3 Normalization statistics

These refits use a separate set of aligned one-second windows, which is the same for all three normalization settings. We read the raw recordings before any patient normalization. The training-set setting uses statistics from the training windows, the window setting uses only the current window, and the patient setting uses the patient's retained recording (joining retained records where there are several). We add 10^{-9} to each SD before dividing. Spatial mean-plus-SD normalization follows each setting, and we refit the features, encoder, and classifiers for each one. This comparison does not test whether the raw filtering and resampling are causal.

All four intervals include zero. Window statistics give the highest SIDP score on PTB.

Spatial preprocessing. We also refit the feature transforms, the ridge classifier, and all classifiers for three spatial preprocessing settings on top of the existing loader: the loader preprocessing alone, channel-mean subtraction, and mean subtraction with SD normalization (Table 26 and Figure 3).

Table 23: Refits with different normalization statistics on aligned one-second windows (window macro-F1). Neural methods.

Dataset / statistics	SIDP	MLP	Support MLP
ADFTD / Patient	.501387	.505990	.495815
ADFTD / Window	.486062	.486158	.486745
ADFTD / Training set	.488705	.491660	.486407
PTB / Patient	.829900	.822399	.822967
PTB / Window	.874812	.878776	.878650
PTB / Training set	.831007	.829228	.829152

Table 24: Refits with different normalization statistics on aligned one-second windows (window macro-F1). Other methods.

Dataset / statistics	Direct ridge	Thresholded ridge	Prototype
ADFTD / Patient	.501509	—	.499113
ADFTD / Window	.487222	—	.479086
ADFTD / Training set	.497786	—	.488617
PTB / Patient	.807345	.829683	.805043
PTB / Window	.858845	.864442	.859641
PTB / Training set	.816878	.834589	.815764

Table 25: SIDP differences from normalization with patient statistics.

Dataset	Statistics	Difference	Paired 95% CI
ADFTD	Window	-.015325	[-.039970, +.008309]
ADFTD	Training set	-.012682	[-.044584, +.020743]
PTB	Window	+.044912	[-.039619, +.126073]
PTB	Training set	+.001108	[-.045947, +.048444]

Table 26: Spatial preprocessing with refitted models (window macro-F1).

Dataset / preprocessing	SIDP	MLP	Support MLP	Direct ridge	Prototype
ADFTD / Loader only	.430825	.435299	.434274	.436910	.435039
ADFTD / Mean	.524185	.519643	.517269	.521836	.513972
ADFTD / Mean + SD	.504316	.508037	.502739	.502354	.499025
PTB / Loader only	.750746	.752264	.751204	.744246	.743485
PTB / Mean	.868616	.866302	.864348	.846972	.846386
PTB / Mean + SD	.840943	.837927	.837117	.823456	.822553

Signal units and sampling. For ADFTD, we use the released EEG derivatives, resampled by MNE to 256 Hz, with primary windows of 256 samples. For PTB, we use the WFDB physical samples, decimated from 1,000 to 250 Hz with a zero-phase FIR filter, with primary windows of 250 samples. For single-label PTB-XL, we use the distributed 100 Hz waveforms with windows of 128 samples. EEG is in volts, and ECG is in mV. In the main ADFTD and PTB experiments, channel standardization uses the patient's retained windows after the per-patient cap. The normalization refits use the larger recording context. The spatial normalization uses the population SD with the floor $\max(10^{-6}\,\mathrm{RMS}_t(\sigma_t), 10^{-12})$.

J.4 MULTI-LABEL PTB-XL ON THE STANDARD SPLIT

This experiment uses PTB-XL v1.0.3 with complete ten-second records at 100 Hz. We use the standard patient-disjoint split recommended by the dataset authors, with folds 1–8 for training, fold 9 for validation, and fold 10 for testing. We keep all five mapped diagnostic superclasses (NORM, MI, STTC, CD, and HYP), including co-occurring labels. We exclude records without a mapped superclass. This leaves 21,388 eligible records and excludes 411 of the 21,799 metadata rows. The waveform and header checksums agree with v1.0.3.

Table 27: Multi-label cohort after keeping records with a mapped diagnostic superclass.

Split	Records	Patients
Training	17084	14823
Validation	2146	1917
Test	2158	1877

Descriptor tokens. The cached tensor has shape $21{,}388 \times 12 \times 4 \times 50$. We divide each ten-second lead into four non-overlapping 2.5-second segments. The 29 global descriptors are computed over the full record and repeated in all four tokens. Each token adds 20 local non-spectral descriptors (the 29 descriptors without the nine spectral ones) and the position of its segment center, which is one of $\{0.125, 0.375, 0.625, 0.875\}$. The initial construction has 51 features, including a duration fraction that always equals 0.25. The training transform removes this constant feature, which leaves 50. The signed-log and median/IQR scaling, with an SD fallback, are fitted on 100,000 sampled training tokens. The 14 metadata features enter separately.

Spatial channel-mean subtraction and SD normalization are applied to the full ten-second record. The token descriptors go directly into SIDP, without the ridge classifier. Independent sigmoid outputs with binary cross-entropy replace the single-label softmax, and each label has its own positive and negative support summaries, as described below. Table 28 lists the layer widths. This model has 82,946 parameters, compared with 77,186 for the single-label margin model.

Table 28: Token and network sizes for multi-label PTB-XL. The metadata are separate from the 50 descriptor features.

Component	Dimensions or configuration
Descriptor input	$12 \times 4 \times 50$ per record
Temporal key/value projection	$50 \to 64$
Temporal / canonical / probe slots	$R_t = 4$, $R_c = 8$, $R_p = 4$
Canonical attention	Four heads; latent width 64
Metadata network	$14 \to 64 \to 64$
Class-contrast network	$192 \to 64 \to 64$
Probe network	$64 \to 64 \to 64$
Query/key/value projections	$64 \to 64$
Interaction scorer	$256 \to 64 \to 1$

The linear, MLP, support MLP, and prototype baselines average over the leads and segments before classification. Therefore, the comparisons between SIDP and the pooled baselines measure the combined effect of keeping the token structure and of the classifier.

Support sets and training. For diagnosis k, a record is positive if $y_k = 1$ and negative if $y_k = 0$. Records that are positive for another diagnosis but not for k are therefore negative for k, and records positive for k are never in its negative set. Our sampler draws five positive and five negative patients per diagnosis. It then removes duplicate records and recomputes the memberships from each record's full label vector. The 32 saved evaluation support sets each contain 49–50 distinct records and patients, with no repeated patient within a support set. For training and validation draws, the sampler does not guarantee that a patient appears for only one diagnosis. Query patients never appear in the support set. Table 29 gives the actual counts per label, since these support sets do not follow the $P = 5$, $K = 4$ design.

Table 29: Positive and negative record counts in the 32 fixed evaluation support sets.

Diagnosis	Positive records	Negative records
NORM	13–20	30–37
MI	10–20	30–40
STTC	11–23	27–39
CD	10–18	32–40
HYP	7–15	35–43

The neural models use sigmoid outputs, binary cross-entropy, 50 epochs, a batch size of 64, AdamW with a learning rate of 10^{-3}, and weight decay of 10^{-4}. We select the checkpoint with the highest mean validation macro-AUROC over eight support sets. Each neural seed processes 854,200 training query records during training. For each label, we choose the threshold that maximizes the validation F1 of the positive class on the probabilities averaged over support sets. Ties go to the value closest to 0.5, and a record is positive if $p > \theta_k$. The prototype classifier has no learned head. All methods use the same training labels and the same fixed evaluation support sets.

Metrics and intervals. We compute each metric per run and average over seeds and support sets. Macro-AUROC averages the ROC areas of the labels, and AUPRC is the non-interpolated average precision. Labels stay record-specific, even for patients with several records. The bootstrap resamples patients and keeps all their records and co-occurring labels, and it averages the F1 of each run. If a label has no true and no predicted positives, its F1 is set to zero. We report 2,000-replicate intervals for F1 only, and we did not compute paired intervals for AUROC or average precision, or apply a multiple-comparison correction. The label mapping and the macro-AUROC follow the benchmark utilities, and we do not reproduce the full set of benchmark metrics.

Table 30: All five multi-label methods on the standard test split. AP is average precision. The intervals are for macro-F1 and resample patients.

Method	AUROC	AP	Macro-F1	F1 95% CI
Linear	.735092	.489673	.507196	[+.490454, +.523691]
MLP	.779083	.548265	.546471	[+.530574, +.561493]
Prototype	.532700	.289355	.363930	[+.356003, +.371857]
SIDP	.845022	.656212	.621494	[+.606006, +.635948]
Support MLP	.757432	.505705	.521816	[+.506327, +.535636]

Table 31: Multi-label PTB-XL: SIDP minus each method. Paired macro-F1 differences with patient resampling, without multiple-comparison correction.

Comparator	Difference	Paired 95% CI
Linear	+.114298	[+.100324, +.127944]
MLP	+.075023	[+.063326, +.087448]
Prototype	+.257564	[+.243197, +.270410]
Support MLP	+.099678	[+.088174, +.110997]

SIDP's gains over the pooled baselines reflect the complete pipeline, and Figure 7 summarizes them. Figure 8 shows all metrics for SIDP and the two MLPs, including the overlap groups of Table 33. Section 4.3 tests the individual components on PTB.

Table 32 and Figure 9 give SIDP's results for each diagnostic superclass.

Some test patients are also in the single-label subset, which we used during model development. In both groups, SIDP scores above the MLP and support MLP. Patients in the no-overlap group may still have been examined in other development experiments. We did not compute paired intervals for these groups.

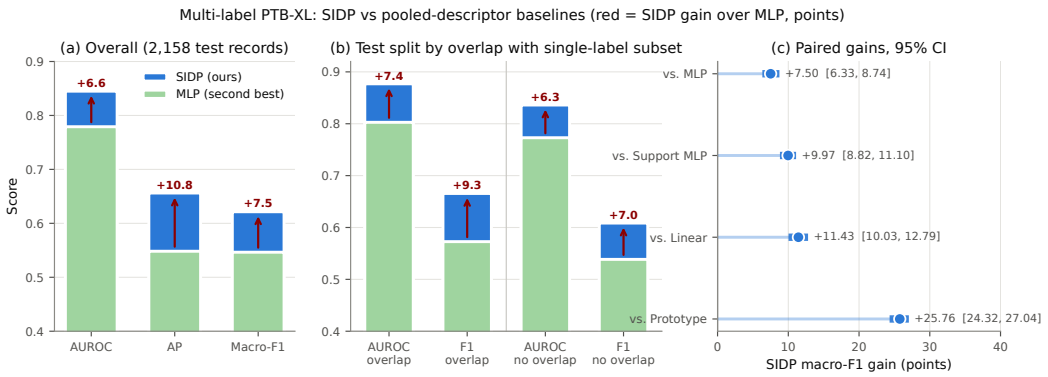


Figure 7: Multi-label PTB-XL. (a) SIDP and the MLP on the full test split. (b) The same comparison on test patients with and without overlap with the single-label subset (Table 33). (c) SIDP minus each pooled-descriptor baseline in macro-F1 points, with paired 95% intervals (Table 31).

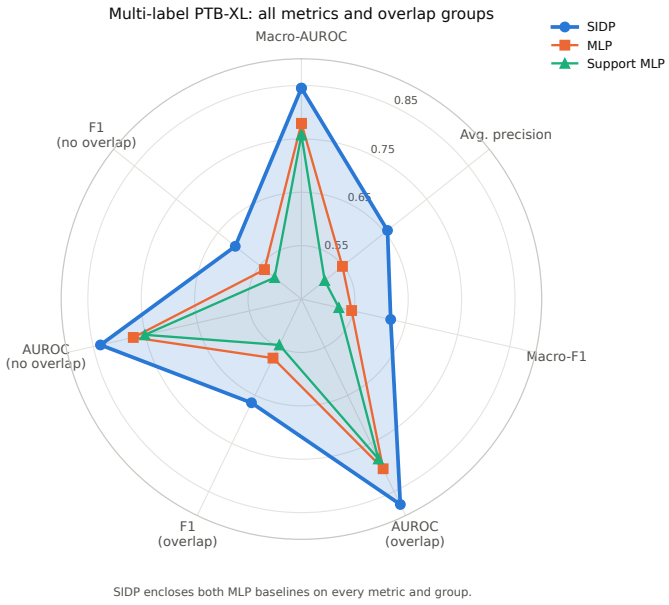


Figure 8: Multi-label PTB-XL: macro-AUROC, average precision, and macro-F1 for SIDP, the MLP, and the support MLP, on the full test split and on the overlap groups of Table 33.

Table 32: SIDP results for each diagnostic superclass on multi-label PTB-XL.

Diagnosis	AUROC	Average precision	F1
NORM	.899934	.858306	.807899
MI	.831070	.626390	.613832
STTC	.871586	.681645	.646758
CD	.825787	.680340	.596890
HYP	.796731	.434381	.442092

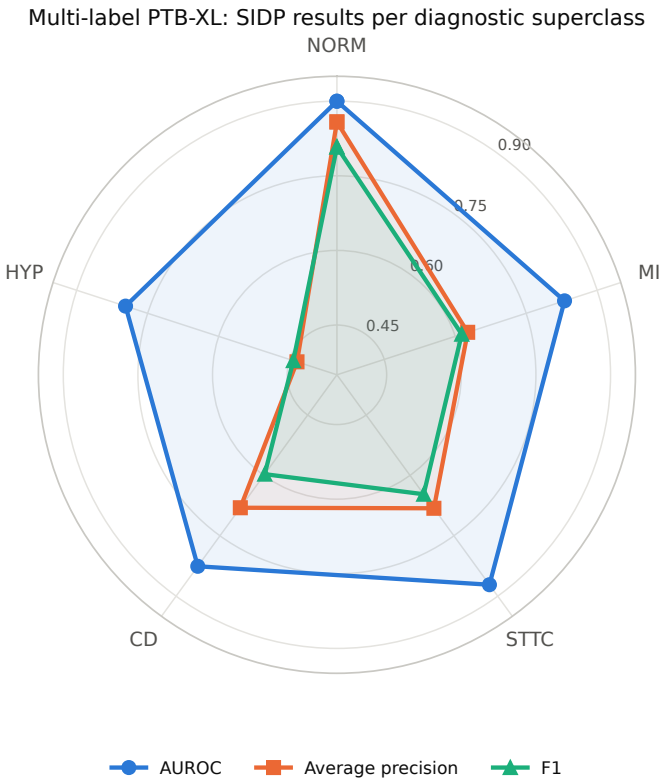


Figure 9: Multi-label PTB-XL: SIDP's AUROC, average precision, and F1 for each diagnostic superclass (Table 32).

Table 33: Test split divided by overlap with the single-label PTB-XL subset. Overlap: 419 patients (481 records) that are also in the 4,000-patient single-label subset. No overlap: 1,458 patients (1,677 records) that are not.

Group	Method	AUROC	Macro-F1
Overlap	SIDP	.876984	.665591
Overlap	MLP	.802747	.572714
Overlap	Support MLP	.782300	.545354
No overlap	SIDP	.836107	.608531
No overlap	MLP	.773021	.538523
No overlap	Support MLP	.750785	.514452

Table 34: Training resources on multi-label PTB-XL. Seconds sum training and validation over three seeds, and they are not normalized across hardware. Epochs are listed for seeds 1/2/3, and the SD is over the seed-mean AUROC.

Method	Parameters	Selected epochs	Fit seconds	AUROC SD
Linear	255	50, 50, 48	604.1	.000353
MLP	7749	48, 42, 49	643.6	.000653
Prototype	0	1, 1, 1	0.7	.000000
SIDP	82946	24, 40, 42	1769.9	.001481
Support MLP	13889	45, 50, 49	698.5	.001361

Limitations of the evaluation. We did not run controls with equal input information or more strongly tuned baselines. We also did not evaluate transfer to a new cohort without training, SIDP on frozen CBraMod features, or support sets with natural artifacts.