SCoPe: Semantic Context-Adaptive Prototypical Learning for Clinical Outcome Prediction

Anonymous ACL submission

Abstract

Clinical outcome prediction from lengthy clinical documents is challenging because task-relevant information is sparsely distributed across thousands of tokens. Existing methods often rely on predefined intermediate units, such as fixed-length chunks or n-gram spans. However, their forced boundaries may cut through clinically meaningful expressions or separate locally dependent tokens, causing a mismatch between predefined segments and the underlying semantic boundaries. We propose Semantic Context-Adaptive Unit Prototypical Learning (SCoPe), a framework that adaptively groups adjacent tokens into variable-length phrase units based on local contextual continuity and performs prototype-based classification at this intermediate granularity. We further introduce a Distributional Alignment Regularization objective that aligns distributions induced from class-conditional prototype similarities with Gaussian priors, stabilizing prototype learning under highly imbalanced settings. Experiments on two MIMIC benchmarks show that SCoPe consistently outperforms competitive baselines. [1]

1 Introduction

In clinical practice, clinical notes capture a patient's overall hospital course, from admission through treatment to discharge. Automated analysis of such records for tasks including mortality prediction and patient phenotyping has been an active area in clinical natural language processing domain for a long time (Nye et al., 2018; Johnson et al., 2016, 2023; Huang et al., 2019; Mullenbach et al., 2018; Naik et al., 2022; Alsentzer et al., 2019; Chen et al., 2020; Yang et al., 2024, 2022). These documents are often lengthy, with task-relevant information unevenly distributed among extensive routine documentation. To address this, recent works have converged on constructing intermediate-granularity representations between tokens and the document. This has motivated research in this domain to move beyond purely token-level or document-level representations and model clinical documents at intermediate granularities.

Existing methods instantiate this idea at different levels, including fixed-length chunks (Pappagari et al., 2019; Chalkidis et al., 2022; Dai et al., 2022; Li et al., 2025), document sections (Gong et al., 2025), sentences (Ding et al., 2020; Chen et al., 2020), n-gram spans (Wei and Zhu, 2025), and individual tokens (Van Aken et al., 2022). These intermediate representations have been shown to improve classification performance in clinical settings.

Despite their differences in granularity and aggregation strategy, these methods share a common limitation: the level of granularity at which they operate is determined by predefined rules and remains fixed regardless of the input. Basic semantic units such as sentences and multi-token clinical concepts (e.g., "systolic heart failure") vary considerably in length, yet none of these approaches can adapt their unit boundaries to such variability. Fixed boundaries may split coherent semantic units and disrupt their contextual continuity. A natural next step is thus to allow the granularity to adapt to the local semantic structure of the text, addressing contextual adaptiveness.

To address these challenges, we propose Semantic Context-Adaptive Unit Prototypical Learning (SCoPe), an end-to-end framework for clinical outcome prediction from long documents. SCoPe first constructs variable-length semantic units by adaptively identifying token-level semantic boundaries based on local contextual adjacency. It then performs phrase-level prototype-based classification, where semantic unit representations are matched against class-specific prototype patterns. To stabilize prototype learning, we further introduce a dis-

[1] Code available at https://anonymous.4open.science/r/clinical_diffusion_generator-1F8B/

tributional regularization loss that aligns prototype-level similarity representations with Gaussian priors.

Extensive experiments on clinical outcome prediction tasks derived from MIMIC (Johnson et al., 2016, 2023) demonstrate that SCoPe achieves strong performance on challenging, highly imbalanced clinical benchmarks; further evaluation on Hyperpartisan News Detection (Kiesel et al., 2019) confirms that the approach generalizes effectively to non-clinical long-document classification as well, consistently outperforming competitive baselines across both settings, while producing interpretable semantic boundaries that offer a complementary perspective to existing interpretability approaches.

2 Related Work

2.1 Long-Document Representation

Adapting Transformer-based models to long-document processing has been approached from multiple angles, ranging from modifications to the attention mechanism (Vaswani et al., 2017) itself to post-encoding strategies that construct intermediate-grandulaity representations from the encoded tokens. On the encoder side, Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020) extend the feasible input length to 4,096 tokens through sparse attention mechanism, while ModernBERT (Warner et al., 2025) further supports sequences of up to 8,192 tokens. Clinical ModernBERT (Lee et al., 2025) and BioClinical ModernBERT (Sounack et al., 2025) adapt these long-context encoders to the biomedical and clinical domain through continued pretraining on domain-specific corpora. On the post-encoding side, a dominant strategy is hierarchical modeling, in which a document is split into chunks that are encoded independently and then aggregated for document-level prediction. Methods in this family differ primarily in their aggregation mechanism: RoBERT and ToBERT (Pappagari et al., 2019) employ a Long-Short-Term Memory network (Graves, 2012) and a Transformer encoder, respectively; ChunkBERT (Zheng et al., 2023) combines BERT-based phrase representations with a graph-based ranking algorithm; HAT (Chalkidis et al., 2022) introduces hierarchical attention over chunk representations; and (Dai et al., 2022) encodes each segment with a shared RoBERTa(Liu et al., 2019) encoder, models cross-segment interactions with segment-level Transformer layers, using overlapping segments to reduce context fragmentation. More recently, ConceptMoE (Huang et al., 2026) departs from chunk-based segmentation by compressing token-level representations into phrase-level representations, thereby reducing the number of units involved in self-attention module.

2.2 Prototype-based Classification

Prototypical learning was popularized by Prototypical Networks, which learn an embedding space where each class is represented by the centroid of its support examples, enabling few-shot classification via nearest-prototype matching(Snell et al., 2017). This paradigm naturally provides interpretability as a byproduct, since predictions can be traced to specific prototype similarities. In the vision domain, (Yang et al., 2018) establishes class-specific convolutional prototype learning for robust classification. In NLP, ProtoryNet (Hong et al., 2023) models prototype trajectories to capture sequential reasoning patterns over text. More recently, ProtoLens (Wei and Zhu, 2025) advances prototype-based text classification by using a Dirichlet Process Gaussian Mixture Model to model n-gram-level prototype similarities, enabling prototype-specific sub-sentence span extraction with semantically aligned prototypes. In the clinical domain, ProtoPatient (Van Aken et al., 2022) combines prototypical networks with label-wise attention to match patient representations against label-wise prototypes at the token level.

While these methods demonstrate the effectiveness of prototype-based classification across different matching granularities, they are designed for standard-length inputs and have not been evaluated on long-document benchmarks where the length of documents far exceeds the token limits of transformer-based models.

3 Methodology

3.1 Problem Formulation

Given a document represented as a sequence of L tokens, we first obtain token representations from a pretrained encoder:

$$H = [h_1, h_2, \ldots, h_L], \quad h_i \in R^D \qquad (1)$$

where D denotes the hidden dimension of the encoder, and h_i is the representation of the i-th token. The goal is to predict a binary label $y \in \{0, 1\}$ for

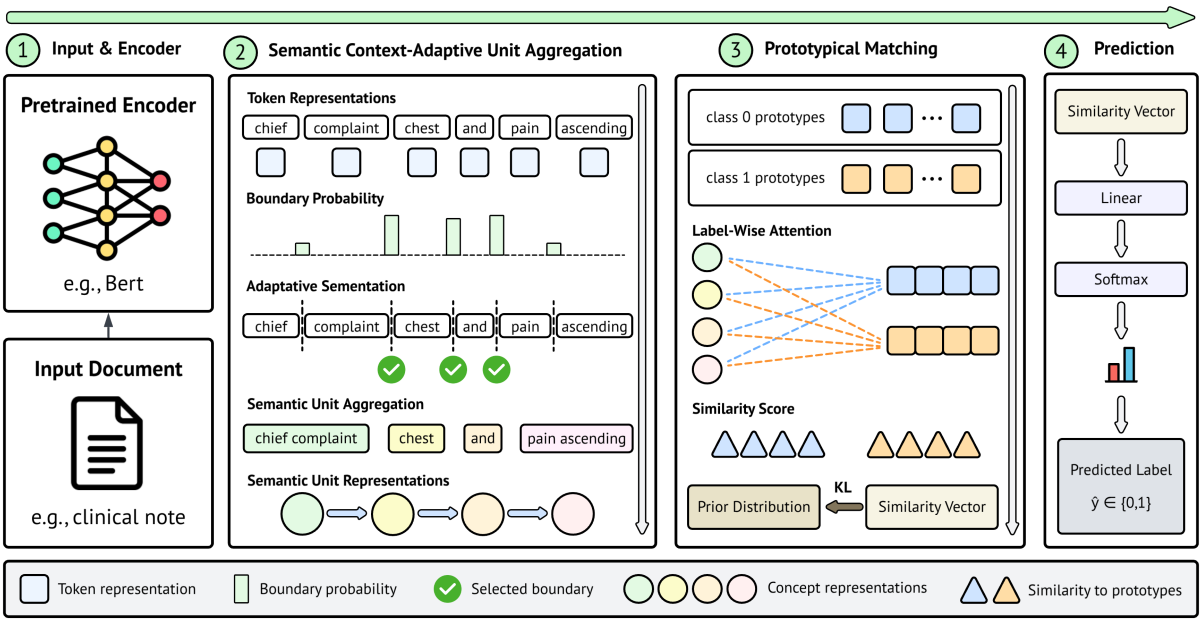

Figure 1: Overview of our general processing pipeline. The initial input, the sequence of token embeddings, is extracted from the pre-trained encoder. The sequence will be input to the concept-merging module to be adaptively aggregated into local semantic unit representations. The sequence of the semantic unit representation will calculate the distance between prototypes and each semantic unit. The final similarity vector will pass a linear layer to get the probability for each of the classes.

the document. As shown in Figure 1, our framework decomposes this task into three components: (1) a *pretrained bert-based encoder* that produces token-level representations H from the input document; (2) a *Semantic Context-adaptive Unit Aggregation Module* that partitions H into a sequence of variable-length semantic unit representations:

$$C = [c_1, c_2, \ldots, c_T], \quad T \ll L \tag{2}$$

where T is the number of resulting semantic units and $c_t \in R^D$ is the representation of the t-th semantic unit; and (3) a *Prototype-based Classifier* that produces a prediction based on similarities between semantic unit representations and learnable class-specific prototype sets for each class.

3.2 Semantic Context-Adaptive Unit Aggregation Module

The Semantic Context-Adaptive Unit Aggregation Module compresses token-level representations into variable-length phrase units. Inspired by token-to-concept compression (Huang et al., 2026), our approach uses local semantic relations encoded by the pretrained model to adaptively determine token grouping, thereby shifting downstream classification from isolated tokens to phrase-level representations. With the token representation h_i from a clinical document, we compute an adjacency score for each neighboring token pair $(i, i+1)$. Specifically, we apply two learnable projection layers, W_k and W_q, where the former token is mapped to a key representation, and the latter token is mapped to a query representation. We then use the cosine similarity between the projected representations, k_i and q_{i+1}, as the adjacency score for the token pair, as shown in Equation 3.

$$\begin{aligned} k_i &= W_k h_i, \qquad q_{i+1} = W_q h_{i+1} \\ s_i &= \cos(k_i, q_{i+1}) = \frac{k_i^\top q_{i+1}}{\|k_i\|_2 \|q_{i+1}\|_2}, \quad i = 1, \ldots, L-1 \end{aligned} \tag{3}$$

As a result, this stage produces a sequence of pairwise local similarity scores that characterize how strongly each token connects to its immediate successor in the projected representation space.

We convert the cosine similarity into an initial boundary probability through a linear mapping:

$$p_i = \frac{1 - s_i}{2}, \quad i = 1, \ldots, L-1 \tag{4}$$

With continuous boundary probabilities, the module applies a sharpening transformation to encourage more discrete boundary decisions. Let $p_i \in [0, 1]$ denote the boundary probability at position i. We define the sharpened probability $\hat{p}_i$

as

$$\hat{p}_i = \begin{cases} p_i^{1/\alpha}, & p_i > 0.5, \\ 1 - (1 - p_i)^{1/\alpha}, & p_i \leq 0.5, \end{cases} \tag{5}$$

where $\alpha > 1$ is the sharpening rate. This transformation amplifies the contrast between high and low boundary probabilities, sharpening soft predictions into near-binary boundary decisions.

Finally, we sample a binary boundary indicator b_i from the corresponding Bernoulli distribution,

$$b_i \sim \text{Bernoulli}(\hat{p}_i), \qquad b_i \in \{0, 1\}, \tag{6}$$

where $b_i = 1$ indicates that position i is selected as a boundary.

To characterize the relationship between the continuous boundary probabilities and the discretely sampled boundary decisions at the global level, we first compute the number of valid content positions,

$$N = \sum_{i=1}^{T} m_i, \;\; m_i \in \{0, 1\} \tag{7}$$

where m_i denotes whether the semantic unit i exists. Based on the continuous boundary probability p_i, we then define the global expected boundary as

$$G = \frac{1}{N} \sum_{i=1}^{T} p_i m_i, \tag{8}$$

which reflects the model's overall expectation of the proportion of boundary positions. Furthermore, based on the sampled binary boundary mask b_i, we define the empirical boundary frequency as

$$\text{Freq} = \frac{1}{N} \sum_{i=1}^{T} b_i m_i, \tag{9}$$

which represents the proportion of positions that are hard-sampled as boundaries among all valid positions. Finally, we construct an auxiliary loss

$$\mathcal{L}_{\text{aux}} = \frac{R}{R-1} \left((R-1)\,\text{Freq}\, G + (1 - \text{Freq})(1 - G) \right), \tag{10}$$

to measure the degree of alignment between the global expected boundary rate and the empirical sampled boundary frequency, where R controls the relative weighting between the boundary term and the non-boundary term.

Tokens between two consecutive sampled boundaries are merged into a single semantic unit, then mean pooling is applied over their representations to obtain the aggregated unit representation.

After obtaining the phrase-level representations, we further apply Exponential Moving Average (EMA) smoothing (Tarvainen and Valpola, 2017) over the semantic units sequence to restore continuity between adjacent semantic unit. Let $\{c_t\}_{t=1}^{T}$ denote the original semantic unit representations. The forward EMA recursively aggregates the current semantic unit with the previous smoothed state:

$$c_t = \gamma c_t + (1 - \gamma) c_{t-1}, \qquad t = 1, \ldots, T-1, \tag{11}$$

where γ is the EMA coefficient controls the weight of information transported from the previous semantic unit to the next.

3.3 Balanced Prior Aligned Prototypical Learning

3.3.1 Prototypical Matching

After phrase-level modeling, the classifier needs to organize semantic unit representations into class-specific decision spaces. This requires modeling not only the global distinction between classes, but also the diverse semantic distributions that may appear within each class.

To address this need, we adopt a prototype-based classifier. Each class c is associated with a learnable prototype set $\mathcal{P}_c = \{p_j^{(c)}\}_{j=1}^{K_c}$, where K_c denotes the number of prototypes of each class. Each prototype serves as a class-specific reference pattern, allowing the model to compare the input against multiple structured semantic patterns within each class. The prototypes are initialized using K-means++(Arthur et al., 2007) on the `[CLS]` embeddings of training instances from the corresponding class and are updated during training. The initialization details are provided in Appendix C.2.

To construct class-specific semantic anchors, the model averages the projected prototypes within each class-specific prototype set:

$$a_c = \frac{1}{N_c} \sum_{j=1}^{N_c} W_{\text{proj}} \left(p_j^{(c)} \right), \tag{12}$$

where W_{proj} denotes a learnable projection layer. In this way, each class is represented by a prototype-derived semantic anchor, which serves as a central reference direction estimated from the class-specific prototype set.

With the class-specific semantic anchors, we further compute class-conditional attention over the

semantic representations to identify which semantic units are most relevant to each class. Specifically, given an input sequence of representations $\{c_t\}_{t=1}^{T}$ and the class-specific semantic anchor a_c for class c, we first compute the matching score between semantic unit t and class c using a dot product:

$$e_{c,t} = c_t^{\top} a_c. \tag{13}$$

We then normalize these scores over the semantic unit dimension for each class independently, using a temperature-scaled softmax to obtain the class-specific attention weights:

$$\alpha_{c,t} = \frac{\exp(e_{c,t}/\tau)}{\sum_{t'=1}^{T} \exp(e_{c,t'}/\tau)}, \tag{14}$$

where τ denotes the attention temperature.

Based on the resulting class-conditional attention distribution, we aggregate the input representations into a class-specific weighted representation:

$$z_c = \sum_{t=1}^{T} \alpha_{c,t} c_t. \tag{15}$$

In this way, the model produces not only an aggregated representation z_c for each class, but also an explicit phrase-level attention distribution $\alpha_{c,t}$ that indicates how strongly each semantic unit contributes to the representation of that class. This design enables the model to highlight class-relevant evidence at the phrase level, thereby improving the interpretability of the prediction process.

3.3.2 Distributional Alignment Regularization

After obtaining the class-conditional aggregated representations, we further match each of them against the prototype set of the corresponding class.

We define the class-specific prototype similarity vector $s^{(c)} \in R^{K_c}$, using negative Euclidean distance, whose j-th element is computed as

$$s_j^{(c)} = -\left\| z_c - p_j^{(c)} \right\|_2, \qquad j = 1, \ldots, K_c. \tag{16}$$

Equivalently,

$$s^{(c)} = \left[s_1^{(c)}, \ldots, s_{K_c}^{(c)} \right]. \tag{17}$$

Under this definition, a smaller Euclidean distance corresponds to a higher similarity score. Through this class-specific prototype matching process, the model produces a set of prototype-level similarity scores that quantify how closely the input aligns with different learned prototype representations within each class.

With the prototype similarity scores for the two classes, we introduce a class-conditional Kullback-Leibler (KL) divergence (Kingma and Welling, 2013) regularization loss over a latent prototype-level variable inferred from the prototype similarity vector. For the KL regularization loss, we use mini-batch notation. Let $\mathcal{B}_c$ denotes the subset of samples in the mini-batch whose label is c. For each sample $i \in \mathcal{B}_c$, let $s_i^{(c)} \in R^{K_c}$ denote its class-specific prototype similarity vector. We stack these vectors into

$$S^{(c)} = \left[s_i^{(c)} \right]_{i \in \mathcal{B}_c} \in R^{|\mathcal{B}_c| \times K_c}, \tag{18}$$

We estimate the variational distribution parameters as

$$\mu^{(c)} = f_\mu\left(S^{(c)}\right), \qquad \sigma^{(c)} = \mathrm{softplus}\left(f_\sigma\left(S^{(c)}\right)\right) + \epsilon, \tag{19}$$

where $\mu^{(c)}, \sigma^{(c)} \in R^{|\mathcal{B}_c| \times K_c}$ denote the predicted mean and standard deviation, respectively, and $\epsilon > 0$ is a small constant added for numerical stability. Here, f_μ and f_σ denote two multi-layer perceptrons used to predict the mean and standard deviation.

Based on these parameters, we define a Gaussian variational distribution over a latent prototype-level variable:

$$q^{(c)}(l^{(c)} \mid S^{(c)}) = \mathcal{N}\left(l^{(c)}; \mu^{(c)}, \mathrm{diag}\left((\sigma^{(c)})^2\right)\right), \tag{20}$$

where $l^{(c)} \in R^{|\mathcal{B}_c| \times K_c}$ denotes a latent prototype-level variable whose variational distribution is parameterized by $\mu^{(c)}$ and $\sigma^{(c)}$.

For each sample $i \in \mathcal{B}_c$, the corresponding variational distribution is

$$q^{(c)}(l_i^{(c)} \mid s_i^{(c)}) = \mathcal{N}\left(l_i^{(c)}; \mu_i^{(c)}, \mathrm{diag}\left((\sigma_i^{(c)})^2\right)\right), \tag{21}$$

where $l_i^{(c)} \in R^{K_c}$ denotes a latent prototype-level variable. The prior mean is initialized as

$$m^{(c)} = \log \frac{1}{K_c} \mathbf{1}_{K_c}, \tag{22}$$

where $\mathbf{1}_{K_c}$ denotes an all-one vector of length K_c. The prior mean is updated during training using an EMA of the posterior means from samples of the corresponding class.

For samples whose ground-truth label is c, the class-level KL term is computed by averaging the

element-wise KL contributions over samples in $\mathcal{B}_c$ and prototype dimensions:

$$\mathcal{L}_{\text{KL}}^{(c)} = \frac{1}{|\mathcal{B}_c| K_c} \sum_{i \in \mathcal{B}_c} \sum_{j=1}^{K_c} d_{i,j}^{(c)}. \tag{23}$$

Here, $d_{i,j}^{(c)}$ denotes the KL contribution for sample i and the j-th prototype dimension. Let $\mu_{i,j}^{(c)}$ and $\sigma_{i,j}^{(c)}$ denote the corresponding entries of $\mu^{(c)}$ and $\sigma^{(c)}$, and let $m_j^{(c)}$ denote the j-th entry of $m^{(c)}$. For diagonal Gaussian distributions,

$$d_{i,j}^{(c)} = \frac{1}{2} \left[\log \frac{\sigma_0^2}{(\sigma_{i,j}^{(c)})^2} + \frac{(\sigma_{i,j}^{(c)})^2 + (\mu_{i,j}^{(c)} - m_j^{(c)})^2}{\sigma_0^2} - 1 \right], \tag{24}$$

The final KL regularization loss is computed as the class-proportion-weighted average:

$$\mathcal{L}_{\text{KL}} = \sum_{c \in \{0,1\}} \frac{|\mathcal{B}_c|}{|\mathcal{B}|} \mathcal{L}_{\text{KL}}^{(c)}, \tag{25}$$

Finally, the KL regularization term is added to the main classification objective:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \beta \mathcal{L}_{\text{KL}} + \lambda \mathcal{L}_{\text{aux}}, \tag{26}$$

where $\mathcal{L}_{\text{cls}}$ denotes the class-weighted cross-entropy loss, where the class weights are computed based on the number of samples in each class, $\mathcal{L}_{\text{aux}}$ denotes the auxiliary loss, and β and λ control the strengths of the KL and auxiliary loss terms, respectively.

4 Experiments and Setup

4.1 Datasets

Clinical Outcome Prediction Tasks. We evaluate SCoPe on two clinical outcome prediction tasks derived from the MIMIC electronic health record databases. Both tasks use de-identified discharge summaries as input and target binary clinical outcomes. Clinical outcome prediction is inherently characterized by class imbalance, as adverse events such as mortality or specific diagnoses represent a small fraction of the patient population. This property is reflected in both datasets: Depression has a positive rate of approximately 27%, while Mortality exhibits extreme imbalance with a positive rate of only 3.6%.

MIMIC-IV Mortality Prediction. (Yoon et al., 2025) A binary classification task constructed from the MIMIC-IV electronic health record dataset (Johnson et al., 2023), aiming to predict outpatient mortality using de-identified discharge notes. We follow the dataset splits provided by (Yoon et al., 2025). The dataset consists of 49,832 de-identified discharge notes labeled as 30-day out-of-hospital mortality or 30-day out-of-hospital alive.

MIMIC-III Depression Prediction. (Gehrmann et al., 2018) A binary classification task constructed from the MIMIC-III electronic health record dataset (Johnson et al., 2016), aiming to predict depression using de-identified discharge notes with an average document length of approximately 2,868 tokens. This dataset consists of 1,271 clinical notes labeled as depression or non-depression.

Hyperpartisan (Kiesel et al., 2019) A binary classification dataset consisting of 1,273 news articles labeled as hyperpartisan or non-hyperpartisan. The average document length is 586 words.

4.2 Baselines

TrLDC (Dai et al., 2022) splits documents into overlapping chunks encoded by sparse-attention transformers, and aggregates chunk representations through a Transformer block with label-wise attention for document-level classification.

IRIS (Li et al., 2025) segments documents into fixed-length chunks, stores their embeddings in a vector database, and retrieves the most task-relevant chunks using learnable query vectors. A linear attention mechanism aggregates the retrieved embeddings for classification.

ModernBERT (Warner et al., 2025) is a general-domain bidirectional encoder supporting sequences of up to 8,192 tokens. We use it as the base encoder for Hyperpartisan.

Clinical-ModernBERT (Lee et al., 2025) adapts ModernBERT architecture to the biomedical and clinical domain through continued pretraining on clinical corpora. We use it as the base encoder for two clinical datasets.

4.3 Results

Table 1 summarizes the main results across all three benchmarks. Overall, SCoPe achieves the best performance on the majority of metrics across all three benchmarks. On Depression, SCoPe outperforms Clinical-ModernBERT in F1 (.854 vs .844, +.010) and Accuracy (.920 vs .914, +.006). On Mortality, where extreme class imbalance poses a substantially harder prediction problem, SCoPe achieves

Table 1: Main results on three long-document classification benchmarks. Best results are in **bold**. All methods use domain-adaptive pretrained encoders: ModernBERT-large for Hyperpartisan and Clinical-ModernBERT for Depression and Mortality. Since these encoders are domain-specific, inapplicable settings are marked with "-".

	Hyperpartisan				Depression				Mortality			
Method	Acc.	AUC	PRC	F1	Acc.	AUC	PRC	F1	Acc.	AUC	PRC	F1
TrLDC	.860	.946	.938	.848	.898	**.929**	.813	.828	.961	.767	.181	.240
ModernBERT-large	.875	.950	.942	.860	–	–	–	–	–	–	–	–
Clinical-ModernBERT	–	–	–	–	.914	.891	.869	.844	.957	.837	.235	.268
IRIS	.865	.937	.926	.847	.557	.590	.336	.335	.726	.771	.089	.137
SCoPe (ours)												
w/o KL	.914	.947	**.956**	.900	.914	.901	.869	.842	.962	**.849**	.240	.299
w/ KL	**.922**	**.954**	.943	**.891**	**.920**	.894	**.876**	**.854**	**.962**	.841	**.249**	**.308**

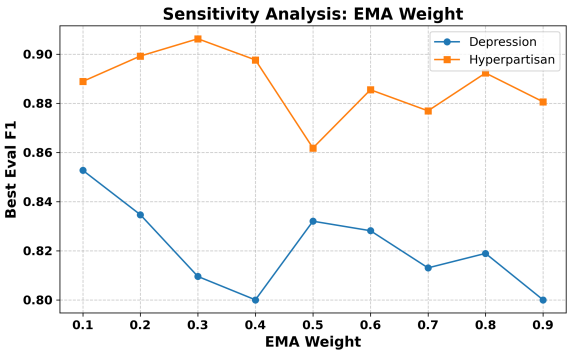

Figure 2: Sensitivity of SCoPe to the EMA smoothing coefficient.

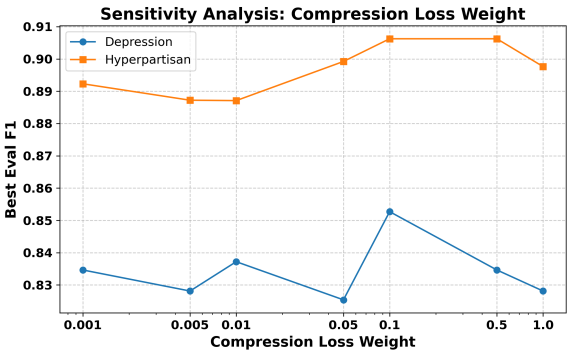

Figure 3: Sensitivity of SCoPe to the compression loss weight.

the highest scores across all four metrics, improving over Clinical-ModernBERT in F1 (.308 vs .268, +.040) and PRAUC (.249 vs .235, +.014). The ablation results in Table 5 in Appendix B reveal a consistent pattern: semantic context-adaptive aggregation alone raises F1 from .268 to .299 (+.031), and distributional regularization provides further improvement to .308 (+.009).

Comparing across the two clinical datasets, the gains from SCoPe over the baselines are notably larger on Mortality than on Depression, suggesting that the proposed framework is particularly effective under a more extreme class imbalance scenario. On Hyperpartisan, a general-domain benchmark with relatively balanced class distribution, SCoPe achieves the highest scores across all four metrics, outperforming ModernBERT-large in F1 (.891 vs .860, +.031) and AUROC (.954 vs .950, +.004). The more detailed and comprehensive ablation results in Table 5 in Appendix B confirm that both semantic unit aggregation (+.039 in F1) and distributional regularization (+.007 in F1) contribute positively. These results confirm that the benefits of SCoPe extend to general long-document classification beyond clinical settings.

4.4 Effect of Compression and Smoothing on Semantic Unit Formation

Semantic Context-Adaptive Unit Aggregation module is mainly governed by two factors: the compression loss weight and the EMA smoothing coefficient. The former controls the influence of the auxiliary compression loss $\mathcal{L}_{\text{aux}}$, determining how strongly the model is encouraged to merge tokens into semantic units. The latter controls information propagation between adjacent semantic units, with larger values producing stronger smoothing. We analyze the impact of both factors on downstream classification performance.

Figure 2 shows the effect of the EMA smoothing coefficient. Depression performs best at 0.1, while Hyperpartisan peaks at 0.3. Performance generally decreases with larger coefficients, suggesting that light smoothing helps stabilize boundary signals, whereas excessive smoothing can blur task-relevant local evidence.

As shown in Figure 3, both datasets obtain the best F1 when the compression loss weight is set to 0.1. Smaller weights provide insufficient guidance for forming coherent phrase units, whereas larger weights may overemphasize compression and degrade boundary quality. This suggests that

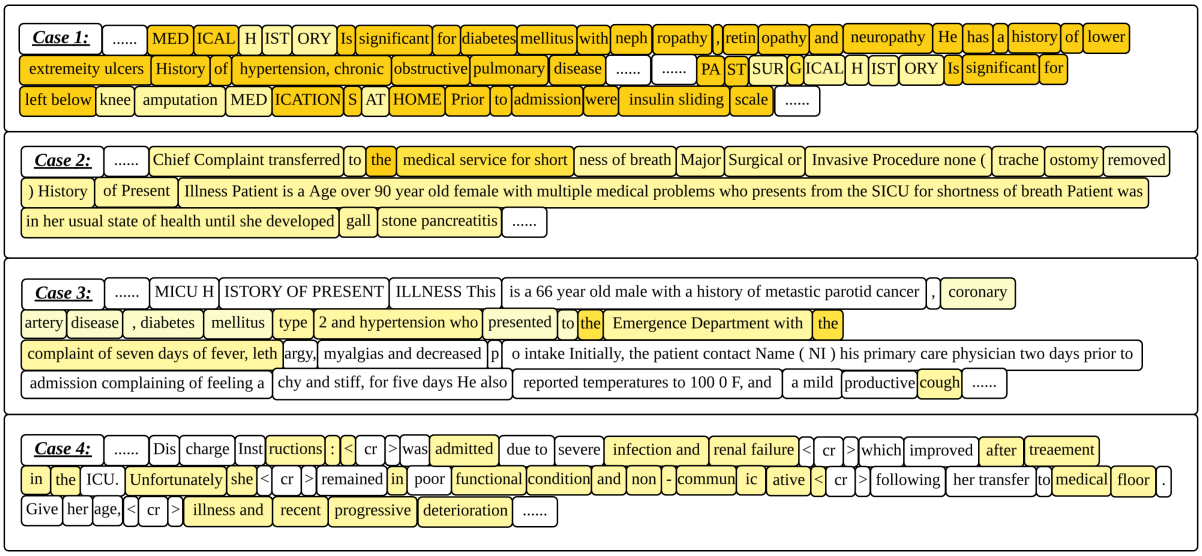

Figure 4: Visualization of learned phrase units for clinical prediction. Cases 1–3 correspond to depression prediction, and Case 4 corresponds to mortality prediction. Bordered spans indicate model-aggregated phrase units; darker yellow denotes higher attention weight.

a moderate compression weight best balances semantic unit formation with the main classification objective.

4.5 Interpretability Analysis

In Case 1, the model assigns high attention weight to phrase-level units related to chronic disease burden and functional impairment, including “extremity ulcers,” “hypertension, chronic,” ”left below”, ”knee”, ”amputation”, and ”insulin sliding”, ”scale”. Although these phrases are not explicit mentions of depression, they describe clinically relevant comorbidities, diabetic complications, mobility limitations, and ongoing treatment needs. This suggests that the model can capture indirect contextual signals associated with depression risk rather than relying only on explicit diagnostic mentions.

In Case 2, compared with the phrase-level units in Case 1, the model forms coarser semantic units that aggregate longer contextual descriptions, including the patient’s age, prior health status, multiple medical problems, SICU transfer, and acute illness trajectory into a single semantic unit. This suggests that the module can adaptively adjust the granularity of semantic aggregation according to context.

In Case 3, the model attends to both chronic comorbidities, including metastatic cancer, coronary artery disease, type 2 diabetes, and hypertension, and acute illness descriptions, such as fever, myalgias, reduced oral intake, and productive cough. Compared with Case 1, this demonstrates that the module can aggregate semantic units with different clinical functions depending on the local context.

In Case 4, for mortality prediction, the model focuses on semantic units such as "infection,” “renal failure, "functional" "condition” ”non-communicative,” and ”recent" "progressive" "deterioration,” all of which are clinically plausible indicators of poor prognosis. In addition, the attention to ”unfortunately” suggests that the model may capture clinician sentiment or pragmatic cues associated with anticipated deterioration.

5 Conclusion

In this work, we introduced SCoPe, a framework for clinical long-document classification that adaptively constructs intermediate-granularity semantic units from token representations without any predefined segmentation. SCoPe identifies semantic boundaries based on the degree of semantic continuity between adjacent tokens and employs a distributional regularization to stabilize prototype learning for class-conditional semantic unit attribution. Experiments on clinical outcome prediction tasks and Hyperpartisan News Detection demonstrate that SCoPe consistently outperforms strong baselines, with particularly notable gains under extreme class imbalance, while providing inherent interpretability at a new intermediate granularity level.

Limitations

Although SCoPe obtains promising results on benchmark EHR datasets such as MIMIC-III (Gehrmann et al., 2018) and MIMIC-IV (Yoon et al., 2025), it is currently evaluated in EHR discharge summaries that are exclusively written in the English language. Its applicability and effectiveness for non-English EHR discharge summaries require further investigation. Further research is needed to develop more robust and reliable language models for clinical applications across diverse languages and populations. In addition, the quality of the learned semantic prototypes heavily depends on the training data. If the data contains inherent biases, these biases may be reflected in the prototypes, potentially leading to biased predictions or interpretations.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the 2nd clinical natural language processing workshop*, pages 72–78.

David Arthur, Sergei Vassilvitskii, and 1 others. 2007. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Ilias Chalkidis, Xiang Dai, Manos Fergadiotis, Prodromos Malakasiotis, and Desmond Elliott. 2022. An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint arXiv:2210.05529*.

Qingyu Chen, Jingcheng Du, Sun Kim, W John Wilbur, and Zhiyong Lu. 2020. Deep learning with sentence embeddings pre-trained on biomedical corpora improves the performance of finding similar sentences in electronic medical records. *BMC Medical Informatics and Decision Making*, 20(Suppl 1):73.

Xiang Dai, Ilias Chalkidis, Sune Darkner, and Desmond Elliott. 2022. Revisiting transformer-based models for long document classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7212–7230.

Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Cogltx: Applying bert to long texts. *Advances in Neural Information Processing Systems*, 33:12792–12804.

Sebastian Gehrmann, Franck Dernoncourt, Yeran Li, Eric T Carlson, Joy T Wu, Jonathan Welt, John Foote Jr, Edward T Moseley, David W Grant, Patrick D Tyler, and 1 others. 2018. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PloS one*, 13(2):e0192360.

Lei Gong, Aidong Zhang, and Kishlay Jha. 2025. Boosting clinical outcome prediction with context-aware feature imputation and disentanglement. In *2025 IEEE International Conference on Data Mining (ICDM)*, pages 1234–1243.

Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Dat Hong, Tong Wang, and Stephen Baek. 2023. Protorynet-interpretable text classification via prototype trajectories. *Journal of Machine Learning Research*, 24(264):1–39.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Zihao Huang, Jundong Zhou, Xingwei Qu, Qiyang Min, and Ge Zhang. 2026. Conceptmoe: Adaptive token-to-concept compression for implicit compute allocation. *arXiv preprint arXiv:2601.21420*.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, and 1 others. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 829–839.

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

Simon A Lee, Anthony Wu, and Jeffrey N Chiang. 2025. Clinical modernbert: An efficient and long context encoder for biomedical text. *arXiv preprint arXiv:2504.03964*.

Fengnan Li, Elliot D Hill, Jiang Shu, Jiaxin Gao, and Matthew M Engelhard. 2025. Iris: Interpretable retrieval-augmented classification for long interspersed document sequences. In *Proceedings of the 63rd Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 30263–30283.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)*, pages 1101–1111.

Aakanksha Naik, Sravanthi Parasa, Sergey Feldman, Lucy Lu Wang, and Tom Hope. 2022. Literature-augmented clinical outcome prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 438–453.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 197–207.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, pages 838–844. ieee.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Thomas Sounack, Joshua Davis, Brigitte Durieux, Antoine Chaffin, Tom J Pollard, Eric Lehman, Alistair EW Johnson, Matthew McDermott, Tristan Naumann, and Charlotta Lindvall. 2025. Bioclinical modernbert: A state-of-the-art long-context encoder for biomedical and clinical nlp. *arXiv preprint arXiv:2506.10896*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30.

Betty Van Aken, Jens-Michalis Papaioannou, Marcel Naik, Georgios Eleftheriadis, Wolfgang Nejdl, Felix Gers, and Alexander Löser. 2022. This patient looks like that patient: Prototypical networks for interpretable diagnosis prediction from clinical text. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 172–184.

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

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, and 1 others. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547.

Bowen Wei and Ziwei Zhu. 2025. Protolens: Advancing prototype learning for fine-grained interpretability in text classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4503–4523.

Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2018. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3474–3482.

Zhichao Yang, Avijit Mitra, Sunjae Kwon, and Hong Yu. 2024. Clinicalmamba: A generative clinical language model on longitudinal clinical notes. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 54–63.

Zhichao Yang, Shufan Wang, Bhanu Pratap Singh Rawat, Avijit Mitra, and Hong Yu. 2022. Knowledge injected prompt based fine-tuning for multi-label few-shot icd coding. In *Findings of the association for computational linguistics: EMNLP 2022*, pages 1767–1781.

WonJin Yoon, Shan Chen, Yanjun Gao, Zhanzhan Zhao, Dmitriy Dligach, Danielle S Bitterman, Majid Afshar, and Timothy Miller. 2025. Lcd benchmark: long clinical document benchmark on mortality prediction for language models. *Journal of the American Medical Informatics Association*, 32(2):285–295.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Yuan Zheng, Rihui Cai, Maihemuti Maimaiti, and Kahaerjiang Abiderexiti. 2023. Chunk-bert: Boosted keyword extraction for long scientific literature via bert with chunking capabilities. In *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 385–392. IEEE.

A Appendix

A.1 Hyperparameter Sweeps

Hyperparameter sweep for ModernBERT-large on Hyperpartisan. We performed a grid search over the following hyperparameters for `answerdotai/ModernBERT-large` on the Hyperpartisan dataset. The best model checkpoint was selected according to `positive class F1`. The explored hyperparameter ranges were: learning rate $\in \{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$, gradient accumulation steps $\in \{1, 8, 24\}$, LoRA rank $r \in \{8, 16\}$, LoRA scaling factor $\alpha \in \{16, 32\}$, LoRA injection layers $\in \{0, 1, 4, 8, 12\}$, learning rate scheduler $\in \{\texttt{linear}, \texttt{cosine}\}$, and number of training epochs $\in \{20, 30\}$. All other settings were fixed: batch size $= 2$, evaluation batch size $= 2$, maximum input length $= 8192$, weight decay $= 0.01$, warmup ratio $= 0.05$, max gradient norm $= 0.5$, early stopping patience $= 10$, early stopping threshold $= 0.03$, and random seed $= 42$.

Hyperparameter sweep for the SCoPe Hyperpartisan. We further conducted a grid search for our concept-based ModernBERT-large variant on the Hyperpartisan dataset, under the setting without the balanced-loss regularizer and using only label-wise attention and concept merging. The sweep objective was `positive class F1 of evaluation dataset`. We varied the following hyperparameters: attention temperature $\in \{1, 2, 3\}$, compression ratio $\in \{2, 4\}$, gradient accumulation steps $\in \{1, 8\}$, learning rate $\in \{5\times10^{-5}, 1\times10^{-4}\}$, LoRA injection layers $\in \{0, 12\}$, learning rate scheduler $\in \{\texttt{linear}, \texttt{cosine}\}$, number of training epochs $\in \{20, 30\}$, and number of prototypes $\in \{5, 10\}$. All other hyperparameters were fixed: batch size $= 2$, evaluation batch size $= 2$, maximum input length $= 8192$, embedding dimension $= 1024$, projection dimension $= 1024$, encoder learning rate $= 5 \times 10^{-5}$, prototype learning rate $= 0.05$, LoRA rank $= 16$, LoRA alpha $= 16$, auxiliary loss weight $\lambda_{\text{aux}} = 0.1$, EMA alpha $= 0.3$, sharp rate $= 6$, similarity metric=`euclidean`, weight decay $= 0.01$, warmup ratio $= 0.05$, max gradient norm $= 0.5$, early stopping patience $= 10$, early stopping threshold $= 0.03$, VIB coefficient $\beta = 0$, VIB warmup ratio $= 0$, and random seed $= 42$.

Hyperparameter sweep with balanced aligned loss on Hyperpartisan. After selecting the concept-merging and label-wise attention configuration, we conducted an additional grid search on the Hyperpartisan dataset with the balanced aligned loss enabled. We varied the following hyperparameters: attention temperature $\in \{1, 2, 3\}$, compression ratio $\in \{2, 4\}$, gradient accumulation steps $\in \{1, 8\}$, learning rate $\in \{5\times10^{-5}, 1\times10^{-4}\}$, LoRA injection layers $\in \{0, 12\}$, learning rate scheduler $\in \{\texttt{linear}, \texttt{cosine}\}$, number of training epochs $\in \{20, 30\}$, and number of prototypes $\in \{5, 10\}$. All other hyperparameters were fixed: batch size $= 2$, evaluation batch size $= 2$, maximum input length $= 8192$, embedding dimension $= 1024$, projection dimension $= 1024$, encoder learning rate $= 5 \times 10^{-5}$, prototype learning rate $= 0.05$, LoRA rank $= 16$, LoRA alpha $= 16$, auxiliary loss weight $\lambda_{\text{aux}} = 0.1$, EMA alpha $= 0.3$, sharp rate $= 6$, similarity metric=`euclidean`, weight decay $= 0.01$, warmup ratio $= 0.05$, max gradient norm $= 0.5$, early stopping patience $= 10$, early stopping threshold $= 0.03$, VIB coefficient $\beta = 0$, VIB warmup ratio $= 0$, and random seed $= 42$. This sweep covers 384 hyperparameter configurations in total.

Hyperparameter sweep for Clinical-ModernBERT on Depression. We performed a grid search over the following hyperparameters for `Simonlee711/Clinical_ModernBERT` on the Depression dataset. The best model checkpoint was selected according to `positive class F1`. The explored hyperparameter ranges were: learning rate $\in \{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$, gradient accumulation steps $\in \{1, 8, 24\}$, LoRA injection layers $\in \{0, 1, 4\}$, and learning rate scheduler $\in \{\texttt{linear}, \texttt{cosine}\}$. All other settings were fixed: batch size $= 2$, evaluation batch size $= 2$, maximum input length $= 8192$, LoRA rank $r = 16$, LoRA scaling factor $\alpha = 16$, weight decay $= 0.01$, warmup ratio $= 0.05$, max gradient norm $= 0.5$, number of training epochs $= 20$, early stopping patience $= 10$, early stopping threshold $= 0.03$, and random seed $= 42$.

Hyperparameter sweep for SCoPe on Depression. We performed a grid search over the following hyperparameters for SCoPe with `Simonlee711/Clinical_ModernBERT` on the Depression dataset. The search objective was `Positive Class F1`, while the best model checkpoint was selected according to `positive class F1 on evaluation dataset`. The explored hyperparameter ranges were: learning rate $\in \{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$, gradient accumulation steps $\in \{1, 8, 24\}$, LoRA in-

jection layers $\in \{0, 12\}$, learning rate scheduler $\in \{\texttt{linear}, \texttt{cosine}\}$, attention temperature $\in \{1, 2, 3\}$, number of prototypes $\in \{5, 10\}$, and projection dimension $\in \{768, 1024\}$. All other settings were fixed: batch size $= 2$, evaluation batch size $= 2$, maximum input length $= 8192$, compression ratio $= 2$, EMA $\alpha = 0.3$, auxiliary loss weight $\lambda_{\text{aux}} = 0.1$, sharpening rate $= 6$, similarity method $=$ euclidean, LoRA rank $r = 16$, LoRA scaling factor $\alpha = 16$, weight decay $= 0.01$, warmup ratio $= 0.05$, max gradient norm $= 0.5$, number of training epochs $= 20$, early stopping patience $= 10$, early stopping threshold $= 0.03$, and random seed $= 42$.

Hyperparameter sweep for SCoPe (with KL) on Depression. We performed a grid search over the following hyperparameters for SCoPe with distributional regularization using `Simonlee711/Clinical_ModernBERT` on the Depression dataset. The best model checkpoint was selected according to `positive class F1 on evaluation dataset`. The explored hyperparameter ranges were: attention temperature $\in \{1, 2, 3\}$, EMA $\alpha \in \{0.1, 0.3, 0.5\}$, LoRA injection layers $\in \{0, 1, 4\}$, learning rate scheduler $\in \{\texttt{linear}, \texttt{cosine}\}$, and number of prototypes $\in \{1, 3, 5\}$. All other settings were fixed: learning rate $= 1 \times 10^{-4}$, batch size $= 2$, evaluation batch size $= 2$, gradient accumulation steps $= 1$, maximum input length $= 8192$, compression ratio $= 2$, auxiliary loss weight $\lambda_{\text{aux}} = 0.1$, KL weight $\beta = 0.01$, KL warmup ratio $= 0.1$, sharpening rate $= 6$, similarity method $=$ euclidean, projection dimension $= 768$, LoRA rank $r = 16$, LoRA scaling factor $\alpha = 16$, weight decay $= 0.01$, warmup ratio $= 0.05$, max gradient norm $= 0.5$, number of training epochs $= 20$, early stopping patience $= 10$, early stopping threshold $= 0.03$, and random seed $= 42$.

A.2 Dataset Statistics

Table 2: Statistics of the Hyperpartisan dataset.

	Train	Dev	Test
Total samples	1,016	127	128
Positive	436	61	55
Negative	580	66	73
Avg length (all)	586.9	572.8	591.6
Avg length (pos.)	748.2	693.0	773.5
Avg length (neg.)	465.7	461.7	454.5

Table 3: Statistics of the Mortality dataset.

	Train	Dev	Test
Total samples	34,759	7,505	7,568
Positive	1,267	302	261
Negative	33,492	7,203	7,307
Avg length (all)	4,137.9	4,147.9	4,143.0
Avg length (pos.)	4,929.2	4,981.2	4,976.0
Avg length (neg.)	4,107.9	4,112.9	4,113.3

Table 4: Statistics of the Depression dataset.

	Train	Dev	Test
Total samples	1,107	188	314
Positive	302	70	88
Negative	805	118	226
Avg length (all)	2,867.8	2,950.4	3,034.8
Avg length (pos.)	2,970.2	3,147.8	3,340.2
Avg length (neg.)	2,829.3	2,833.3	2,916.0

B Ablation Study

The complete ablation results across all three datasets are provided in Table 5, showing the individual contribution of each module to the SCoPe framework in higher precision with precision and recall.

C Case Study

Figure 5 visualizes the learned concept boundaries on a sample from the Depression prediction dataset. The model successfully aggregates clinically meaningful multi-token phrases into coherent semantic units, such as "of Parkinson's disease", "hypertension, and recent aspiration", and "mental status". However, numerical tokens such as dates and identifiers (e.g., "2 132 6 26") remain fragmented into individual token-level units, indicating that the current boundary prediction mechanism is less effective at recognizing numeric sequences as coherent units. This suggests that numeric information, which lacks the contextual semantic continuity present in natural language phrases, represents a remaining challenge for the semantic context-adaptive aggregation approach.

C.1 Hyperparmeter Sensivity

As shown in Figure 6, the optimal number of per-class prototypes differs across datasets: Depression achieves its best F1 with 3 prototypes per class, while Hyperpartisan peaks at 5–10. Since each class maintains its own set of prototypes, this differ-

Table 5: Ablation studies on three datasets.

Method	Acc.	AUROC	PRAUC	F1	Precision	Recall
Hyperpartisan						
ModernBERT-large	0.8750	0.9499	0.9418	0.8597	0.8305	0.8909
SCoPe w/ EMA, w/o KL	0.91406	0.94695	**0.95856**	0.90009	0.89286	0.90909
SCoPe w/ EMA, w/ KL	**0.92188**	**0.95355**	0.94287	**0.91071**	**0.89474**	**0.92727**
Mortality Prediction						
ClinicalModernBERT	0.95679	0.83762	0.23536	0.26846	0.32258	0.22989
SCoPe w/o KL	0.96221	**0.84899**	0.23973	0.29902	**0.41497**	0.23372
SCoPe w/ KL	**0.96195**	0.84085	**0.24891**	**0.30769**	0.41290	**0.24521**
Depression Prediction						
ClinicalModernBERT	0.91401	0.89104	0.86874	0.84393	0.85882	0.82955
SCoPe w/o KL	0.91401	**0.90145**	0.86909	0.84211	0.86747	0.81818
SCoPe w/ KL	**0.92038**	0.89368	**0.87639**	**0.8538**	**0.87952**	**0.82955**

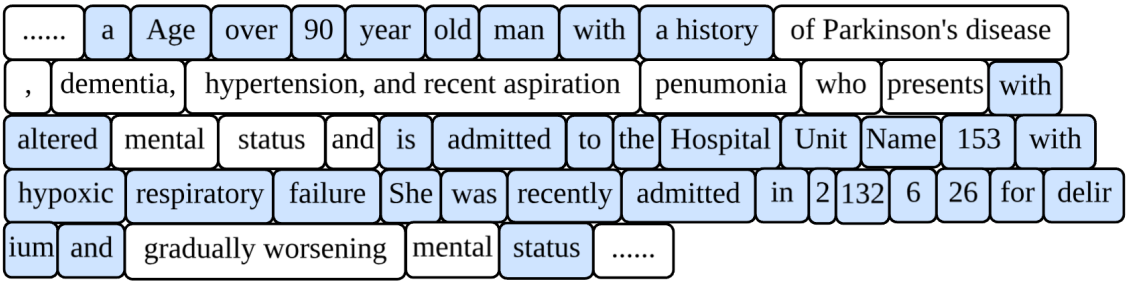

Figure 5: Visualization of learned semantic boundaries on a Mortality prediction sample. Each colored block represents a concept identified by SCoPe. The model groups semantically related tokens into coherent semantic units such as "Age over 90 year old man" and "hypertension, and recent aspiration", while functional tokens like "with" and "and" tend to form single-token words or merge into adjacent units.

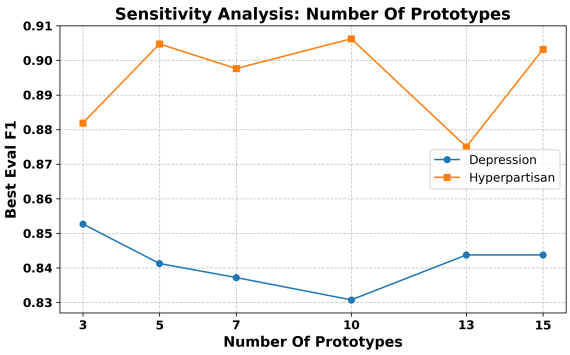

Figure 6: Sensitivity of SCoPe to the Number of Prototypes.

ence likely reflects the available training signal per class. In the more imbalanced Depression dataset, the minority class contains fewer samples, and a small number of prototypes is sufficient to capture its distributional patterns; additional prototypes lack adequate training signal and may introduce noise. In the more balanced Hyperpartisan dataset, each class has sufficient samples to support a larger prototype set that captures more diverse semantic sub-patterns. Despite these differences, SCoPe maintains relatively stable performance across the tested range on both datasets, with F1 variations within approximately two percentage points.

C.2 Prototype Initialization

We initialize the class-specific prototypes using K-means++ clustering on the training-set representations. Given the training embeddings and their binary class labels, we first separate the embeddings according to the class label. If the input embeddings are token-level representations with shape $[N, L, d]$, we use the first-token representation, corresponding to the `[CLS]` embedding, as the instance-level representation. If the input embeddings are already instance-level representations with shape $[N, d]$, they are used directly.

For each class $c \in \{0, 1\}$, let $\mathcal{H}_c = \{h_i : y_i =$

$c\}$ denote the set of training embeddings belonging to class c. We independently apply K-means++ clustering(Arthur et al., 2007) to $\mathcal{H}_c$ with K_c clusters, where K_c is the number of prototypes assigned to class c. The resulting cluster centers are used as the initial prototype vectors:

$$\{p_j^{(c)}\}_{j=1}^{K_c} = \mathrm{KMeans}++(\mathcal{H}_c, K_c). \tag{27}$$

In our implementation, K-means is run with multiple random initializations and a fixed random seed. The maximum number of iterations is set to 300.

When the number of training instances in a class is smaller than the requested number of prototypes, we first set the number of clusters to the available number of class-specific instances and then duplicate selected cluster centers to obtain K_c prototype vectors. After initialization, the prototypes are converted to learnable model parameters and are updated during training.

C.3 Implementation Details

All experiments are conducted on a single NVIDIA A6000 GPU with 48GB memory. We use the AdamW optimizer for all models. Training time varies across datasets: each run takes approximately 48 hours on Mortality, 1 hour on Depression, and 40 minutes on Hyperpartisan. For baseline reproduction, TrLDC and IRIS are trained using the hyperparameters reported in their original papers, with the maximum input length adjusted to match our experimental setting. IRIS uses the same base encoder as SCoPe for a fair comparison. For Clinical-ModernBERT and ModernBERT-large, we conduct a hyperparameter search over learning rate, learning rate scheduler, and gradient accumulation steps that covers the same search space used for SCoPe.