Learning Under Extreme Label Imbalance in EHRs: A Dependency-Aware Loss for Multi-Label Classification

Iris Szu-Szu Ho* IRIS.S.S.HO@ED.AC.UK
Lars Werne L.P.J.WERNE@SMS.ED.AC.UK
School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Konrad Rawlik KONRAD.RAWLIK@ED.AC.UK
Centre for Inflammation Research, University of Edinburgh, Edinburgh, United Kingdom

Bruce Guthrie BRUCE.GUTHRIE@ED.AC.UK
Medical School, University of Edinburgh, Edinburgh, United Kingdom

Sohan Seth SOHAN.SETH@ED.AC.UK
School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

Abstract

Extreme multi-label next-visit diagnosis forecasting from electronic health records is dominated by label sparsity. Each visit contains only a handful of positive ICD-10 codes among thousands of candidates, yet codes are strongly correlated through comorbidity structure. In this regime, standard element-wise objectives (such as focal, and class-balanced loss) often maximize sensitivity at the cost of severe precision degradation, producing clinically impractical alert volumes. We propose an architecture-compatible dependency-aware ranking loss that (i) reweights per-code correctness under severe imbalance, (ii) aggregates errors with rank-based emphasis on the hardest labels, and (iii) regularizes predictions with a learned pairwise dependency term in the output space. Using an EHR Transformer backbone, we evaluate on the CPRD cohort (V=1,538 codes), benchmarking loss functions on 200,000 patients and validating scalability up to 3.2 million. The proposed objective shifts the precision–recall trade-off toward fewer false positives while maintaining competitive sensitivity, and preserves overall ranking quality (PRC–AUC comparable to weighted BCE). In addition, it yields an auditable population-level dependency matrix summarizing learned co-occurrence structure. These results suggest that explicit output-space structure can improve the precision–recall trade-off in sparse, high-dimensional next-visit diagnosis prediction from EHRs.

* Corresponding author: iris.s.s.ho@ed.ac.uk

Data and Code Availability This paper uses data from the Clinical Practice Research Datalink (CPRD), which provides anonymized primary care records from the UK. Access to CPRD data is subject to protocol approval by the Independent Scientific Advisory Committee (ISAC) and licensing agreements. Therefore, the raw data cannot be made publicly available. Implementations of the proposed dependency-aware loss and baseline loss functions are publicly available.[1]

Institutional Review Board (IRB) This study does not require Institutional Review Board (IRB) approval. The study was conducted using the CPRD, which consists entirely of de-identified, retrospective electronic health records. As the data is anonymized and involves no direct interaction with human subjects, this work is exempt from IRB review.

1. Introduction

Electronic Health Records (EHRs) capture longitudinal patient trajectories and enable predictive models to forecast future diagnoses and stratify risk (Wang et al., 2024). Transformer-based architectures, including BEHRT (Li et al., 2020) and Med-BERT (Rasmy et al., 2021), have become common backbones for learning representations from sequences of visits and clinical concepts. In multi-label diagnosis prediction, however, performance is often limited less by the encoder than by the training objective. Most

1. https://github.com/Irisisi/dependency-aware-loss

systems optimise a per-label sigmoid binary cross-entropy (BCE), implicitly treating each code as an independent Bernoulli outcome.

This standard paradigm faces two central difficulties in next-visit ICD-10 code prediction. First, the label space is extremely sparse. At a typical visit, only a small number of diagnoses are recorded (e.g., 2–8 positives) out of a vocabulary of thousands, so positives are orders of magnitude rarer than negatives. Under this regime, weighted cross-entropy (BCE) often prioritises the overwhelming number of easy negatives, yielding high specificity but low sensitivity to rare diagnoses. Conversely, losses designed to emphasise hard or rare examples, such as focal loss (Lin et al., 2017) or class-balanced reweighting (Cui et al., 2019), can over-correct under extreme sparsity, achieving high recall by producing a large number of false alerts and thereby driving precision to levels that are difficult to operationalise in clinical review workflows.

Second, element-wise multi-label objectives decompose supervision as a sum over labels, which is equivalent to modelling each code as an independent Bernoulli given the patient representation. They factorise the joint distribution as $\prod_{v=1}^{V} P(y_v \mid x)$. This ignores clinically grounded dependencies between diagnoses (e.g., type 2 diabetes co-occurring with chronic kidney disease) and provides no explicit penalty for predicting incoherent or implausible diagnosis sets. While a Transformer encoder may learn some correlations implicitly in its latent space, the objective itself does not constrain the output to reflect comorbidity structure.

To address these challenges, we propose a dependency-aware loss tailored to high-dimensional, sparse multi-label prediction. The objective (i) prioritises hard codes through rank-weighted aggregation, mitigating gradient dilution under sparsity, and (ii) introduces a learnable interaction regulariser that couples labels to encourage coherent comorbidity structure. This yields an architecture-compatible loss (requiring only per-code logits) and an inspectable dependency matrix for global auditing. Our central question is whether coupling labels and prioritising relative ranking can improve present and new diagnosis detection under extreme sparsity without collapsing precision.

2. Related Work

EHR Transformers and input structure. Transformer architectures are widely used for modelling EHR trajectories, including next-visit prediction and masked modelling of clinical concepts (Li et al., 2020; Rasmy et al., 2021). Many approaches represent a patient record as a sequence of visits and embed clinical concepts with visit-/time-aware context (Li et al., 2020; Rasmy et al., 2021; Rupp et al., 2023; Moore et al., 2024). Within a visit, diagnoses are inherently set-valued (unordered), making invariance to within-visit permutations a natural inductive bias. Set-function models such as Deep Sets and attention-based Set Transformers formalise architectures that respect this symmetry (Zaheer et al., 2017; Lee et al., 2019). More broadly, empirical work on tabular and set-structured data shows that neural pipelines can become sensitive to arbitrary feature/element order when invariance is not enforced (Zhu et al., 2022). We therefore adopt an explicitly order-invariant within-visit representation, ensuring that our loss function comparisons are not confounded by sensitivity to arbitrary input ordering.

Distinction from scalar and binary prediction tasks. A large body of EHR prediction work targets scalar or low-dimensional outcomes, such as mortality risk, readmission, or treatment effect estimation, where the output is a single probability or a small number of classes (Yang et al., 2022; Darabi et al., 2020; Lee et al., 2025). In contrast, next-visit diagnosis prediction is an extreme multi-label task. The output is a high-dimensional binary vector over a large ICD vocabulary, with only a handful of positives per visit. This regime yields a qualitatively different optimisation landscape, as gradients from rare positives are easily diluted by the vast accumulation of negatives.

Loss design under extreme sparsity and label dependence. Most diagnosis predictors are trained with sigmoid cross-entropy, which treats each label as an independent Bernoulli given the patient representation (Li et al., 2020). Under extreme sparsity, standard BCE is often dominated by easy negatives. This has motivated the use of imbalance-aware objectives such as weighted BCE, focal loss (Lin et al., 2017), and class-balanced reweighting (Cui et al., 2019). However, these objectives remain element-wise. They rescale per-label contributions but still optimise each label independently, ignoring clinically grounded correlations.

In very high-dimensional spaces, such reweighting can shift the operating point toward high recall at the cost of markedly reduced precision. This necessitates loss functions that can (i) separate true codes from the vast background of negatives and (ii) encode output-level structure reflecting known comorbidity patterns.

3. Methodology

3.1. Dataset and Preprocessing

Cohort. We use anonymised EHRs from the CPRD, a UK database with linked secondary-care ICD–10 diagnoses. We selected this large-scale cohort specifically to evaluate optimisation behaviour in a high-dimensional, extremely sparse label space. This allows us to rigorously evaluate the loss function's stability and scalability across varying data regimes (up to 3.2 million patients), rather than focusing on external validation across smaller, lower-dimensional cohorts. We use 5,000 patients for hyperparameter tuning and stability checks, and assess scalability on 200,000 and 3.2 million patients using the same protocol.

Code vocabulary and scope. We use a long-term condition codelist, containing 1,538 ICD-10 codes observed in the cohort to focus on multimorbidity patterns (Anonymous Authors, 2022). Acute conditions and non-diagnostic entries are excluded. To strictly isolate the impact of the loss function on extreme sparsity and dependence, we restrict inputs to diagnoses. We use diagnosis-only inputs as a controlled setting to isolate the effect of the objective under extreme multi-label sparsity and comorbidity structure. Additional modalities such as medications and laboratory values are clinically important, but would introduce modality-specific preprocessing, missingness, and temporal-alignment assumptions, making it harder to attribute gains specifically to the loss function.

Visit construction and multi-hot representation. For each patient i, diagnosis events are sorted by date to recover the temporal order of visits. Records from the same day are grouped into a single visit, yielding visit-level diagnosis sets $\{S_{i,\ell}\}_{\ell=1}^{L_i}$ ordered in time by their visit dates, where $S_{i,\ell} \subseteq \{1,\dots,V\}$ and $V = 1{,}538$. Within each visit, diagnoses are treated as an unordered set (no within-visit sequence is assumed). Each visit is represented as a multi-hot vector

$$x_{i,\ell} \in \{0,1\}^V, \qquad [x_{i,\ell}]_v = \mathbb{I}\{v \in S_{i,\ell}\},$$

so the patient trajectory is $X_i = [x_{i,1},\dots,x_{i,L_i}]^\top \in \{0,1\}^{L_i \times V}$.

Prediction task. We study next-visit diagnosis prediction as an extreme multi-label problem. Given history up to visit ℓ, the model predicts the ICD–10 code vector at visit $\ell+1$. In addition to overall code prediction, we report performance on new-code prediction, where positives are codes that have not previously appeared in the patient's history.

3.2. Backbone Predictor

We use a hierarchical EHR encoder to map a patient's past visits to per-code logits for the next visit. The backbone is identical for all losses in Section 3.5. We choose this causal, order-invariant hierarchical Transformer so that differences reflect the loss, not architectural confounds. Our comparisons isolate the effect of the training objective.

Within-visit set encoding. Each ICD–10 code $v \in \{1,\dots,V\}$ has a learned embedding $e_v^{(\text{icd})} \in \mathbb{R}^D$. Given the multi-hot visit vector $x_{i,\ell}$ (Section 3.1), we extract the set of embeddings for active codes and summarise them with permutation-invariant TopK attention (Gupta et al., 2021):

$$v_{i,\ell} = \text{TopK}\Big(\{e_v^{(\text{icd})} : [x_{i,\ell}]_v = 1\}\Big) \in \mathbb{R}^D.$$

This construction respects the set-valued nature of within-visit diagnoses and prevents dependence on any arbitrary within-visit ordering. Full TopK equations are provided in Appendix B.

Temporal and demographic covariates and embeddings. Each visit representation is augmented with visit position, age, sex, and the time since the previous visit. Let $d_{i,\ell}$ denote the date of visit ℓ for patient i. We compute

$$a_{i,\ell} = \text{age}_i(d_{i,\ell}), \qquad \Delta\tau_{i,\ell} = \begin{cases} 0, & \ell = 1, \\ d_{i,\ell} - d_{i,\ell-1}, & \ell > 1, \end{cases}$$

and denote sex by s_i (broadcast to all visits). In the model, we map these covariates and the visit index ℓ to learned embeddings in $\mathbb{R}^D$,

$$e_{i,\ell}^{(\text{pos})},\ e_{i,\ell}^{(\text{age})},\ e_{i,\ell}^{(\text{time})},\ e_{i,\ell}^{(\text{sex})} \in \mathbb{R}^D,$$

and form the input to the across-visit encoder by addition (Appendix C):

$$z_{i,\ell}^{(0)} = v_{i,\ell} + e_{i,\ell}^{(\text{pos})} + e_{i,\ell}^{(\text{age})} + e_{i,\ell}^{(\text{time})} + e_{i,\ell}^{(\text{sex})}.$$

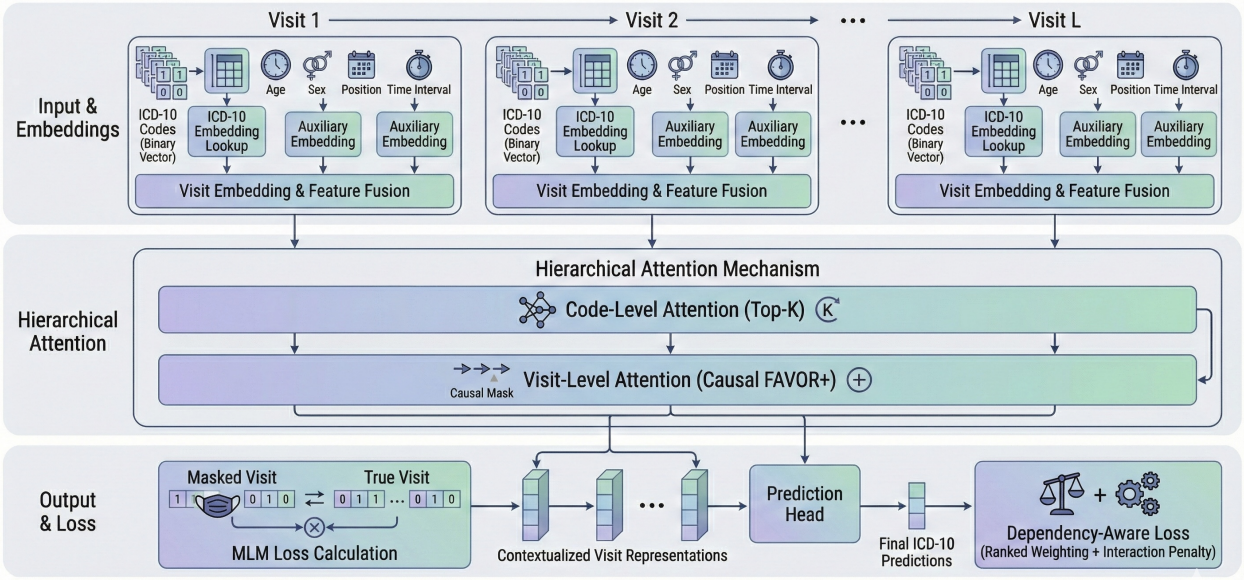

Figure 1: Model Architecture: EHR-Hierarchical Transformer (EHR-HiT).

Across-visit temporal encoder (causal). The sequence $(z_{i,1}^{(0)}, \ldots, z_{i,L_i}^{(0)})$ is passed through a causal transformer based on FAVOR+ (Choromanski et al., 2020), yielding contextualised visit states $(z_{i,1}^{(1)}, \ldots, z_{i,L_i}^{(1)})$ with attention restricted to past visits only. FAVOR+ provides linear-time attention in sequence length. Details are in Appendix D.

Next-visit multi-label prediction. Given the contextualised state at visit ℓ, the model outputs logits $\hat{y}_{i,\ell+1} \in \mathbb{R}^V$ for the codes at the next visit:

$$\hat{y}_{i,\ell+1} = f_\theta\left(z_{i,\ell}^{(1)}\right), \qquad \hat{p}_{i,\ell+1} = \sigma(\hat{y}_{i,\ell+1}).$$

All losses considered in Section 3.5 operate on these per-code logits/probabilities. The architecture above is fixed.

3.3. New-Code-Enriched Masked-Visit Training

To align training with early detection, we use a new-code-enriched masked-visit objective for optimisation and for computing the validation loss used in learning-rate scheduling and early stopping. For each patient trajectory i, eligible target visits are visits containing at least one ICD-10 code that has not appeared earlier in the patient's history. Each eligible visit is selected independently with probability $p_{\text{visit}} = 0.15$. Trajectories with no selected eligible visit contribute no supervised target for this objective in that iteration. If no eligible visit is selected in an entire mini-batch, the supervised update is skipped.

For a selected target visit ℓ, we apply a BERT-style corruption scheme (Devlin et al., 2019) to the active ICD-10 codes at that visit: 80% are removed from the corrupted input representation, 10% are replaced with randomly sampled ICD-10 codes, and 10% remain unchanged. The original visit vector $\mathbf{x}_{i,\ell}$ is therefore replaced by a corrupted version $\tilde{\mathbf{x}}_{i,\ell}$ in the same V-dimensional ICD-10 input space. The model receives the history up to visit $\ell - 1$ together with the corrupted target visit $\tilde{\mathbf{x}}_{i,\ell}$, and produces masked-visit reconstruction logits $\hat{\mathbf{y}}_{i,\ell}^{\text{mask}}$ for the original, uncorrupted multi-label diagnosis vector at visit ℓ.

Although supervision is enriched for visits containing new diagnoses, the prediction target remains the full diagnosis vector. The model outputs probabilities over all ICD-10 codes ($V = 1{,}538$), predicting which codes are present at the selected visit rather than only which codes are newly incident. The incident-code distinction is applied only at evaluation time to assess early case-finding utility. We evaluated $p_{\text{visit}} \in \{0.15, 0.30, 0.40, 0.60, 0.80\}$ while keeping the within-visit 80/10/10 corruption scheme fixed (Ap-

pendix H). Performance differences were not statistically significant, but larger sampling probabilities increased computation by up to 41% due to slower convergence. We therefore use $p_{\text{visit}} = 0.15$ as the default.

For threshold selection, calibration, held-out testing, and inference, no target-visit sampling, corruption, or masked-visit reconstruction is applied. Given the observed history up to visit ℓ, the model produces test-time forecasting logits $\hat{\mathbf{y}}^{\text{test}}_{i,\ell+1}$ for the complete diagnosis vector at the next visit $\ell+1$.

3.4. Dependency-Aware Loss

The proposed training objective operates on masked target visits (Section 3.3). For each valid entry $(b,\ell,v) \in \mathcal{S}$, the model outputs logits $\hat{y}^{b,\ell,v} \in \mathbb{R}$ and probabilities $\hat{p}^{b,\ell,v} = \sigma(\hat{y}^{b,\ell,v})$ with targets $y^{b,\ell,v} \in \{0,1\}$.

Imbalance-aware correctness. We use a bounded, differentiable per-code correctness score $c^{b,\ell,v} \in [0,1]$, and then re-weight it within each mini-batch to counter extreme sparsity.

$$c^{b,\ell,v} = 1 - \left|\hat{p}^{b,\ell,v} - y^{b,\ell,v}\right| = \begin{cases} \hat{p}^{b,\ell,v}, & y^{b,\ell,v} = 1, \\ 1 - \hat{p}^{b,\ell,v}, & y^{b,\ell,v} = 0 \end{cases}, \tag{1}$$

and re-scale it on each mini-batch to counter extreme sparsity. Let

$$P = \sum_{(b,\ell,v)\in\mathcal{S}} y^{b,\ell,v}, \qquad N = \sum_{(b,\ell,v)\in\mathcal{S}} \left(1 - y^{b,\ell,v}\right), \tag{2}$$

where $\mathcal{S} = \{(b,\ell,v) : y^{b,\ell,v} \in \{0,1\}\}$ denotes valid positions. We define

$$w_{\text{pos}} = \frac{N}{P+\epsilon}, \qquad w_{\text{neg}} = \alpha_{\text{neg}} \frac{P}{N+\epsilon}, \tag{3}$$

where $\epsilon > 0$ ensures numerical stability and $\alpha_{\text{neg}} > 0$ tunes the relative weight on negatives, thereby controlling the precision–recall trade-off. The class-weighted correctness is

$$\tilde{c}^{b,\ell,v} = \begin{cases} w_{\text{pos}}\, c^{b,\ell,v}, & y^{b,\ell,v} = 1, \\ w_{\text{neg}}\, c^{b,\ell,v}, & y^{b,\ell,v} = 0 \end{cases}. \tag{4}$$

The raw correctness score $c^{b,\ell,v}$ lies in $[0,1]$. After class reweighting, however, $\tilde{c}^{b,\ell,v}$ is an optimisation score and is not constrained to lie in $[0,1]$. Consequently, the aggregate score below should be interpreted as a training objective quantity, not as a calibrated probability or bounded accuracy measure.

Rank-weighted (listwise) aggregation. Within a visit (b,ℓ), we sort the valid values $\{\tilde{c}^{b,\ell,v}\}_{v\in\mathcal{V}^{b,\ell}}$ in descending order $\tilde{c}^{b,\ell}_{(1)} \geq \cdots \geq \tilde{c}^{b,\ell}_{(|\mathcal{V}^{b,\ell}|)}$ and compute an ordered weighted average that emphasises the lowest class-weighted correctness entries:

$$\mathcal{C}^{b,\ell} = \sum_{i=1}^{|\mathcal{V}^{b,\ell}|} \frac{2i}{|\mathcal{V}^{b,\ell}|\left(|\mathcal{V}^{b,\ell}|+1\right)}\, \tilde{c}^{b,\ell}_{(i)}. \tag{5}$$

This listwise weighting counters gradient dilution by assigning the largest coefficients to the smallest class-weighted correctness entries.

Label-space interaction. Element-wise losses optimise each label independently. Unlike the flexible multi-label dependence losses surveyed by Hüllermeier et al. (2022), which primarily study structured dependence modelling in the output distribution, the proposed objective is architecture-compatible and optimisation-driven, combining imbalance-aware correctness, rank-weighted aggregation, and a lightweight learned pairwise compatibility term over per-label logits. To encourage output-space compatibility among predicted labels, we add a learned quadratic interaction on the unsorted vector $\tilde{\mathbf{c}}^{\,b,\ell} = (\tilde{c}^{b,\ell,v})_{v=1}^{V}$:

$$\begin{aligned} \mathcal{V}^{b,\ell} &= \{v : y^{b,\ell,v} \in \{0,1\}\}, \\ \text{raw}^{b,\ell} &= \frac{1}{|\mathcal{V}^{b,\ell}|}\, (\tilde{\mathbf{c}}^{\,b,\ell})^{\top} \mathbf{W}\, \tilde{\mathbf{c}}^{\,b,\ell}, \\ s^{b,\ell} &= \sigma\big(\text{raw}^{b,\ell}\big), \\ \mathcal{I}^{b,\ell} &= 1 - s^{b,\ell}. \end{aligned} \tag{6}$$

where $|\mathcal{V}^{b,\ell}| = \sum_{v=1}^{V} \mathbb{I}[y^{b,\ell,v} \in \{0,1\}]$ is the number of valid non-padding labels for that target visit. Because $\mathbf{W}$ is symmetric with zero diagonal,

$$(\tilde{\mathbf{c}}^{\,b,\ell})^{\top} \mathbf{W} \tilde{\mathbf{c}}^{\,b,\ell} = 2 \sum_{u<v} W_{uv} \tilde{c}^{b,\ell,u} \tilde{c}^{b,\ell,v}.$$

Thus, each unordered label pair is counted twice. This constant factor is absorbed into the tuned interaction strength λ_{reg}. Equivalently, one could include a factor of $1/2$ in the quadratic term without changing the qualitative form of the objective.

The TopK attention layer models relationships between codes in the representation space, producing a permutation-invariant summary of the active diagnoses within each visit. The label-space interaction

term plays a different role. Whereas standard multi-label objectives such as BCE, focal loss, and class-balanced loss treat output labels independently, the quadratic term $(\tilde{\mathbf{c}}^{\,b,\ell})^{\top}\mathbf{W}\,\tilde{\mathbf{c}}^{\,b,\ell}$ introduces explicit coupling between label-level optimisation signals. This encourages predictions to follow a learned output-space compatibility structure, reducing isolated weak positive predictions that can contribute to precision collapse under extreme imbalance. The resulting matrix $\mathbf{W}$ is also directly inspectable. High-weight entries can be audited as predictive compatibilities under the training objective, rather than as causal effects or empirical co-occurrence frequencies. We do not supervise W with empirical co-occurrence, as this would impose a prevalence-biased dependency prior rather than a task-specific predictive compatibility structure. We parameterise $\mathbf{W}$ as a free matrix and enforce symmetry with a zero diagonal at each forward pass, so that $\mathbf{W} = \mathbf{W}^{\top}$ and $\mathrm{diag}(\mathbf{W}) = \mathbf{0}$ throughout training. The matrix is regularised using decoupled weight decay through AdamW.

Final objective. For each selected masked target visit, the dependency-aware objective is

$$\mathcal{L}^{b,\ell} = 1 - \mathcal{C}^{b,\ell} + \lambda_{\mathrm{reg}}\,\mathcal{I}^{b,\ell}. \tag{7}$$

Here, $\mathcal{C}^{b,\ell}$ denotes the class-weighted correctness term, $\mathcal{I}^{b,\ell}$ denotes the label-space interaction regulariser, and $\lambda_{\mathrm{reg}} \geq 0$ controls the strength of this regularisation. We minimise the average objective over selected masked training targets. Since $\mathcal{C}^{b,\ell}$ is a weighted correctness score rather than a log-likelihood, $\mathcal{L}^{b,\ell}$ is used as an optimisation objective and is not constrained to be non-negative. Appendix I provides the full formulation.

3.5. Benchmarking Loss Functions

We compare the proposed dependency-aware objective against three standard element-wise losses commonly used for imbalanced multi-label prediction: weighted BCE, class-balanced loss (Cui et al., 2019), and focal loss (Lin et al., 2017). All losses are evaluated under the same encoder, optimiser, masking strategy, data splits, and threshold-selection protocol. Only the training objective is changed. Weighted BCE serves as the standard baseline for sparse multi-label diagnosis prediction. It applies code-specific positive weights so that rare positive labels contribute more strongly to the loss. Class-balanced loss assigns a code-level weight based on the effective number of positive examples observed in the mini-batch. In our implementation, this code-level weight is applied to all valid entries for that code. Focal loss further downweights easy examples and emphasises hard or misclassified labels through a focusing term. For fairness, all baseline weights are estimated after excluding padded labels, and all objectives are averaged over the same valid label entries. Full mathematical definitions, masking details, and numerical-stability choices are provided in Appendix J. In the results, weighted BCE is denoted as BCE for brevity.

Empirically, these objectives differ in how they respond to extreme sparsity. In our experiments, weighted BCE produced a more conservative operating point, whereas class-balanced and focal losses increased sensitivity to rare or hard labels but also tended to increase false positives. The proposed dependency-aware loss is designed to target an intermediate operating regime by combining imbalance-aware correctness, rank-weighted aggregation over hard labels, and an explicit output-space dependency regulariser.

4. Experiments

4.1. Model Training and Hyperparameter Tuning

All models are trained with AdamW (Loshchilov and Hutter, 2017) using a learning rate of 1×10^{-4} and weight decay of 1×10^{-4}. We monitor validation loss for early stopping and use a ReduceLROnPlateau schedule that halves the learning rate when validation loss does not improve for three consecutive epochs. Training uses mixed precision for efficiency. Detailed hyperparameters and architectural configurations are provided in Appendix E.

We evaluate all loss functions (BCE, focal, class-balanced, and dependency-aware) under an identical backbone architecture (Section 3.2), optimiser, masking strategy (Section 3.3), and data splits. We perform patient-level 3-fold cross-validation. In each fold, one fold is held out for testing, and the remaining folds are used for training with a validation split for learning-rate scheduling, early stopping, and threshold selection. Validation loss for learning-rate scheduling and early stopping is computed using the same masked-visit denoising objective as training. Threshold selection and post-hoc calibration are performed

separately on the validation split using the uncorrupted next-visit forecasting protocol used at held-out test time. To account for optimisation variance, we repeat each fold with three random seeds and report 95% CI/mean(SD) across the resulting nine runs.

Hyperparameters of the dependency-aware loss are tuned and described in Appendix F. Specifically, we sweep α_{neg}, which controls the relative weight on negatives and therefore the precision–recall trade-off, and λ_{reg}, which scales the label-interaction regulariser. We additionally perform ablations by removing individual loss components to quantify their contributions in Appendix K. All tuned settings are then fixed for the larger-scale experiments on 200,000 and 3.2 million patients.

To test architectural compatibility beyond the Transformer, we repeat key comparisons with a temporal convolutional network (TCN) backbone and observe the same qualitative trends (Appendix L). We also examine recall as a function of available history length at a fixed decision threshold (Appendix M).

4.2. Evaluation Metrics

We evaluate performance using micro-averaged precision, recall, F_β, balanced accuracy, and the area under the precision–recall curve (PRC-AUC; average precision). Given the extreme sparsity of the label space, we prioritise metrics that remain informative when negatives vastly outnumber positives.

Threshold-free evaluation (PRC-AUC vs. AUROC). Under severe class imbalance, AUROC can appear high even when precision is low, because it is dominated by true negatives (Davis and Goadrich, 2006). We therefore report PRC-AUC (average precision) as the primary threshold-independent measure of ranking quality.

Threshold-dependent metrics and operating point. For point metrics (precision, recall, F_β, balanced accuracy), we evaluate at a single decision threshold τ chosen on the validation split and then held fixed on the test split. To define a clinically conservative operating point under extreme imbalance, we select τ to control false positives on validation (high-specificity setting) and apply the same τ for all reported point metrics. We additionally report $F_{\beta=2}$ to summarise the precision–recall trade-off with increased emphasis on recall.

Contextualising absolute performance. A typical visit contains 2 (Q1) to 8 (Q3) true codes out of V=1,538, i.e. only $\approx$ 0.13%–0.52% of labels are positive per visit. If a random predictor selects k codes uniformly and the visit contains M true codes, then its expected precision is M/V and its expected recall is k/V. For $M = 2$ to $M = 8$, the expected random precision is therefore approximately 0.0013–0.0052, with balanced accuracy close to 0.50. Accordingly, precision values in the 10%–15% range correspond to substantial enrichment over random selection under extreme label imbalance.

Statistical significance. We assess significance using a paired permutation (randomization) test on matched fold–seed runs (N=9 pairs). P-values are estimated from 10,000 random sign-flips of the paired score differences, yielding a nonparametric test that does not assume normality (Noguchi et al., 2021; DiCiccio and Efron, 1996).

Post-hoc calibration analysis. Because discrimination does not guarantee well-calibrated probabilities, we evaluated calibration of the predicted per-code risks (Guo et al., 2017). The dependency-aware objective is designed primarily for discrimination and operating-point behaviour, so we applied post-hoc isotonic calibration. An isotonic regressor was fitted on the validation split only, using pooled multilabel predictions and binary outcomes across all valid visit–code entries, and then applied unchanged to the held-out test split. We report expected calibration error, Brier score, and log loss on the test set, with additional inspection of the higher-probability region because the label space is extremely sparse and most predictions are near zero.

5. Results

Impact of Loss Function on Learning Dynamics

We first examine optimisation stability under extreme label sparsity. Figure 2 shows the training and validation loss trajectories together with validation balanced accuracy. Because the four objectives use different weighting and scaling schemes, raw loss values are not directly comparable across methods. We therefore use balanced accuracy as a common diagnostic of learning progress. All methods are trained with the same optimisation protocol: AdamW with learning

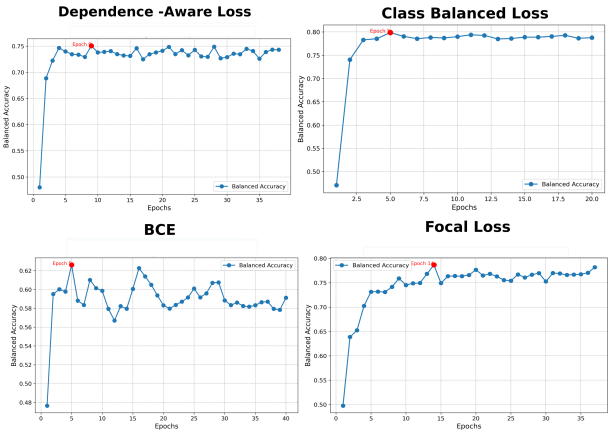

Figure 2: Comparison of Model Training Stability Using BCE Loss and Dependency-Aware Loss Across Different Folds.

rate 1×10^{-4}, weight decay 1×10^{-4}, and ReduceLROnPlateau scheduling. Under this fixed protocol, the BCE baseline shows greater volatility in validation balanced accuracy across epochs. In contrast, the dependency-aware, focal, and class-balanced losses exhibit smoother convergence and more stable validation plateaus. This pattern suggests that per-batch reweighting and output-space coupling can reduce sensitivity to extreme within-batch prevalence fluctuations.

Performance comparison: overall prediction and new-code detection. Table 1 reports standard present-versus-absent metrics together with a dedicated new-code detection row. On balanced accuracy (new-code detection), the dependency-aware loss reaches 0.738, substantially above BCE (0.598), while focal and class-balanced losses achieve higher values (0.857 and 0.785) through a much more recall-biased operating regime.

Across the standard metrics, the objectives show distinct precision–recall profiles. Focal and class-balanced losses achieve very high sensitivity (0.947 and 0.959), but precision falls to 0.020 and 0.009. BCE gives the highest precision (0.212), but much lower sensitivity (0.247), meaning that many positive diagnoses are missed. The dependency-aware loss provides the best middle ground. It more than doubles sensitivity relative to BCE (0.526 vs. 0.247), retains much higher precision than focal and class-balanced losses (0.119), and achieves the best F_2 score (0.311). Its PRC-AUC (0.163) is comparable to BCE (0.170, $p > 0.05$) and higher than focal (0.103) and class-balanced loss (0.129, $p < 0.05$; Figure 3). Overall, the proposed objective improves the selected operating point while preserving competitive threshold-free ranking quality.

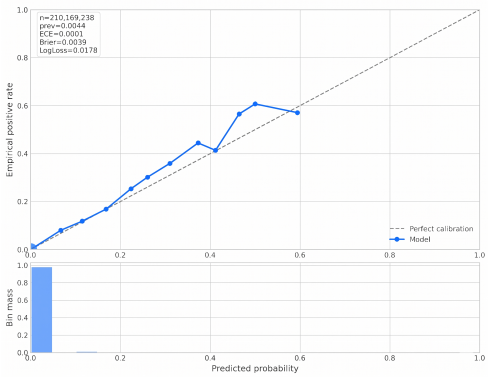

Figure 4: Reliability diagram for post-hoc isotonic-calibrated next-visit code predictions on the held-out test set.

Probability calibration. Because ranking performance alone does not ensure reliable probability estimates, we evaluated post-hoc calibration for the dependency-aware model. In our multilabel next-visit diagnosis setting, absolute risk corresponds to the probability that a given ICD-10 code occurs at the next visit. We fitted an isotonic regressor on the validation set and applied it unchanged to the untouched test set. After calibration, the reliability curve lies close to the 45° line over most of the prediction mass, whereas the raw scores are overconfident in the higher-probability region (Figure 4). The calibrated probabilities achieve ECE 0.00015, Brier score 0.00388, and log loss 0.0178 overall. Among predictions with estimated probability at least 0.5, ECE remains low at 0.0328. These results show that the proposed objective can be paired with reliable absolute risk estimates through a simple validation-only calibration step.

Scalability. We further validated the objective on larger cohorts to assess stability in separate runs with re-formed train/validation/test splits at each scale. On 200,000 patients, the model achieved a balanced accuracy of 0.75 (95% CI 0.73–0.77) and improved ranking quality (PRC-AUC 0.19). Scaling to the full 3.2-million patient cohort maintained similar performance, yielding a balanced accuracy of 0.79 (95% CI

Table 1: Comparison of Loss Functions on Key Metrics (95% CI). **Bold** indicates the best performance in a row; † indicates the lowest performance. Note that the dependency-aware loss is the only objective that never yields the lowest performance across any metric, demonstrating better stability. See Appendix G for complete definitions.

Metric	Dependency-aware loss	BCE	Class balanced loss	Focal loss
Balanced Accuracy (present vs absent)	0.754 (0.747–0.762)	0.621 (0.603–0.639)†	0.784 (0.779–0.788)	**0.856 (0.851–0.860)**
Balanced Accuracy (new-code detection)	0.738 (0.729–0.746)	0.598 (0.583–0.613)†	0.785 (0.779–0.791)	**0.857 (0.853–0.861)**
Recall (Class 1) / Sensitivity	0.526 (0.511–0.540)	0.247 (0.211–0.283)†	**0.959 (0.948–0.970)**	0.947 (0.936–0.957)
Recall (Class 0) / Specificity	0.986 (0.982–0.989)	**0.999 (0.999–0.999)**	0.593 (0.560–0.627)†	0.763 (0.750–0.776)
Precision	0.119 (0.108–0.130)	**0.212 (0.164–0.261)**	0.009 (0.006–0.011)†	0.020 (0.010–0.021)
F_2 Score	**0.311 (0.293–0.330)**	0.235 (0.194–0.275)	0.043 (0.030–0.055)†	0.092 (0.074–0.098)
PRC-AUC	0.163 (0.149–0.177)	**0.170 (0.158–0.182)**	0.129 (0.117–0.141)	0.103 (0.086–0.121)†

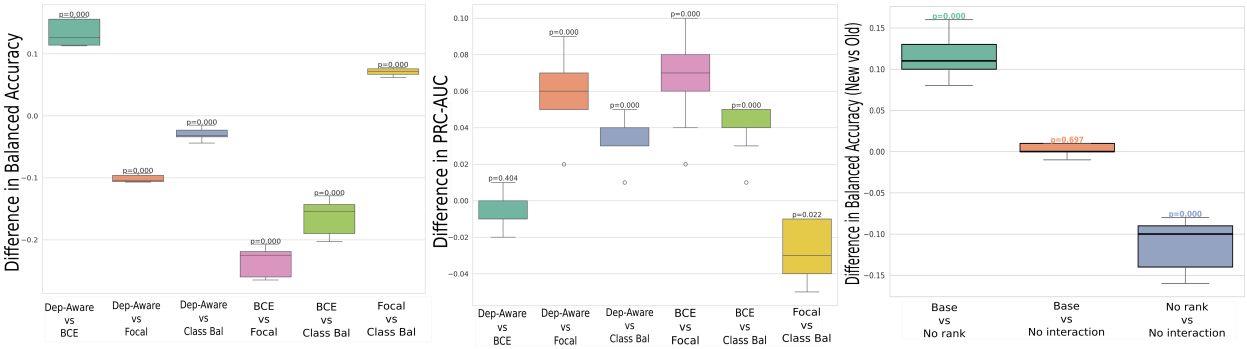

Figure 3: Pairwise Differences Between Loss Functions and Ablation of Dependency-Aware Components.

0.77–0.82) and PRC-AUC of 0.19, indicating stable behaviour at population scale.

Ablation study: disentangling ranking and interaction

To identify the source of the gains, we ablate the two core components of the dependency-aware loss (Appendix K). Removing rank-weighted aggregation (Eq. 5) produces a pronounced collapse in operating-point behaviour, with precision dropping from 0.119 to 0.010 and balanced accuracy dropping from 0.738 to 0.623 (with significant paired differences, $p < 0.05$). This supports the interpretation that prioritising the lowest-correctness labels through rank weighting is the dominant mechanism that preserves precision under extreme sparsity. Removing the interaction regulariser (Eq. 6) yields smaller changes in aggregate metrics ($p > 0.05$), consistent with its role in targeting sparse pairwise structure rather than visit-level balanced accuracy. Nonetheless, removing it consistently reduces recall. We therefore retain it in the full objective. It injects an explicit compatibility bias that discourages medically incoherent co-predictions and yields an auditable dependency matrix, without materially degrading balanced accuracy at the selected operating point.

Qualitative analysis of learned dependencies.

A key advantage of the proposed approach is global transparency. Figure 5 visualises the learned dependency matrix $\mathbf{W}$. Unlike attention mechanisms, which are instance-specific and operate in latent representation space, $\mathbf{W}$ is a single population-level parameter acting in the output space through Eq. (6). It summarises which label co-predictions the objective encourages or discourages. Because $\mathbf{W}$ is symmetric ($\mathbf{W} = \mathbf{W}^\top$) with a zero diagonal, it captures undirected associations rather than directional or causal effects. Qualitatively, the strongest couplings align with recognizable clinical and coding associations. Chronic kidney disease (N18.5) has high positive compatibility with malignancy-related codes (C*), consistent with the documented CKD–cancer interrelationship (Hu et al., 2022). Diabetes–renal structure is also recovered: type 2 diabetes with complications (E11.6) couples to nephropathy/CKD diagnoses, aligning with diabetic kidney disease pathways (Bakris, 2011). Endocrine–hepatic interactions also ap-

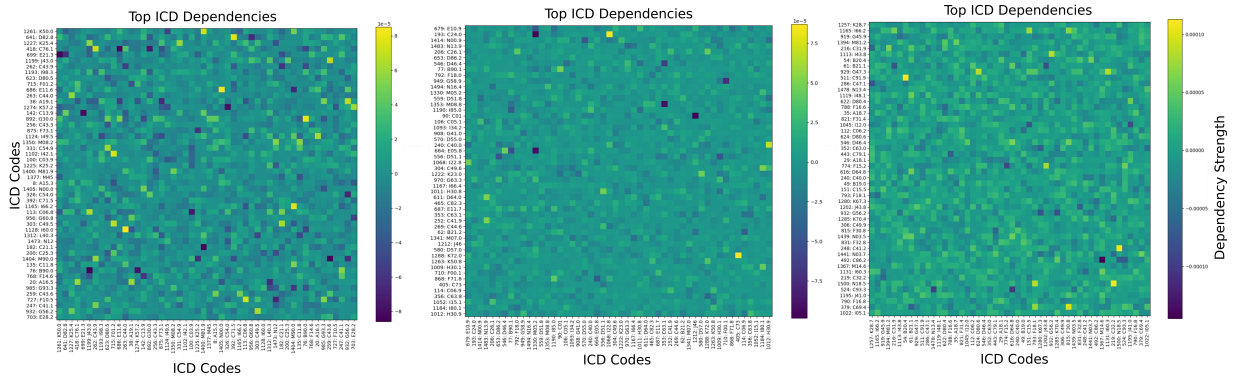

Figure 5: Heatmaps of High-Confidence ICD-10 Code Dependencies

pear, with thyroid-related codes (e.g., E05.* / C73) co-occurring with hepatic dysfunction (e.g., K72.*), consistent with the thyroid–liver axis (Piantanida et al., 2020). Finally, post-tuberculosis sequelae (B90.*) couple with dementia/Alzheimer's codes (e.g., F00.* / G30.0), matching cohort evidence of increased dementia risk after TB (Peng et al., 2015).

To assess whether the learned dependencies reflect broader structure rather than isolated high-weight pairs, we clustered the learned output-space matrix $\mathbf{W}$ over the full label space using the final 3.2M-patient checkpoint. The clustered heatmap in Figure 10 shows coherent block structure, indicating that the dependency term learns organised diagnosis communities rather than diffuse pairwise noise. Table 6 provides readable examples of high-weight positive edges from the same matrix. These recover recognizable clinical and coding patterns, including type 2 diabetes–hypertension, depression–anxiety, and diabetes–retinopathy. Spectral clustering further identifies interpretable communities, including a coronary disease cluster containing I20., I21.9, and I25. codes, and a diabetes/complications cluster containing E10.1, E10.9, and H36.0. Together, the results support the intended interpretation of $\mathbf{W}$ as an auditable output-space compatibility matrix. The learned structure allows domain experts to inspect coherent and unexpected label relationships without aggregating millions of instance-level attention maps.

Computational efficiency Despite the additional interaction term, the dependency-aware loss remains computationally efficient. On an NVIDIA A100 (80GB) and on the 5,000-patient development set (same backbone and training configuration across objectives), average training time per epoch differed by about 10% across objectives (6.0 min dependency-aware vs. 6.5 min BCE and 5.9 min focal). The loss implementation is fully vectorised and avoids explicit loops over the label space, so the dominant cost remains the Transformer backbone rather than loss evaluation when scaling to larger cohorts.

6. Discussion and Conclusion

This study systematically evaluated training objectives for forecasting multi-label ICD-10 diagnoses under extreme sparsity. Across matched folds and seeds, focal and class-balanced objectives mitigate positive-signal dilution and maximise recall, but they do so with precision below 2%, implying that the majority of flagged codes are false positives. In clinical decision support, such alert volumes are difficult to operationalise and can contribute to alarm fatigue, undermining trust and uptake even when sensitivity is high (Cvach, 2012; Ancker et al., 2017). Standard BCE exhibits the opposite limitation. It is comparatively precise but misses a substantial fraction of true diagnoses. The primary contribution of this work is therefore not a uniform improvement on every metric, but an objective that shifts the operating regime toward a more favourable precision–recall balance. It improves recall over BCE while avoiding the extreme precision collapse of other imbalance-handling losses.

Mechanistically, the ablation study indicates that the rank-weighted aggregation is the dominant driver of this operating-point behaviour. By emphasising the lowest-correctness (hardest) labels within each target visit, the loss counteracts gradient dilution caused by the overwhelming mass of easy negatives. This

introduces an inductive bias closer to listwise ranking, prioritising relative separation between active codes and a large negative background, rather than treating each label as an independent Bernoulli objective with uniform aggregation (Burges et al., 2005; Cao et al., 2007). This perspective is consistent with the empirical pattern that focal and class-balanced objectives can inflate recall by amplifying weak signals broadly, while degrading precision in extremely sparse, high-dimensional label spaces.

The quadratic interaction term provides a complementary form of dependency awareness. Transformer encoders such as BEHRT and Med-BERT can capture correlations implicitly in representation space via self-attention (Li et al., 2020; Rasmy et al., 2021), yet the training objectives are typically element-wise and provide no explicit coupling between output labels. The proposed interaction regulariser addresses this by coupling labels during optimisation through an explicit compatibility term. Although ablations indicate that rank-weighted aggregation is the primary driver of improvements in balanced accuracy, we retain the interaction term because it acts as a structural filter. It injects a compatibility bias into the gradients, discouraging medically incoherent co-predictions that aggregate metrics may not penalise. Moreover, it yields a single population-level dependency matrix $\mathbf{W}$ that is directly auditable (Lafferty et al., 2001; Zhang and Zhou, 2013). This transparency supports model auditing by allowing domain experts to inspect the comorbidity structure encouraged by the objective and to flag unexpected associations.

A key practical implication is that absolute precision values must be interpreted in the context of extreme base rates and evaluation granularity. Each visit is scored against $V{=}1538$ candidate labels, so even a modest micro-precision corresponds to a large enrichment over random guessing. At the same time, our results do not imply that any of the evaluated objectives are directly deployable as an autonomous alerting system without additional constraints. Instead, the proposed loss is most naturally suited to shortlist-style decision support (e.g., ranking and presenting a limited number of candidate codes under a fixed alert budget) or as the first stage in a multi-step pipeline with downstream filtering and calibration. Reporting PRC-AUC alongside thresholded point metrics is essential here, because AUROC can remain high under extreme imbalance even when precision is clinically unacceptable (Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015). Several limitations remain. The objective introduces additional hyperparameters that require tuning, and the learned dependency structure is a static population-level summary that cannot adapt to patient-specific subphenotypes. Our evaluation also focuses on long-term conditions. Generalisation to acute settings and external datasets remains to be established. We include a TCN compatibility check, but broader validation across backbones and multimodal settings remains an important next step. Future work should also evaluate explicit alert-budgeted operating points and more data-efficient parameterisations of W, such as low-rank or group-structured forms.

In summary, effective extreme multi-label diagnosis prediction requires not only expressive encoders but objectives aligned with the realities of sparse, correlated medical labels. Under severe imbalance, common losses occupy qualitatively different operating regimes. By prioritising hard labels via rank-weighted aggregation and introducing an explicit output-space compatibility bias with an auditable dependency matrix, the proposed loss moves the model toward a less extreme and more operationally useful precision–recall trade-off.

Acknowledgements

This work was supported by UK Research and Innovation (UKRI) through the Centre for Doctoral Training in Biomedical AI programme. We are grateful to Professor Ian Simpson, Programme Director of the CDT in Biomedical AI, for his support of this work and related conference travel. This work was also supported by the National Institute for Health and Care Research (NIHR) Artificial Intelligence and Multimorbidity: Clustering in Individuals, Space and Clinical Context (AIM-CISC) grant NIHR202639, which provided funding for data access. The views expressed are those of the authors and not necessarily those of the NIHR or the Department of Health and Social Care.

References

Jessica S Ancker, Alison Edwards, Sarah Nosal, Diane Hauser, Elizabeth Mauer, Rainu Kaushal, and With the HITEC Investigators. Effects of workload, work complexity, and repeated alerts on alert fatigue in

a clinical decision support system. *BMC medical informatics and decision making*, 17(1):1–9, 2017.

George L Bakris. Recognition, pathogenesis, and treatment of different stages of nephropathy in patients with type 2 diabetes mellitus. In *Mayo Clinic Proceedings*, volume 86, pages 444–456. Elsevier, 2011.

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, pages 1–38, 2020.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

Maria Cvach. Monitor alarm fatigue: an integrative review. *Biomedical instrumentation & technology*, 46(4):268–277, 2012.

Sajad Darabi, Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. Taper: Time-aware patient ehr representation. *IEEE journal of biomedical and health informatics*, 24(11):3268–3275, 2020.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Thomas J DiCiccio and Bradley Efron. Bootstrap confidence intervals. *Statistical science*, 11(3):189–228, 1996.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Ankit Gupta, Guy Dar, Shaya Goodman, David Ciprut, and Jonathan Berant. Memory-efficient transformers via top-k attention. *arXiv preprint arXiv:2106.06899*, pages 1–14, 2021.

Mengsi Hu, Qianhui Wang, Bing Liu, Qiqi Ma, Tingwei Zhang, Tongtong Huang, Zhimei Lv, and Rong Wang. Chronic kidney disease and cancer: inter-relationships and mechanisms. *Frontiers in Cell and Developmental Biology*, 10:868715, 2022.

Eyke Hüllermeier, Marcel Wever, Eneldo Loza Mencia, Johannes Fürnkranz, and Michael Rapp. A flexible class of dependence-aware multi-label loss functions. *Machine Learning*, 111(2):713–737, 2022.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 1–8, 2001.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR, 2019.

Seungyeon Lee, Ruoqi Liu, Feixiong Cheng, and Ping Zhang. A deep subgrouping framework for precision drug repurposing via emulating clinical trials on real-world patient data. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2347–2358, 2025.

Yikuan Li, Shishir Rao, José Roberto Ayala Solares, Abdelaali Hassaine, Rema Ramakrishnan, Dexter Canoy, Yajie Zhu, Kazem Rahimi, and Gholamreza Salimi-Khorshidi. Behrt: transformer for electronic health records. *Scientific reports*, 10(1):7155, 2020.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Alex Moore, Bastien Orset, Arrash Yassaee, Benjamin Irving, and Davide Morelli. Healthrecordbert (herbert): Leveraging transformers on electronic health records for chronic kidney disease risk stratification. *ACM Transactions on Computing for Healthcare*, 5 (3):1–18, 2024.

Kimihiro Noguchi, Frank Konietschke, Fernando Marmolejo-Ramos, and Markus Pauly. Permutation tests are robust and powerful at 0.5% and 5% significance levels. *Behavior Research Methods*, 53 (6):2712–2724, 2021.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Mathieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Yi-Hao Peng, Chih-Yu Chen, Ching-Hua Su, Chih-Hsin Muo, Kuan-Fei Chen, Wei-Chih Liao, and Chia-Hung Kao. Increased risk of dementia among patients with pulmonary tuberculosis: A retrospective population-based cohort study. *American Journal of Alzheimer's Disease & Other Dementias®*, 30(6):629–634, 2015.

E Piantanida, S Ippolito, D Gallo, E Masiello, P Premoli, C Cusini, S Rosetti, J Sabatino, S Segato, F Trimarchi, et al. The interplay between thyroid and liver: implications for clinical practice. *Journal of endocrinological investigation*, 43:885–899, 2020.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):1–13, 2021.

Maurice Rupp, Oriane Peter, and Thirupathi Pattipaka. Exbehrt: Extended transformer for electronic health records. In *International Workshop on Trustworthy Machine Learning for Healthcare*, pages 73–84. Springer, 2023.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:1–14, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:1–11, 2017.

Jiaqi Wang, Junyu Luo, Muchao Ye, Xiaochen Wang, Yuan Zhong, Aofei Chang, Guanjie Huang, Ziyi Yin, Cao Xiao, Jimeng Sun, et al. Recent advances in predictive modeling with electronic health records. *arXiv preprint arXiv:2402.01077*, 2024.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*, 2022.

Fan Yang, Jian Zhang, Wanyi Chen, Yongxuan Lai, Ying Wang, and Quan Zou. Deepmpm: a mortality risk prediction model using longitudinal ehr data. *BMC bioinformatics*, 23(1):1–27, 2022.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *31st Conference on Neural Information Processing System*, pages 1–11, 2017.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

Yujin Zhu, Zilong Zhao, Robert Birke, and Lydia Y Chen. Permutation-invariant tabular data synthesis. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5855–5864. IEEE, 2022.

Appendix A. Compute Resources

All experiments were run on a Kubernetes cluster with NVIDIA A100-SXM4 (80 GB) GPUs. Unless otherwise stated, runs used a single GPU. On the development set used for hyperparameter tuning and ablations, training required approximately 5 minutes per epoch (training pass only; validation/testing excluded). We found that the additional compute cost of the dependency-aware interaction term was negligible compared with the Transformer backbone, adding $< 10\%$ overhead in step time relative to BCE under the same configuration.

Appendix B. Binary structured EHR input and TopK attention

Binary structured EHR data. Our training samples consist of binary visit-level vectors denoting the presence or absence of long-term-condition ICD-10 codes recorded at a given healthcare visit. After preprocessing (Section 3.1), data are provided to the model as a binary input tensor

$$\mathbf{I} \in \{0,1\}^{B\times L\times V},$$

where B is the batch size, L is the maximum number of visits per patient, and V is the size of the ICD-10 long-term-condition vocabulary. The same V-dimensional ICD-10 input space is used for corrupted training inputs. Corruption is implemented by modifying entries within this ICD-code vector: selected active codes may be removed, replaced by randomly sampled ICD-10 codes, or left unchanged. No additional prediction label is introduced. For each patient b and visit ℓ, let

$$X_{b,\ell} = \{v : I_{b,\ell,v} = 1\}$$

denote the set of active conditions at that visit.

Input to TopK attention. Each active condition g_v is expanded into a D-dimensional embedding using a learned condition-embedding matrix $\mathbf{E} \in \mathbb{R}^{V\times D}$. In tensor form,

$$\mathbf{H} \in \mathbb{R}^{B\times L\times V\times D}, \qquad H_{b,\ell,v,d} = I_{b,\ell,v}E_{v,d}.$$

The condition dimension is preserved, so relationships between individual long-term conditions can be modelled within each visit. Attention is applied over the active set $X_{b,\ell}$, rather than over absent condition slots.

TopK attention. To model within-visit relationships between co-occurring conditions, we adopt the TopK attention mechanism of Gupta et al. (2021). For notational simplicity, the equations below show one attention head with per-head query/key dimension d_h; the implementation uses six heads.

For visits with $|X_{b,\ell}| > 0$, the embedding tensor $\mathbf{H}$ is linearly projected into query, key, and value tensors. For one head,

$$Q_{b,\ell,v,d} = \sum_{d'} H_{b,\ell,v,d'}W_{Q,d',d},$$
$$K_{b,\ell,v,d} = \sum_{d'} H_{b,\ell,v,d'}W_{K,d',d},$$
$$V_{b,\ell,v,d} = \sum_{d'} H_{b,\ell,v,d'}W_{V,d',d},$$

where $d = 1,\dots,d_h$ and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D\times d_h}$ for each head.

For active query and key conditions $v, v' \in X_{b,\ell}$, the unnormalised attention score is

$$A_{b,\ell,v,v'} = \frac{1}{\sqrt{d_h}} \sum_{d=1}^{d_h} Q_{b,\ell,v,d}K_{b,\ell,v',d}.$$

Let $\mathrm{Top}_\kappa(b,\ell,v)$ denote the indices of the κ largest scores among active key conditions $v' \in X_{b,\ell}$ for query condition v. If fewer than κ active conditions are available, all active conditions are retained. The masked attention scores are then

$$A_{\mathrm{top},b,\ell,v,v'} = \begin{cases} A_{b,\ell,v,v'}, & v' \in \mathrm{Top}_\kappa(b,\ell,v), \\ -\infty, & v' \in X_{b,\ell} \setminus \mathrm{Top}_\kappa(b,\ell,v). \end{cases}$$

so that non-selected active conditions receive zero mass after the softmax. Equivalently, the softmax is computed only over the selected active set. Each active condition embedding is refined as

$$O_{b,\ell,v,d} = \sum_{v'\in X_{b,\ell}} \mathrm{Softmax}_{v'}\left(A_{\mathrm{top},b,\ell,v,v'}\right) V_{b,\ell,v',d}, \qquad v \in X_{b,\ell}.$$

Finally, the visit-level embedding is obtained by averaging over the active conditions:

$$H_{\mathrm{visit},b,\ell,d} = \begin{cases} \dfrac{1}{|X_{b,\ell}|} \displaystyle\sum_{v\in X_{b,\ell}} O_{b,\ell,v,d}, & |X_{b,\ell}| > 0, \\ 0, & |X_{b,\ell}| = 0. \end{cases}$$

Visits with $|X_{b,\ell}| = 0$ are excluded from downstream temporal attention using the visit mask. This corresponds to the visit-level vectors $\mathbf{v}_{i,\ell}$ in Section 3.2.

Appendix C. Visit Embeddings

For each patient b with up to L visits we produce one visit embedding $\mathbf{h}_{b,\ell}^{\text{visit}} \in \mathbb{R}^{256}$ per visit index ℓ. At visit ℓ, we first compute five channels, each in $\mathbb{R}^{256}$:

$$\mathbf{h}_{b,\ell}^{\text{visit}} = \mathbf{h}_{b,\ell}^{\text{ICD}} + \mathbf{h}_{b,\ell}^{\text{pos}} + \mathbf{h}_{b,\ell}^{\text{age}} + \mathbf{h}_{b,\ell}^{\text{sex}} + \mathbf{h}_{b,\ell}^{\text{ival}}, \qquad \mathbf{h}_{b,\ell}^{(\cdot)} \in \mathbb{R}^{256}.$$

The code-level TopK attention uses six heads, and the visit-level FAVOR+ encoder likewise uses six fast-attention heads operating over $\mathbf{H}^{\text{visit}} \in \mathbb{R}^{B\times L\times 256}$.

Components of a visit vector

- **ICD-10 code attended embedding** (256 dim) — present ICD–10 codes receive 256-dim token vectors via TopK multi-head attention, producing $\mathbf{h}_{b,\ell}^{\text{ICD}} \in \mathbb{R}^{256}$.
- **Position embedding** (256 dim) — we maintain a learned lookup table of length $L_{\max} = 96$ ((Devlin et al., 2019)). At run-time we look up the 256-dim vector for visit index ℓ and add it to the other channels. If longer sequences are ever needed, this slot can be swapped to sinusoidal (Vaswani et al., 2017) or rotary embeddings (Su et al., 2024).
- **Age embedding** (256 dim) — the raw age in years at visit ℓ of patient b is mapped to a 256-dim vector and added:
$$\mathbf{h}_{b,\ell}^{\text{age}} \in \mathbb{R}^{256}.$$
- **Sex embedding** (256 dim) — the binary code $\text{sex}_{b,\ell} \in \{0, 1\}$ is mapped to a 256-dim vector and added:
$$\mathbf{h}_{b,\ell}^{\text{sex}} \in \mathbb{R}^{256}.$$
- **Visit time interval embedding** (256 dim) — let $\Delta t_{b,\ell} = t_{b,\ell} - t_{b,\ell-1}$ be the days since the previous visit; we map $\Delta t_{b,\ell}$ to a 256-dim vector and add it:
$$\mathbf{h}_{b,\ell}^{\text{ival}} \in \mathbb{R}^{256}.$$

We then use addition

$$\mathbf{h}_{b,\ell}^{\text{visit}} = \mathbf{h}_{b,\ell}^{\text{ICD}} + \mathbf{h}_{b,\ell}^{\text{pos}} + \mathbf{h}_{b,\ell}^{\text{age}} + \mathbf{h}_{b,\ell}^{\text{sex}} + \mathbf{h}_{b,\ell}^{\text{ival}} \in \mathbb{R}^{256},$$

and stack over b, ℓ to get $\mathbf{H}^{\text{visit}} \in \mathbb{R}^{B\times L\times 256}$, which is fed into the transformer encoder.

Appendix D. Fast attention Mechanism for Visit-Level Embeddings

Extracting temporal and clinical patterns from EHRs requires attention layers that are both scalable and causal. Conventional multi-head attention (Vaswani et al., 2017) can capture rich dependencies, but dense attention has quadratic cost in the number of visits and, if used without a causal mask, allows future visits to influence earlier representations. We therefore use FAVOR+ (Choromanski et al., 2020), which approximates softmax attention with positive random features and can be implemented in linear time with causal prefix sums.

D.1. From dense softmax attention to random-feature attention

Let $\mathbf{H}_{\text{visit}} \in \mathbb{R}^{B\times L\times D}$ denote the sequence of visit-level embeddings. For clarity, the equations below omit the batch and head indices and describe a single attention head. The implementation uses standard multi-head FAVOR+ attention. Given a visit sequence $H \in \mathbb{R}^{L\times D}$, learned projections produce queries, keys, and values:

$$Q = HW_Q, \qquad K = HW_K, \qquad V = HW_V,$$

with

$$W_Q \in \mathbb{R}^{D\times d}, \qquad W_K \in \mathbb{R}^{D\times d}, \qquad W_V \in \mathbb{R}^{D\times d_v},$$

so that

$$Q, K \in \mathbb{R}^{L\times d}, \qquad V \in \mathbb{R}^{L\times d_v}.$$

Dense softmax attention computes

$$\text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V,$$

where d is the per-head query/key dimension. The $L \times L$ attention matrix is the quadratic bottleneck.

FAVOR+ uses a positive random feature map $\phi : \mathbb{R}^d \to \mathbb{R}^r$ to approximate the softmax kernel:

$$\exp\left(\frac{q^\top k}{\sqrt{d}}\right) \approx \phi(q)^\top \phi(k),$$

where r is the number of random features. We follow Choromanski et al. (2020) for the exact positive orthogonal random feature construction used to approximate the softmax kernel. With $\Phi_Q = \phi(Q) \in \mathbb{R}^{L\times r}$

and $\Phi_K = \phi(K) \in \mathbb{R}^{L\times r}$, the non-causal linear attention form can be written as

$$A(Q,K,V) \approx Z^{-1}\Phi_Q\left(\Phi_K^\top V\right),$$

where the diagonal normalizer is

$$Z = \operatorname{diag}\left[\Phi_Q\left(\Phi_K^\top \mathbf{1}_L\right)\right].$$

This avoids forming the dense $L \times L$ attention matrix, reducing the attention computation to linear dependence on the sequence length for fixed feature dimension r.

D.2. Causality via prefix sums

Medical forecasting requires each visit representation to depend only on the information available up to that visit. FAVOR+ implements causal linear attention by replacing global sums with prefix sums. Let $\phi(q_\ell)$ and $\phi(k_\ell)$ denote the random-feature query and key vectors at visit ℓ. Define

$$S_\ell = \sum_{i=1}^{\ell} \phi(k_i)v_i^\top \in \mathbb{R}^{r\times d_v}, \qquad z_\ell = \sum_{i=1}^{\ell}\phi(k_i) \in \mathbb{R}^r.$$

The causal attention output at visit ℓ is

$$o_\ell = \frac{\phi(q_\ell)^\top S_\ell}{\phi(q_\ell)^\top z_\ell + \varepsilon},$$

with a small $\varepsilon > 0$ used for numerical stability. This form ensures that the representation at visit ℓ depends only on visits $1,\ldots,\ell$ and not on future visits.

In the full model, the TopK ICD attention module first produces the visit embeddings $\mathbf{H}_{\text{visit}}$. These embeddings are then passed through multi-head causal FAVOR+ attention, the head outputs are concatenated and projected, and the resulting sequence is passed to the next Transformer block. The encoder is therefore linear in the number of visits, memory-efficient for long EHR histories, and consistent with the past-only forecasting setup used throughout the paper.

Appendix E. Model configuration

All experiments reported in this paper used the same settings.

Table 2: Model Configuration Parameters

Parameter	Value
Batch size (B)	98
Maximum sequence length (L)	96
Vocabulary size (V)	1538
Number of attention heads (H_{attn})	6
TopK for ICD attention (κ)	16
Hidden size (D)	256
Hidden dropout probability (p_{drop})	0.1
Layer normalization epsilon (ϵ_{LN})	1×10^{-12}
Number of random features ($R_{\text{FAVOR+}}$)	64
Intermediate size (D_{FFN})	1024
Number of Transformer layers (L_{hidden})	6
Attention dropout probability (p_{attn})	0.1

Sequence length and TopK attention. We fix the maximum visit sequence length at $L = 96$ as a pragmatic trade-off between coverage and computational cost. In the curated CPRD, most patient trajectories contain only a small number of visits (median = 4, minimum = 2), so $L = 96$ provides substantial headroom beyond typical histories. The choice of L was reviewed with clinical collaborators to ensure that the retained context is plausible for longitudinal risk modelling while remaining tractable for routine evaluation.

For the small long tail of patients with histories longer than L, we avoid systematically privileging either the earliest or the most recent portion of the record by sampling a contiguous window of length L uniformly at random from the available visits and preserving the chronological order within that window. This yields a simple, reproducible rule that limits computational load while reducing bias that could arise from always truncating at the head or tail of the sequence.

We additionally use ICD-level TopK attention with $k = 16$ to control within-visit sparsity. Visits can contain many correlated codes, and restricting attention to the k most salient code embeddings reduces noise from dense co-documentation while keeping computation bounded. In practice, $k = 16$ was selected to capture multi-problem visits without collapsing to an overly diffuse representation.

Appendix F. Hyperparameter Tuning of the Dependency-Aware Loss Function

To evaluate our dependency-aware loss under extreme label imbalance, we performed a two-stage sweep (Table 3) over its two key hyperparameters: the negative-weight multiplier α_{neg}, which controls the relative emphasis on negatives via $w_{\text{neg}} = \alpha_{\text{neg}} \frac{P}{N+\epsilon}$ and therefore trades precision against recall by penalising false positives, and the interaction regularisation strength λ_{reg}, which scales the interaction penalty. We first sweep α_{neg} at fixed $\lambda_{\text{reg}} = 0.10$, then sweep λ_{reg} at fixed $\alpha_{\text{neg}} = 25$. Each setting was assessed via three-fold cross-validation on held-out visits, reporting mean $\pm$ SD for recall, precision, $F_{\beta=2}$, balanced accuracy, and average precision.

Effect of α_{neg}. Holding $\lambda_{\text{reg}} = 0.10$ fixed, increasing α_{neg} produces the expected precision–recall trade-off. Small values ($\alpha_{\text{neg}} \in \{1, 2\}$) under-penalise false positives, yielding very high recall (≈ 0.90) but very low precision (≈ 0.02), with correspondingly low $F_{\beta=2}$ (≈ 0.09–0.11). As α_{neg} increases, precision rises while recall falls; at very large α_{neg} (e.g. 200), recall drops to ≈ 0.08 while precision increases substantially (≈ 0.55), again reducing $F_{\beta=2}$. The best compromise in this sweep is achieved at $\alpha_{\text{neg}} = 25$, with recall ≈ 0.53, precision ≈ 0.12, $F_{\beta=2} \approx 0.31$, balanced accuracy ≈ 0.76 (present/absent), and average precision ≈ 0.16.

Effect of λ_{reg}. Fixing $\alpha_{\text{neg}} = 25$, we sweep λ_{reg} and observe that performance is relatively stable in the range shown in Table 3, with the strongest overall trade-off at $\lambda_{\text{reg}} = 10$. Larger values yield diminishing returns and small declines in $F_{\beta=2}$ and average precision, consistent with mild over-regularisation of the interaction term. Based on these results, we select $\alpha_{\text{neg}} = 25$ and $\lambda_{\text{reg}} = 10$ for all subsequent experiments, as this setting consistently balances sensitivity and precision while capturing meaningful ICD-10 code dependencies.

Appendix G. Definition and Interpretation of Evaluation Metrics

G.1. Test-time evaluation protocol

At test time, the model outputs a probability vector over all V=1538 ICD-10 codes for each target visit. We evaluate performance on a subset of target visits that contain at least one *incident* code, defined as a code that appears for the *first* time in that patient's history at that visit (the "new" codes). Let $\mathcal{T}$ denote this set of target visits.

We report two stratified evaluations on the same $\mathcal{T}$. For a target visit $(b, \ell) \in \mathcal{T}$, let $\mathcal{N}^{b,\ell}$ be the set of new (first-occurrence) codes at that visit, and let $\mathcal{O}^{b,\ell}$ be the set of codes that are present at the visit but have appeared previously in the patient's history ("old" codes). To avoid conflating the two tasks, metrics are computed on different subsets of label entries:

$$\mathcal{S}_{\text{new}} = \Big\{(b, \ell, v) : (b, \ell) \in \mathcal{T},\ v \in \mathcal{N}^{b,\ell} \text{ or } y^{b,\ell,v} = 0\Big\},$$
$$\mathcal{S}_{\text{old}} = \Big\{(b, \ell, v) : (b, \ell) \in \mathcal{T},\ v \in \mathcal{O}^{b,\ell} \text{ or } y^{b,\ell,v} = 0\Big\}.$$

Thus, rows labelled "new" in our tables report metrics computed on $\mathcal{S}_{\text{new}}$ (incident positives versus absent negatives), while rows labelled "old" report metrics on $\mathcal{S}_{\text{old}}$ (previously-seen positives versus absent negatives). In both cases, each triplet (b, ℓ, v) is treated as a binary decision, and all metrics below are computed by pooling outcomes over the corresponding evaluated set $\mathcal{S} \in \{\mathcal{S}_{\text{new}}, \mathcal{S}_{\text{old}}\}$.

In our dataset, a typical visit contains between 2 (first quartile) and 8 (third quartile) true ICD-10 codes out of V=1538 possible labels, corresponding to a positive rate over label entries of approximately 0.13%–0.52%. If a random predictor selects k codes uniformly and the visit contains M true codes, then

$$\mathbb{E}[\text{TP}] = \frac{kM}{V}, \qquad \mathbb{E}[\text{Precision}] = \frac{M}{V}, \qquad \mathbb{E}[\text{Recall}] = \frac{k}{V}.$$

For $M = 2$ to $M = 8$, the expected random precision is therefore approximately 0.0013–0.0052. The expected balanced accuracy is close to 0.50, since $\text{TPR} = k/V$ and $\text{TNR} = 1 - k/V$. This illustrates the challenge posed by extreme label imbalance and contextualises precision values in the 10%–15% range as substantial enrichment over random selection.

Loss	α_{neg}	λ_{reg}	Recall	Precision	$F_{\beta=2}$	Bal. Acc. (present/absent)	Avg. Prec.	Bal. Acc. (new/old)
DA	200	0.10	0.080 ± 0.000	0.553 ± 0.017	0.096 ± 0.001	0.541 ± 0.001	0.160 ± 0.023	0.527 ± 0.007
DA	100	0.10	0.240 ± 0.023	0.203 ± 0.036	0.230 ± 0.011	0.598 ± 0.012	0.163 ± 0.017	0.573 ± 0.007
DA	50	0.10	0.377 ± 0.014	0.173 ± 0.017	0.305 ± 0.017	0.684 ± 0.006	0.157 ± 0.014	0.657 ± 0.017
DA	25	0.10	0.527 ± 0.040	0.120 ± 0.000	0.314 ± 0.011	0.755 ± 0.020	0.163 ± 0.007	0.737 ± 0.014
DA	10	0.10	0.717 ± 0.024	0.063 ± 0.007	0.234 ± 0.017	0.835 ± 0.009	0.167 ± 0.007	0.810 ± 0.011
DA	5	0.10	0.817 ± 0.028	0.040 ± 0.000	0.167 ± 0.001	0.867 ± 0.012	0.167 ± 0.007	0.847 ± 0.007
DA	2	0.10	0.903 ± 0.024	0.023 ± 0.007	0.105 ± 0.026	0.872 ± 0.008	0.167 ± 0.007	0.863 ± 0.007
DA	1	0.10	0.923 ± 0.024	0.020 ± 0.000	0.092 ± 0.000	0.863 ± 0.008	0.163 ± 0.007	0.857 ± 0.007
DA	**25**	**10.00**	$\mathbf{0.517 \pm 0.024}$	$\mathbf{0.123 \pm 0.007}$	$\mathbf{0.315 \pm 0.015}$	$\mathbf{0.750 \pm 0.013}$	$\mathbf{0.167 \pm 0.013}$	$\mathbf{0.730 \pm 0.011}$
DA	25	25.00	0.520 ± 0.011	0.117 ± 0.007	0.306 ± 0.006	0.752 ± 0.004	0.163 ± 0.007	0.730 ± 0.011
DA	25	50.00	0.543 ± 0.035	0.103 ± 0.017	0.271 ± 0.055	0.761 ± 0.014	0.160 ± 0.011	0.743 ± 0.024
DA	25	75.00	0.540 ± 0.030	0.100 ± 0.011	0.286 ± 0.012	0.760 ± 0.012	0.157 ± 0.007	0.733 ± 0.024

Table 3: Hyperparameter sweep results for the dependency-aware loss using three-fold cross-validation. The selected combination, $\alpha_{\text{neg}} = 25$ and $\lambda_{\text{reg}} = 10$, is highlighted in bold.

G.2. Precision and recall

For a single label, let TP be true positives, FP false positives, and FN false negatives (Pedregosa et al., 2011). Precision and recall are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Precision is the fraction of predicted positives that are correct and decreases with additional false positives. Recall is the fraction of actual positives that are detected and decreases with additional false negatives. In our reports, we threshold probabilities at a fixed decision threshold τ selected on the validation split (Section 4.2) and micro-average counts by pooling TP, FP, and FN over all $(b, \ell, v) \in \mathcal{S}$.

G.3. F_β score

To combine precision and recall into a single measure (Pedregosa et al., 2011), we use the F_β score:

$$F_\beta = (1 + \beta^2) \frac{\text{Precision} \times \text{Recall}}{\beta^2 \,\text{Precision} + \text{Recall}}.$$

Here $\beta > 1$ places more weight on recall, while $\beta < 1$ places more weight on precision; F_1 ($\beta = 1$) balances both equally, and F_2 ($\beta = 2$) emphasizes recall twice as much as precision. As with precision/recall, we report micro-averaged F_β over $\mathcal{S}$.

G.4. Balanced accuracy

Balanced accuracy is defined as

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \qquad \text{Balanced accuracy} = \frac{1}{2}\big(\text{Recall} + \text{Specificity}\big),$$

where TN denotes true negatives (Brodersen et al., 2010). By averaging the true positive rate (recall) and true negative rate (specificity), balanced accuracy gives equal importance to both classes and avoids inflated scores when one class is much larger. In multi-label evaluation, we treat each $(b, \ell, v) \in \mathcal{S}$ as a binary decision and micro-average TN and FP together with TP and FN.

G.5. Average precision (precision–recall AUC)

Average precision (AP) summarizes the precision–recall curve as

$$\text{AP} = \sum_{n=1}^{N} \big(R_n - R_{n-1}\big)\, P_n, \quad R_0 = 0,$$

where P_n and R_n are precision and recall at threshold n (Saito and Rehmsmeier, 2015). AP is the area under the precision–recall curve and provides a threshold-independent measure of performance on imbalanced data. In the multi-label setting we compute micro-AP by pooling scores and labels over all $(b, \ell, v) \in \mathcal{S}$ (macro-AP can be obtained by computing per-code AP and averaging across codes).

Together, these metrics provide complementary views of model performance in imbalanced, multi-label set-

tings. Precision and recall expose the trade-off between false positives and false negatives. F_β condenses that trade-off according to the chosen emphasis. Balanced accuracy ensures that both positive and negative classes influence the score equally, and average precision captures overall ranking quality without committing to a single threshold.

G.6. Permutation test for statistical significance

We compare two models' performance using a paired permutation test over 3-fold cross-validation repeated with 3 random seeds, yielding $N = 9$ matched score pairs $\{(s_{1,k}, s_{2,k})\}_{k=1}^{9}$. Let

$$\Delta_{\text{obs}} = \frac{1}{N}\sum_{k=1}^{N}(s_{1,k} - s_{2,k})$$

be the observed mean difference. Under the null hypothesis that the two models perform equally, each paired difference can be sign-flipped (equivalently, the two scores in a pair can be swapped) without changing its distribution. We draw $M = 10{,}000$ random sign-flip patterns, compute

$$\Delta_i = \frac{1}{N}\sum_{k=1}^{N}(s_{1,k}^{(i)} - s_{2,k}^{(i)}),$$

and estimate the two-sided p-value as

$$p \;=\; \frac{1 + \sum_{i=1}^{M} \mathbf{1}\big(|\Delta_i| \geq |\Delta_{\text{obs}}|\big)}{M+1}.$$

Choosing $M = 10{,}000$ gives a Monte Carlo standard error of order $\sqrt{p(1-p)/(M+1)} \lesssim 0.005$, ensuring stable p-value estimates at minimal cost. This non-parametric procedure makes no distributional assumptions and tests whether the observed difference could arise by chance (Noguchi et al., 2021; DiCiccio and Efron, 1996).

Appendix H. Masking Probability Ablation

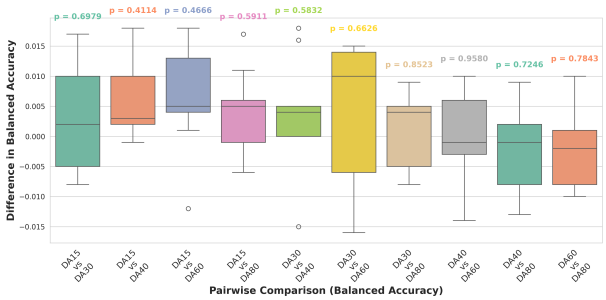

Figure 6: Pairwise Differences in Balanced Accuracy among Different Masking Probabilities

In light of ongoing interest in optimising masked objectives, we evaluate the sensitivity of our visit-level masking strategy to the masking rate. The conventional 15% masking rate in token-level MLM (Devlin et al., 2019) is often adopted by default. Inspired by systematic studies of masking rates (Wettig et al., 2022), we test whether alternative masking rates improve performance in our setting.

Visit-level masking probability. In our framework, the masking probability refers to the probability p_{visit} of selecting a visit as a training target (Section 3.3), not the probability of masking individual tokens. A visit is eligible if it contains at least one ICD–10 code that has not appeared earlier in the patient's history. If a trajectory contains no eligible visit, no MLM masking is applied in that iteration (so it contributes no MLM loss). For each selected visit, we corrupt its active codes using a BERT-style scheme. 80% are replaced by `[MASK]`, 10% by random codes, and 10% remain unchanged.

Ablation protocol. We vary the visit masking probability across five levels: $p_{\text{visit}} \in \{0.15, 0.30, 0.40, 0.60, 0.80\}$. All experiments use the same model architecture, optimiser settings, and data splits. Only p_{visit} changes. The 15% row in Table 4 is an independent rerun under the same nominal hyperparameter setting, not a reuse of the final checkpoint reported in the main results. Evaluation is performed via three-fold cross-validation with three random seeds per fold. Results are summarised in Table 4 and visualised in Figure 6. Across the five masking rates,

Metric	15% masking	30% masking	40% masking	60% masking	80% masking
Balanced Acc. (Pres./Abs.)	0.756 (0.742–0.770)	0.753 (0.743–0.763)	0.749 (0.737–0.761)	0.750 (0.736–0.764)	0.752 (0.741–0.763)
Balanced Acc. (new-code detection)	0.739 (0.724–0.753)	0.734 (0.725–0.744)	0.729 (0.716–0.742)	0.729 (0.714–0.744)	0.731 (0.717–0.745)
PRC–AUC	0.143 (0.131–0.155)	0.143 (0.129–0.158)	0.143 (0.128–0.158)	0.139 (0.124–0.154)	0.141 (0.126–0.156)
$F_{\beta=2}$	0.283 (0.266–0.300)	0.289 (0.276–0.302)	0.283 (0.269–0.297)	0.283 (0.267–0.300)	0.285 (0.271–0.299)
Recall	0.533 (0.504–0.563)	0.527 (0.507–0.547)	0.518 (0.492–0.543)	0.521 (0.492–0.550)	0.526 (0.502–0.549)
Precision	0.099 (0.090–0.108)	0.102 (0.096–0.109)	0.100 (0.092–0.108)	0.099 (0.091–0.107)	0.100 (0.092–0.108)

Table 4: Masking probability ablation results with 95% confidence intervals. Results are from three-fold cross-validation with three seeds under $\alpha_{\text{neg}} = 25$ and $\lambda_{\text{reg}} = 10$.

downstream metrics differ by less than one percentage point, with overlapping 95% confidence intervals and no pairwise comparison reaching statistical significance ($p > 0.05$).

Compute cost. Training time increases substantially at higher masking rates: relative to the 15% baseline (6 min 09 s), the 30% and 40% settings require approximately 9% (6 min 41 s) and 8% (6 min 37 s) more compute, while 60% and 80% masking incur roughly 41% (8 min 40 s) and 37% (8 min 24 s) longer runtimes. Since higher masking rates yield no measurable performance gains but increase training cost, we retain $p_{\text{visit}} = 0.15$ for all main experiments.

Appendix I. Dependency-Aware Loss

Our training objective operates on masked target visits (Section 3.3). For each mini-batch, let $\mathcal{T}$ denote the set of masked target visits (b, ℓ) for which we predict the code vector. For each target $(b, \ell) \in \mathcal{T}$ and code $v \in \{1, \dots, V\}$, the model outputs logits $\hat{y}^{b,\ell,v} \in \mathbb{R}$ and probabilities $\hat{p}^{b,\ell,v} = \sigma(\hat{y}^{b,\ell,v})$ with targets $y^{b,\ell,v} \in \{0, 1\}$ on valid (non-padding) entries. We write $\mathcal{V}^{b,\ell} = \{v : y^{b,\ell,v} \in \{0, 1\}\}$ for the valid codes in that target visit and $|\mathcal{V}^{b,\ell}|$ for its size, and define the valid entry set

$$\mathcal{S} = \{(b, \ell, v) : (b, \ell) \in \mathcal{T},\ v \in \mathcal{V}^{b,\ell}\}.$$

Imbalance-aware correctness. We use a bounded, differentiable per-code correctness score

$$c^{b,\ell,v} = 1 - \left|\hat{p}^{b,\ell,v} - y^{b,\ell,v}\right| = \begin{cases} \hat{p}^{b,\ell,v}, & y^{b,\ell,v} = 1, \\ 1 - \hat{p}^{b,\ell,v}, & y^{b,\ell,v} = 0\ , \end{cases}$$

and re-scale it on each mini-batch to counter extreme sparsity. Let

$$P = \sum_{(b,\ell,v)\in\mathcal{S}} y^{b,\ell,v}, \qquad N = \sum_{(b,\ell,v)\in\mathcal{S}} \big(1 - y^{b,\ell,v}\big),$$

where $\mathcal{S}$ denotes valid target entries in the batch. We define

$$w_{\text{pos}} = \frac{N}{P + \epsilon}, \qquad w_{\text{neg}} = \alpha_{\text{neg}} \frac{P}{N + \epsilon},$$

where $\epsilon > 0$ ensures numerical stability and $\alpha_{\text{neg}} > 0$ tunes the relative weight on negatives (trading precision vs. recall). The class-weighted correctness is

$$\tilde{c}^{b,\ell,v} = \begin{cases} w_{\text{pos}}\, c^{b,\ell,v}, & y^{b,\ell,v} = 1, \\ w_{\text{neg}}\, c^{b,\ell,v}, & y^{b,\ell,v} = 0\ . \end{cases}$$

Rank-weighted (listwise) aggregation. Within a target visit (b, ℓ), we sort the valid values $\{\tilde{c}^{b,\ell,v}\}_{v\in\mathcal{V}^{b,\ell}}$ in descending order $\tilde{c}^{b,\ell}_{(1)} \geq \cdots \geq \tilde{c}^{b,\ell}_{(|\mathcal{V}^{b,\ell}|)}$ and compute an ordered weighted average that emphasises the lowest class-weighted correctness entries:

$$\mathcal{C}^{b,\ell} = \sum_{i=1}^{|\mathcal{V}^{b,\ell}|} \frac{2i}{|\mathcal{V}^{b,\ell}|\big(|\mathcal{V}^{b,\ell}| + 1\big)}\, \tilde{c}^{b,\ell}_{(i)}.$$

This listwise weighting counters gradient dilution by assigning the largest coefficients to the smallest class-weighted correctness entries.

Label-space interaction. Element-wise losses optimise each label independently. To regularise predictions toward coherent comorbidity structure, we add a learned quadratic interaction on the unsorted vector $\tilde{\mathbf{c}}^{\,b,\ell} = (\tilde{c}^{b,\ell,v})_{v=1}^{V}$. For indices $v \notin \mathcal{V}^{b,\ell}$ (padding/ignored labels), we set $\tilde{c}^{b,\ell,v} = 0$, so $\tilde{\mathbf{c}}^{\,b,\ell} \in \mathbb{R}^V$ is well-defined and the quadratic form effectively restricts to valid labels:

$$\begin{aligned} \text{raw}^{b,\ell} &= \frac{1}{|\mathcal{V}^{b,\ell}|}(\tilde{\mathbf{c}}^{\,b,\ell})^\top \mathbf{W} \tilde{\mathbf{c}}^{\,b,\ell}, \\ s^{b,\ell} &= \sigma\big(\text{raw}^{b,\ell}\big), \\ \mathcal{I}^{b,\ell} &= 1 - s^{b,\ell}. \end{aligned}$$

The TopK attention layer models interactions in representation space. It produces a permutation-invariant summary of within-visit code embeddings and helps the encoder form clinically meaningful latent states. In contrast, standard multi-label objectives (BCE, focal, class-balanced) remain element-wise in the output space and do not provide explicit gradient coupling between labels. The interaction term $(\tilde{\mathbf{c}}^{b,\ell})^\top \mathbf{W}\, \tilde{\mathbf{c}}^{b,\ell}$ thus plays a complementary role. It regularises the predicted label distribution toward population-level comorbidity structure, discouraging isolated, weakly supported positive predictions that drive precision collapse under extreme imbalance, while yielding an inspectable matrix $\mathbf{W}$ for global auditing. We parameterise $\mathbf{W}$ via a free matrix and enforce symmetry and a zero diagonal at each forward pass, ensuring $\mathbf{W} = \mathbf{W}^\top$ and $\mathrm{diag}(\mathbf{W}) = \mathbf{0}$ throughout training; $\mathbf{W}$ is regularised via decoupled weight decay (AdamW).

Final objective. For each masked target visit $(b, \ell) \in \mathcal{T}$, the dependency-aware loss is

$$\mathcal{L}^{b,\ell} = 1 - \mathcal{C}^{b,\ell} + \lambda_{\text{reg}}\, \mathcal{I}^{b,\ell}. \tag{8}$$

The mini-batch loss is computed only when $|\mathcal{T}| > 0$. Mini-batches with no selected target visits are skipped. Since eligible target visits contain at least one positive code by construction, selected target batches satisfy $P > 0$. The mini-batch objective is then

$$\mathcal{L}_{\text{batch}} = \frac{1}{|\mathcal{T}|} \sum_{(b,\ell)\in\mathcal{T}} \mathcal{L}^{b,\ell}.$$

Appendix J. Baseline Losses: Mathematical and Implementation Details

This appendix gives the exact baseline objectives used in Section 3.5. All baselines are trained on the same valid supervised entries and differ only in how they weight or modulate the binary classification loss.

Notation and masking. For a mini-batch, let $s = (b, \ell, v)$ index one supervised entry, corresponding to patient b, target visit ℓ, and ICD-10 code v. We write z_s for the model logit, $y_s \in \{0, 1\}$ for the binary label, and $v(s)$ for the code associated with entry s. Padded labels and ignored targets are excluded from all weight estimation and loss averaging. The valid supervised set is

$$\mathcal{S} = \{s : y_s \in \{0, 1\}\}.$$

All losses below are averaged over $\mathcal{S}$. For each code v, let $\mathcal{S}_v$ denote the valid entries associated with that code:

$$\mathcal{S}_v = \{s \in \mathcal{S} : v(s) = v\}.$$

We then define

$$n_v = \sum_{s\in\mathcal{S}_v} y_s, \qquad m_v = |\mathcal{S}_v|.$$

Here n_v is the mini-batch positive count for code v, and $m_v - n_v$ is the corresponding negative count. When $n_v = 0$, we use $\max(n_v, 1)$ in denominators to avoid division by zero. This is only a numerical safeguard, since the positive-weighted term is inactive when no positive examples of code v appear in the mini-batch.

We use the numerically stable binary cross-entropy with logits,

$$\ell_{\text{BCE}}(z, y) = (1 - y)\,\mathrm{softplus}(z) + y\,\mathrm{softplus}(-z),$$

where $\mathrm{softplus}(t) = \log(1 + \exp(t))$.

Weighted binary cross-entropy. Weighted BCE upweights positive examples for codes with low positive support. For each code v, we compute the mini-batch positive weight as the negative-to-positive ratio,

$$w_v = \frac{m_v - n_v}{\max(n_v, 1)}.$$

The weighted BCE objective is

$$\mathcal{L}_{\text{wBCE}} = \frac{1}{|\mathcal{S}|} \sum_{s\in\mathcal{S}} \ell_{\text{wBCE}}(z_s, y_s),$$

where

$$\ell_{\text{wBCE}}(z_s, y_s) = (1-y_s)\,\mathrm{softplus}(z_s) + y_s w_{v(s)}\,\mathrm{softplus}(-z_s).$$

Thus, only the positive term is reweighted, matching the usual positive-weight convention for binary cross-entropy with logits.

Class-balanced loss. We use a mini-batch version of class-balanced weighting based on the effective number of positive examples (Cui et al., 2019). Let $\beta \in [0, 1)$ control the strength of the effective-number correction. For each code v, the raw code-level weight is

$$\alpha_v^{\text{raw}} = \frac{1 - \beta}{1 - \beta^{\max(n_v,1)}}.$$

We normalise these weights so that their average across codes is one:

$$Z = \frac{1}{V}\sum_{j=1}^{V} \alpha_j^{\text{raw}}, \qquad \alpha_v = \frac{\alpha_v^{\text{raw}}}{Z}.$$

In our implementation, α_v is applied to all valid entries associated with code v, including both positive and negative labels. The class-balanced BCE objective is

$$\mathcal{L}_{\text{CB}} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \alpha_{v(s)}\ell_{\text{BCE}}(z_s, y_s).$$

The normalisation by Z helps keep the overall loss scale comparable across mini-batches with different label compositions.

Focal loss. Focal loss downweights easy examples and emphasises entries that are currently misclassified or uncertain (Lin et al., 2017). Let

$$\hat{p}_s = \sigma(z_s), \qquad p_{t,s} = y_s\hat{p}_s + (1-y_s)(1-\hat{p}_s),$$

where $p_{t,s}$ is the probability assigned to the true class. The focal modulation term is $(1-p_{t,s})^\gamma$, with focusing parameter $\gamma \geq 0$.

As a bounded code-specific class weight, we map the class-balanced weight to $(0, 1)$:

$$\tilde{\alpha}_v = \frac{\alpha_v}{1+\alpha_v}.$$

The entry-specific focal weight is

$$\alpha_{t,s} = \tilde{\alpha}_{v(s)}y_s + \big(1-\tilde{\alpha}_{v(s)}\big)(1-y_s).$$

The focal objective is

$$\mathcal{L}_{\text{F}} = -\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} \alpha_{t,s}(1-p_{t,s})^\gamma \log(p_{t,s}+\epsilon),$$

where $\epsilon > 0$ is a small numerical-stability constant.

Implementation summary. All three baselines use the same valid-entry mask $\mathcal{S}$. Padded labels and ignored targets are removed before computing mini-batch counts, weights, and losses. The weights are estimated from the current mini-batch rather than from global corpus frequencies, matching the training protocol used for the dependency-aware objective. The final loss is always averaged over valid supervised entries only.

Appendix K. Ablation of Dependency-Aware Loss Components

To quantify the individual impact of the two terms, interaction penalty and rank–based correctness weighting, on overall model performance, we conducted an ablation study in which each component was deactivated in isolation (Table 5). All other elements of the training pipeline were held constant to serve as a baseline. In one variant, the interaction penalty was removed, while the rank–based weighting remained active. In the other, the rank–based aggregation was replaced by unweighted correctness scores, with the interaction term preserved.

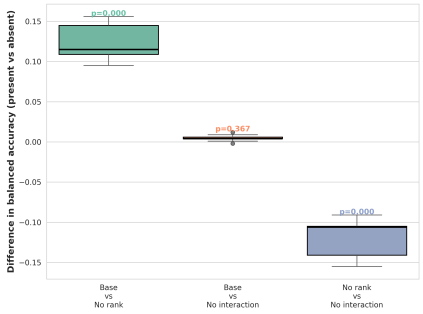

Figure 7: Pairwise Differences in Balanced Accuracy for the Ablation Study of Components in the Dependency-Aware Loss Function

Appendix L. TCN Backbone Compatibility Check

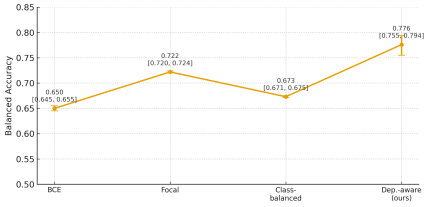

Figure 8: Balanced Accuracy (95% CI) of a TCN under Four Loss Functions.

To test whether the dependency-aware loss is tied only to the Transformer backbone, we integrated it into a Temporal Convolutional Network (TCN). Unlike the Transformer, which uses global self-attention, the

Table 5: Summary of Evaluation Metrics for Dependency Aware Loss Variants (95% CI)

Metric	Dependency-Aware	No rank	No interaction
Balanced Accuracy (present vs absent)	0.754 (0.747–0.762)	0.632 (0.614–0.650)	0.750 (0.740–0.759)
Balanced Accuracy (new-code detection)	0.738 (0.729–0.746)	0.623 (0.605–0.642)	0.736 (0.725–0.746)
Recall (Class 1)	0.526 (0.511–0.540)	0.514 (0.474–0.555)	0.516 (0.498–0.533)
Precision	0.119 (0.108–0.130)	0.010 (0.009–0.010)	0.126 (0.116–0.135)
$F_{\beta=2}$	0.311 (0.293–0.330)	0.046 (0.046–0.047)	0.318 (0.301–0.335)
PRC-AUC	0.163 (0.149–0.177)	0.067 (0.052–0.081)	0.162 (0.149–0.175)
Accuracy	0.981 (0.980–0.982)	0.751 (0.745–0.757)	0.982 (0.982–0.983)

TCN relies on stacked causal convolutions and local receptive fields to model temporal dynamics. This experiment is intended as a secondary architectural compatibility check.

We replicated the training protocol and reused the same loss hyperparameter settings as in the Transformer experiments. The TCN was evaluated on the held-out test split using balanced accuracy (at the fixed decision rule used throughout the paper) and area under the precision–recall curve (PRC AUC). The TCN trained with the dependency-aware loss achieved a balanced accuracy of 0.776 (95% CI: 0.755–0.794) and a PRC AUC of 0.11. In comparison, binary cross-entropy (BCE) yielded 0.650 (95% CI: 0.645–0.655) and 0.19; focal loss yielded 0.722 (95% CI: 0.720–0.724) and 0.04; and class-balanced loss yielded 0.673 (95% CI: 0.671–0.675) and 0.05.

These results highlight a trade-off between global ranking quality and thresholded discrimination under extreme imbalance. BCE attains the highest PRC AUC (0.19), consistent with strong overall ranking, but yields substantially lower balanced accuracy at the operating point used for evaluation (0.650). Conversely, focal and class-balanced objectives improve balanced accuracy relative to BCE, but their PRC AUC collapses to 0.04–0.05, indicating degraded ranking structure. The dependency-aware loss achieves the highest balanced accuracy (0.776) while maintaining a markedly higher PRC AUC (0.11) than other tail-sensitive losses, demonstrating that it improves decision-boundary behaviour without the severe loss of ranking quality observed in focal and class-balanced objectives.

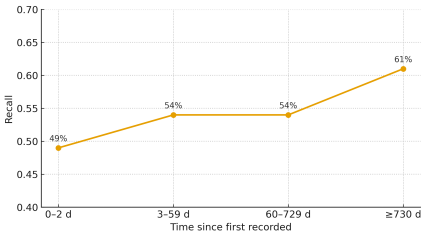

Figure 9: Recall Stratified by Time since a Code was First Recorded in the Patient History ("code age").

Appendix M. Impact of Diagnosis Chronicity on Recall

To probe how recall varies between newly recorded codes and established chronic conditions, we analyze recall as a function of code age (the time elapsed since a specific ICD-10 code was first recorded for a patient). We stratify positive test labels into four bands based on their first-recorded date: incident/very recent (0–2 days), recent (3–59 days), established (60–729 days), and long-term ($\geq$ 730 days). To ensure a fair comparison across bands, we fix a single global decision threshold tuned on the validation set to enforce high specificity (minimizing false positives among absent codes) and apply it uniformly across all strata.

Figure 9 shows a monotonic trend. Recall is approximately 49% for incident/very recent codes and increases with code age, peaking at $\approx$ 61% for codes present for at least two years. This pattern is consistent with the intuition that long-standing conditions are more predictable from prior history, while newly recorded codes are harder and must be inferred from comorbidity context and longitudinal signals.

Appendix N. Auditing the Learned Label-Space Dependencies

Table 6: Examples of high-weight positive learned dependencies from the 3.2M-patient checkpoint.

Code pair	Interpretation	Weight
E11.9–I10	Type 2 diabetes–hypertension	0.0838
F32.9–F41.9	Depression–anxiety	0.0828
J44.0–J44.9	COPD subcodes	0.0698
D50.9–D64.9	Anaemia codes	0.0678
I20.9–I25.1	Ischaemic heart disease	0.0674
H25.1–H26.9	Cataract codes	0.0615
I48–I48.9	Atrial fibrillation codes	0.0584
E10.9–H36.0	Diabetes–diabetic retinopathy	0.0415

Note. Weights are entries of the learned output-space matrix $\mathbf{W}$ from the same checkpoint used in Figure 10. They indicate predictive compatibility under the training objective. They are not causal effects and should not be interpreted as direct empirical co-occurrence frequencies.

Additional observations from the clustered dependency map. Beyond the disease communities discussed in the main text, the clustered heatmap reveals several useful audit signals (Figrue 10). First, some local blocks follow ICD coding structure, with closely related subcodes clustering together, suggesting that $\mathbf{W}$ captures both clinical co-occurrence and coding-level redundancy. Second, several malignancy-related codes form broader positive regions, consistent with shared documentation patterns and multimorbidity around cancer diagnoses. Third, the pronounced horizontal and vertical bands indicate hub-like diagnoses, especially hypertension (I10), whose broad connectivity matches its high prevalence and widespread co-occurrence in the raw EHR data. Finally, the presence of weaker blue regions shows that the interaction term does not only encourage co-prediction. It can also downweight label pairs that are less compatible under the observed data distribution. These patterns make the learned dependency matrix useful as an audit object, because it exposes both expected clinical structure and possible coding-driven associations.

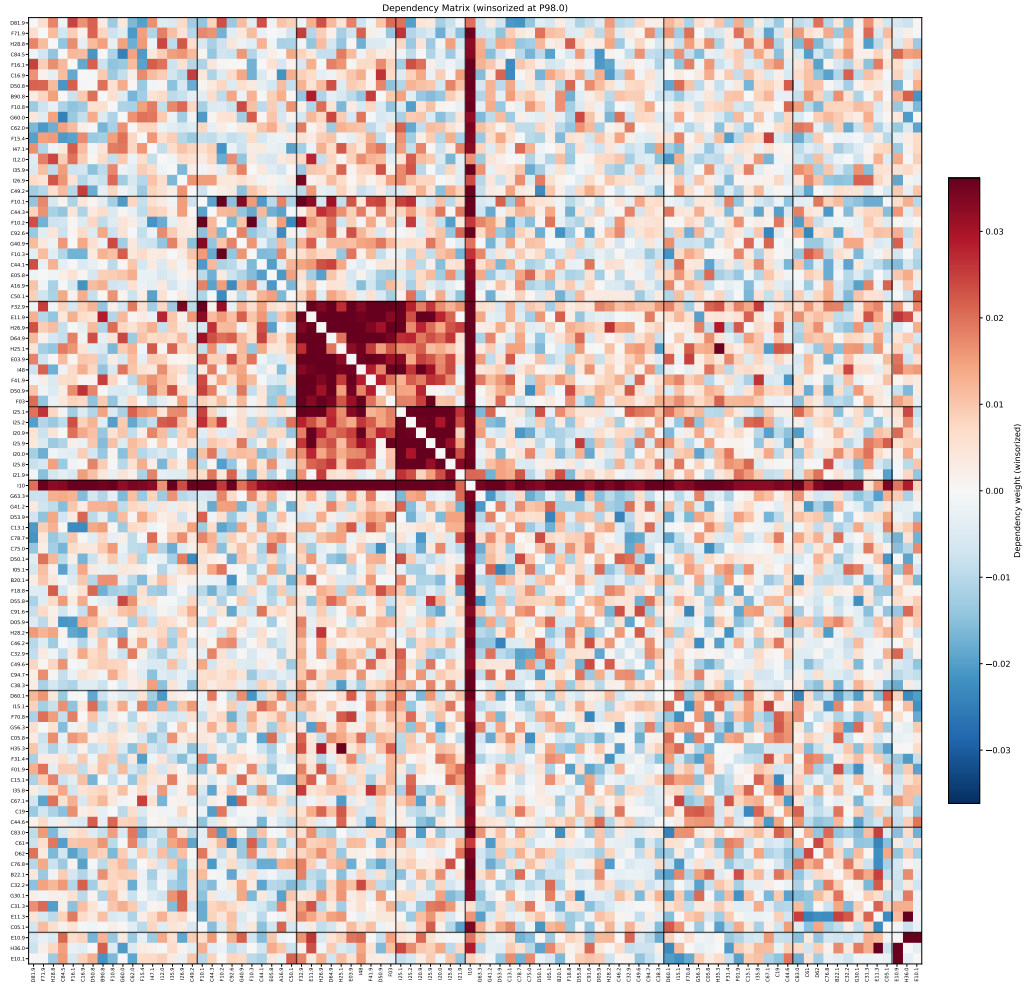

Figure 10: Clustered heatmap of the learned output-space dependency matrix $\mathbf{W}$ from the final 3.2M-patient checkpoint.

Note. Rows and columns correspond to ICD-10 labels ordered by spectral clustering, with grid lines marking the resulting communities. Colours show winsorised dependency weights: positive values indicate label pairs whose co-prediction is encouraged by the dependency objective, whereas negative values indicate discouraged co-prediction. The visible block structure suggests that $\mathbf{W}$ captures coherent diagnosis communities rather than isolated pairwise noise.