Leakage, Small Data and Strong Baselines: An Open, Reproducible Re-examination of Word2Vec–LSTM ICD Coding

Kannan Sekar Annu Radha
Department of Physics
King's College London
London, United Kingdom

Abstract

Automated ICD coding assigns diagnosis codes to clinical narratives. We re-examine an earlier Word2Vec–LSTM ICD-9 coding prototype (NLPK) and replace its unverifiable claims with an open, executable study. First, re-running the prototype shows that its input vector is computed from the *title of the target code itself*. A one-timestep LSTM trained on these vectors reaches 0.518 accuracy on the original 80/20 row split, in which 94 patients appear on both sides. A training-free nearest-title lookup reaches 0.770, exactly the fraction of test rows whose code occurs in training. A label-permutation control falls to 0.007. The prototype therefore measures dictionary lookup, not coding. Second, because MIMIC notes require credentialed access, we benchmark on openly downloadable data: the English CodiEsp-D corpus of 1,000 ICD-10-CM-coded clinical case reports, plus Symptom2Disease as a secondary short-text task. Under a single, validation-only protocol for model selection and thresholds, fine-tuning a pretrained clinical encoder in the PLM-ICD style is by far the strongest approach. Over five seeds, Bio_ClinicalBERT reaches micro-F1 0.738$\pm$0.012 on CodiEsp-50 and 0.686$\pm$0.008 on the top-50 ICD-10 categories, and the 14M-parameter TinyClinicalBERT reaches 0.712$\pm$0.012 and 0.627$\pm$0.015. The best linear model gets 0.422 on CodiEsp-50 (best macro-F1 0.359), and a wider TF-IDF grid only reaches 0.455. Split, overlap and shuffled-label controls find no leakage behind the gap. Among models trained without a fine-tuned pretrained encoder, TF-IDF linear models remain the best. A CAML-style label-attention CNN comes close (0.377 micro-F1, comparable macro-F1); its paired bootstrap gap to plain TF-IDF+LR has a 95% interval that includes zero. The faithful Word2Vec+BiLSTM reproduction of the original draft (0.308), BiLSTM+LAAT (0.187) and frozen TinyClinicalBERT features (0.342) are all significantly worse than TF-IDF+LR. Ablations show that, at this scale, positive-class re-weighting is essential: without it the BiLSTM collapses to the label prior. A bidirectional encoder helps, label attention helps CAML, and Word2Vec initialisation from ICD titles or in-domain text gives no reliable gain over random initialisation. All numbers in this paper come from the released notebook and fine-tuning scripts.

1 Introduction

Clinical coding maps free-text clinical documentation to standard classifications such as ICD-9-CM or ICD-10. Hospitals use these codes for reimbursement, epidemiology and audit. Neural coders such as CAML [Mullenbach et al., 2018], LAAT [Vu et al., 2020] and PLM-ICD [Huang et al., 2022] are usually trained on MIMIC-III discharge summaries [Johnson et al., 2016]. Recent replication work

Preprint.

shows that reported gains are sensitive to splits, thresholds and metric definitions [Edin et al., 2023]. Others argue that benchmarks restricted to frequent codes misrepresent real coding workflows [Gan et al., 2024].

This paper starts from an earlier student prototype (NLPK; Annu Radha [2025]). Its draft reported a macro-F1 of 0.837 for a Word2Vec [Mikolov et al., 2013] + bidirectional LSTM [Hochreiter and Schmidhuber, 1997, Schuster and Paliwal, 1997] coder on 500 ICD-9 codes. We could not find evidence for that number, or for the draft's deployment claims, in the code or data. We therefore treat the prototype as a hypothesis and test it properly. Our contributions are:

1. **A leakage diagnosis (§3).** We re-run the original pipeline on the public MIMIC-III demo tables [Johnson et al., 2019]. Its inputs are a deterministic function of the label, and the public demo contains no note text. A training-free lookup matches the model's ceiling, and a permutation control removes the signal. This is a concrete instance of the data-leakage failure mode catalogued by Kapoor and Narayanan [2023].
2. **An open benchmark protocol (§4).** We run patient-free, credential-free experiments on English CodiEsp-D [Miranda-Escalada et al., 2020]. There are three label spaces (top-50 codes, top-50 three-character categories, and all codes seen at least twice). Validation-only model selection and thresholding are applied, and we report micro/macro-F1, AUC, P@5, R@10, MAP and bootstrap intervals.
3. **A controlled comparison with ablations and error analysis (§5).** We compare linear, recurrent, convolutional label-attention and small pretrained models under the same data. Ablations cover the design choices of the original draft.
4. **Fine-tuned clinical encoders (§5).** PLM-ICD-style fine-tuning of TinyClinicalBERT and Bio_ClinicalBERT over five seeds on one consumer GPU, with leakage controls. It raises CodiEsp-50 micro-F1 from about 0.4 to above 0.7.

2 Related work

Neural ICD coding. Perotte et al. [2014] established hierarchical SVM baselines on MIMIC-II. CAML [Mullenbach et al., 2018] introduced per-label attention over convolutional features. LAAT [Vu et al., 2020] combined label attention with a BiLSTM encoder. PLM-ICD [Huang et al., 2022] adapted pretrained language models with segment pooling and reported 59.8% micro-F1 on the full MIMIC-III label set. Edin et al. [2023] re-implemented these models on MIMIC-III/IV and found that many earlier comparisons were confounded by thresholding and macro-F1 definitions. Gan et al. [2024] make eight recommendations for aligning benchmarks with coding workflows, and recent agentic systems target the same task [Edin et al., 2026]. Literature numbers are given only for context: they use MIMIC, different label spaces and different metrics, so they are not comparable with ours.

Clinical representations. Word2Vec [Mikolov et al., 2013] and CNN text classifiers [Kim, 2014] are standard in clinical NLP. Domain-pretrained transformers include BioBERT [Lee et al., 2020], ClinicalBERT [Alsentzer et al., 2019] and compact distilled variants such as TinyClinicalBERT [Rohanian et al., 2023]. Terminology resources such as the UMLS [Bodenreider, 2004] support synonym expansion. The original draft proposed this but never implemented it.

Open coded corpora. CodiEsp [Miranda-Escalada et al., 2020], from CLEF eHealth 2020, provides 1,000 Spanish clinical case reports with ICD-10 diagnosis and procedure codes, plus English machine translations; we use the Hugging Face mirror [Barcelona Supercomputing Center, Text Mining Unit, 2026]. Symptom2Disease [Gretel.ai, 2023] provides short patient-style symptom descriptions labelled with 22 diagnoses.

Leakage. Kapoor and Narayanan [2023] survey leakage-driven reproducibility failures in ML-based science. Our §3 is an instance of their "features that are proxies for the label" and "non-independence between train and test" categories.

Table 1: The original NLPK pipeline on the MIMIC-III demo diagnosis table. Inputs are averaged Word2Vec vectors of the *target* code's title. LSTM: mean±sd over 3 seeds. Lookup: training-free nearest title vector. Permuted: lookup after shuffling training labels (mean of 100).

Split	Train/test	Shared pts.	Code seen	Majority	Orig. LSTM	Lookup	Permuted
Row split 80/20 (original)	1372/344	94	0.770	0.023	0.518±0.003	0.770	0.007
Patient-disjoint split	1452/264	0	0.754	0.027	0.501±0.005	0.754	0.005

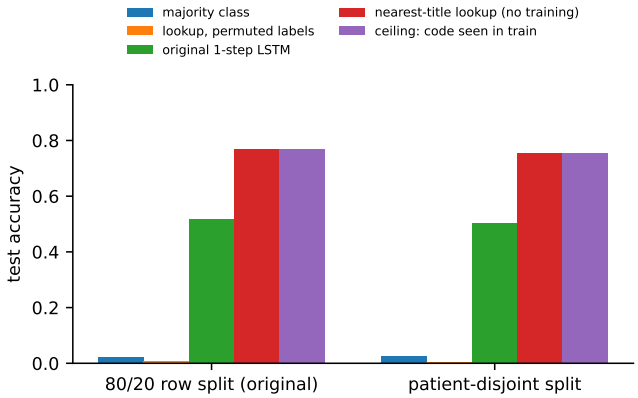


Figure 1: Leakage in the original pipeline. The training-free lookup bounds and exceeds the trained LSTM, and the permutation control collapses to chance.

3 Diagnosing the original pipeline

The attached `NLPGPT2024.py` needs a one-character fix to parse: a stray "≈" in an import. It trains skip-gram Word2Vec (dimension 100, window 7) on ICD-9 *code titles* and averages the vectors of each title. Every diagnosis row is then represented by *the averaged title vector of its own target code*. That vector is passed through a single-timestep LSTM(100) with a softmax over codes, trained with an 80/20 split over rows. We re-implemented it faithfully in PyTorch [Paszke et al., 2019] with gensim [Řehůřek and Sojka, 2010] on the MIMIC-III demo diagnosis table: 1,761 rows, 100 patients and 129 admissions. The public demo's `NOTEEVENTS` table is empty, so no clinical text exists in this setting.

Table 1 and Fig. 1 show the result. Three seeds of the original model give 0.518 accuracy on the row split. A training-free nearest-neighbour lookup over title vectors gives 0.770, which equals the fraction of test rows whose code appears in training: identical codes have identical inputs. Permuting training labels drops the lookup to 0.007. A patient-disjoint split changes little (0.501 / 0.754), because the leak runs through the code title rather than through the patient. The prototype's accuracy therefore reflects the fraction of codes seen in training and the optimisation quality of a lookup table. It says nothing about reading clinical text. A companion audit notebook supplied with the prototype reached the same conclusion. Its separate terminology-retrieval experiment found character TF-IDF retrieved the correct long title from an ICD-9 short title in 53.62% of 11,953 held-out queries, against 35.05% for word TF-IDF. We cite that result as context only; it is not a coding result.

4 Experimental setup

CodiEsp-D (English). We use the machine-translated English texts and the diagnosis annotations of the official train (500) and dev (250) sets; the public mirror does not ship test labels. The official train set is split 85/15 (seeded) into train (425) and validation (75). The official dev set (250) is our held-out **test** set and is never used for selection. Documents average 307 tokens and 11.3 distinct diagnosis codes. The label distribution is extremely long-tailed (Fig. 2): 19.4% of test code mentions are codes that never occur in training. We define three label spaces: **CodiEsp-50**, the 50 most frequent full codes (27.6% of test code mentions); **CodiEsp-Cat50**, the 50 most frequent 3-character ICD-10

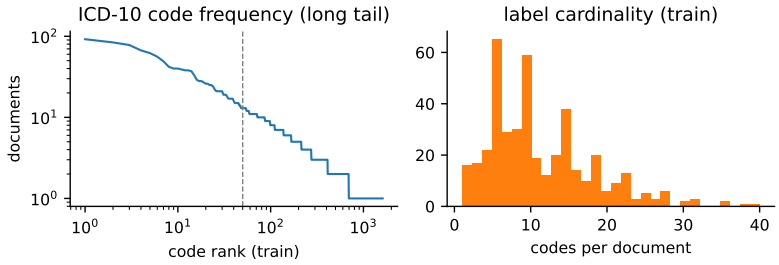


Figure 2: CodiEsp-D (train portion): code frequency is long-tailed (dashed line: top-50 cut-off) and documents carry about 11 codes each.

categories; and **CodiEsp-Full**, all 692 codes with at least two training occurrences (linear models only).

Symptom2Disease. This is the Gretel release [Gretel.ai, 2023] of the Kaggle Symptom2Disease data: 853 train / 212 test descriptions and 22 diseases, with 15% of train held out for validation. It is single-label and partly LLM-rewritten, so it serves as a sanity check of the model families rather than as evidence about coding.

Models. All neural models share one tokenizer and vocabulary. **Label-frequency prior**: scores equal to training label frequencies. **TF-IDF+LR / SVM**: 1–2-gram TF-IDF with one-vs-rest logistic regression or linear SVM [Pedregosa et al., 2011]. C and class weighting (none/balanced) are selected on validation micro-F1. **W2V+BiLSTM (draft reproduction)**: the architecture the draft described, a trainable embedding layer initialised from skip-gram Word2Vec (dimension 100, window 7). The Word2Vec model is trained on an in-domain corpus of MTSamples transcriptions [MTSamples.com, 2026] plus CodiEsp training texts (5,391 documents, 2.55M tokens; 100.0% vocabulary coverage). It is followed by a BiLSTM (128 per direction), max pooling and a sigmoid layer. **CAML** [Mullenbach et al., 2018]: convolution (kernel 10, 256 filters) with per-label attention. **BiLSTM+LAAT** [Vu et al., 2020]: BiLSTM with structured label attention. **Frozen TinyClinicalBERT+LR**: mean-pooled last-layer features of `nlpie/tiny-clinicalbert` [Rohanian et al., 2023, Wolf et al., 2020], a 4-layer model with hidden size 312 per its config. The features are taken over up to four 512-wordpiece segments, averaged, and fed to one-vs-rest LR. **Fine-tuned encoders (PLM-ICD style)** [Huang et al., 2022]: `nlpie/tiny-clinicalbert` (4 layers, 14M parameters) and `emilyalsentzer/Bio_ClinicalBERT` [Alsentzer et al., 2019] (BERT-base, 110M), fine-tuned end-to-end. Each document is split into 128-wordpiece segments (at most 24, i.e. about 3,000 wordpieces, so no training document is truncated). The segments are encoded independently, their token states are concatenated, and a label-attention head produces one logit per label. We train with AdamW (encoder learning rate 10^{-4} for TinyClinicalBERT and $5 \cdot 10^{-5}$ for Bio_ClinicalBERT, head 10^{-3}, weight decay 0.01), 10% linear warm-up then linear decay, 8 documents per step, bf16 autocast and the same capped inverse-frequency positive weights. Training runs for up to 30 epochs, with early stopping on validation micro-F1 (patience 6). Each encoder is trained with seeds 0–4 on the same fixed split. The other neural multi-label models use Adam (learning rate 10^{-3}, batch 64), binary cross-entropy with inverse-frequency positive weights (capped at 30), a maximum of 40 epochs and early stopping on validation micro-F1 (patience 8). Inputs are truncated to 512 tokens, which covers about 90% of documents.

Protocol and metrics. A single global decision threshold is tuned per model on validation micro-F1, then applied to test. We report micro-/macro-F1 (macro over labels with at least one test positive), micro-/macro-ROC-AUC, precision@5, recall@10 and document-level mean average precision. We report 95% bootstrap intervals (1,000 resamples of test documents) and paired bootstrap differences. The CPU models (linear, Word2Vec-based and frozen-feature) are trained with a single seed; the linear models are deterministic given the split. The fine-tuned encoders are trained with five seeds and reported as mean$\pm$SD. For them, bootstrap intervals and paired tests resample test documents and average micro-F1 over seeds.

Table 2: CodiEsp-50 test results (official dev set, 250 documents). CPU models: single seed; fine-tuned encoders: mean±SD over 5 seeds. All rows share the split, label set and validation-only threshold. Best value per column in bold.

Model	Seeds	μF1	MF1	μAUC	MAUC	P@5	R@10	MAP
Label-frequency prior	1	0.214	0.025	0.650	0.496	0.169	0.424	0.283
TF-IDF+LR	1	0.406	0.302	0.874	0.879	0.284	0.705	0.507
TF-IDF+LR (balanced)	1	0.422	0.344	0.884	0.880	0.299	0.746	0.528
TF-IDF+SVM	1	0.419	0.359	0.887	0.882	0.300	0.753	0.528
Frozen TinyClinicalBERT+LR	1	0.342	0.274	0.810	0.802	0.253	0.605	0.422
W2V+BiLSTM (draft repro)	1	0.308	0.295	0.825	0.816	0.214	0.583	0.381
BiLSTM+label attn (LAAT)	1	0.187	0.189	0.732	0.726	0.138	0.401	0.235
CAML (CNN+label attn)	1	0.377	0.346	0.854	0.817	0.278	0.699	0.495
TinyClinicalBERT fine-tuned (PLM-ICD style)	5	0.712 ± 0.012	0.650 ± 0.015	0.950 ± 0.002	0.934 ± 0.002	0.440 ± 0.005	0.900 ± 0.009	0.792 ± 0.007
Bio_ClinicalBERT fine-tuned (PLM-ICD style)	5	**0.738 ± 0.012**	**0.703 ± 0.019**	**0.954 ± 0.002**	**0.937 ± 0.004**	**0.451 ± 0.005**	**0.909 ± 0.009**	**0.805 ± 0.009**

Table 3: CodiEsp-Cat50 (top-50 three-character ICD-10 categories) and CodiEsp-Full (692 codes, linear models) test results (fine-tuned encoders: mean±SD over 5 seeds).

Model	Seeds	μF1	MF1	μAUC	MAUC	P@5	R@10	MAP
Label-frequency prior	1	0.214	0.040	0.644	0.510	0.178	0.379	0.256
TF-IDF+LR	1	0.435	0.359	0.877	0.876	0.335	0.716	0.551
TF-IDF+LR (balanced)	1	0.454	0.390	0.884	0.877	0.346	0.738	0.571
TF-IDF+SVM	1	0.453	0.378	0.883	0.876	0.342	0.737	0.565
Frozen TinyClinicalBERT+LR	1	0.372	0.336	0.814	0.803	0.297	0.637	0.468
W2V+BiLSTM (draft repro)	1	0.358	0.333	0.834	0.815	0.274	0.642	0.429
BiLSTM+label attn (LAAT)	1	0.270	0.249	0.779	0.766	0.221	0.544	0.341
CAML (CNN+label attn)	1	0.392	0.390	0.854	0.822	0.317	0.698	0.494
TinyClinicalBERT fine-tuned (PLM-ICD style)	5	0.627 ± 0.015	0.562 ± 0.021	0.923 ± 0.004	0.903 ± 0.004	0.464 ± 0.009	0.844 ± 0.021	0.724 ± 0.013
Bio_ClinicalBERT fine-tuned (PLM-ICD style)	5	**0.686 ± 0.008**	**0.631 ± 0.015**	**0.939 ± 0.006**	**0.923 ± 0.007**	**0.493 ± 0.009**	**0.880 ± 0.015**	**0.781 ± 0.018**

Model	Seeds	μF1	MF1	μAUC	MAUC	P@5	R@10	MAP
Label-frequency prior	1	0.141	0.005	0.768	0.484	0.169	0.161	0.130
TF-IDF+LR	1	0.281	0.069	0.881	0.861	0.298	0.324	0.267
TF-IDF+LR (balanced)	1	0.301	0.081	0.893	**0.864**	0.322	0.354	0.296
TF-IDF+SVM	1	**0.304**	**0.139**	**0.899**	0.855	**0.326**	**0.380**	**0.305**

5 Results

Main comparison. Tables 2–3 and Fig. 3 report the results. On CodiEsp-50 the best linear model reaches micro-F1 0.422 and macro-F1 0.359. TF-IDF+LR has a bootstrap 95% CI of [0.374, 0.435] for micro-F1. The draft's W2V+BiLSTM reaches 0.308, CAML 0.377, BiLSTM+LAAT 0.187, and frozen TinyClinicalBERT+LR 0.342. Paired bootstrap differences in micro-F1 between TF-IDF+LR and each of these models are W2V+BiLSTM +0.097 [0.066, 0.129], CAML +0.029 [-0.003, 0.061], BiLSTM+LAAT +0.219 [0.191, 0.248], frozen TinyClinicalBERT+LR +0.062 [0.035, 0.090]. The intervals exclude zero for the BiLSTM, LAAT and frozen-TinyClinicalBERT comparisons but not for CAML on CodiEsp-50. On CodiEsp-Cat50, TF-IDF+LR is again significantly better than every neural model, including CAML (+0.044, [0.018, 0.071]). Still, CAML ties the best linear model on Cat50 macro-F1 (0.390), which suggests that label attention mainly helps the rarer labels. On the full label space, absolute micro-F1 drops to 0.304 and macro-F1 to 0.139, because most codes have only a handful of training examples. With only 425 training documents, models that learn sequence representations largely from scratch cannot exploit their extra capacity; LAAT, the most heavily parameterised, is the weakest. Among these models, sparse lexical features capture the strongest cue: the diagnosis term itself is usually written in the case report. Fine-tuned pretrained encoders change this picture (§5).

Fine-tuned clinical encoders. Fine-tuning is the largest effect in the study (Tables 2–3, Fig. 4). On CodiEsp-50, Bio_ClinicalBERT reaches micro-F1 0.738±0.012 and macro-F1 0.703±0.019 (MAP 0.805±0.009). TinyClinicalBERT reaches 0.712±0.012 / 0.650±0.015, against 0.406 / 0.302 for TF-IDF+LR. The seed-pooled paired bootstrap difference in micro-F1 over TF-IDF+LR is +0.333 [0.302, 0.366] for Bio_ClinicalBERT and +0.307 [0.277, 0.338] for TinyClinicalBERT (Holm-corrected $p < 0.001$ for both). On CodiEsp-Cat50 the corresponding numbers are 0.686±0.008 and 0.627±0.015 micro-F1 (macro 0.631±0.015 and 0.562±0.021), with differences +0.250 [0.221, 0.279] and +0.192 [0.164, 0.222]. The larger encoder helps more on the category task (+0.059 micro-F1) than on CodiEsp-50 (+0.026). Seed SDs are 0.008–0.015 micro-F1, an order of magnitude smaller than the gap to the linear models. Fine-tuning the frozen TinyClinicalBERT features end-to-end with label attention doubles their micro-F1 (0.342 → 0.712±0.012). The pretrained representation is

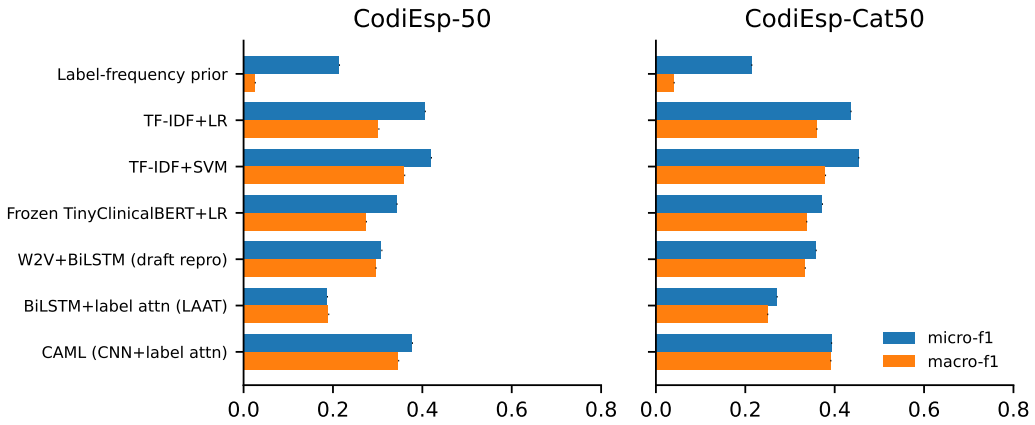
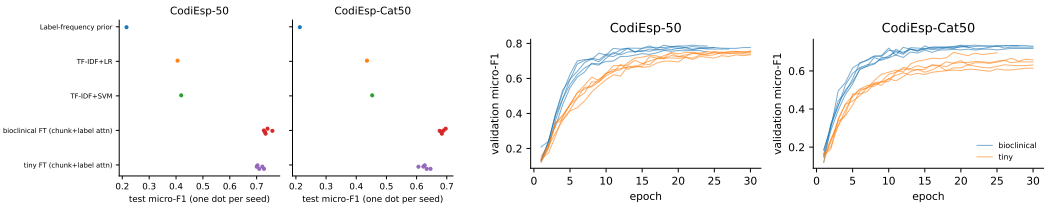


Figure 3: Micro- and macro-F1 of the CPU models on the two top-50 label spaces (single seed; fine-tuned encoders in Table 2 and Fig. 4).



(a) Test micro-F1 per seed

(b) Validation micro-F1 per epoch

Figure 4: Fine-tuned encoders versus the linear baselines on one RTX 3090. (a) One dot per seed. (b) Training curves of all fine-tuning runs.

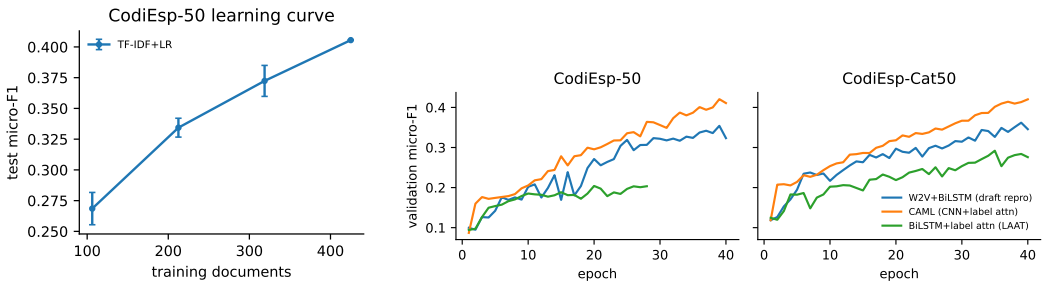
therefore useful at this data scale, but only when it is adapted. A run takes 1.8 (TinyClinicalBERT) and 4.6 (Bio_ClinicalBERT) minutes on CodiEsp-50 on a single RTX 3090.

Is the gap real? Sanity checks. Because the jump is large, we checked the fine-tuning pipeline against the baselines. (a) *Same data.* The fine-tuning runs rebuild the split with the notebook's code. The train/validation/test document IDs, all 752 input files (750 case texts and two label files, SHA-256) and the label sets are identical to those of the baselines. The 50 CodiEsp-50 labels match in order; Cat50 has the same 50 categories, with two tied categories in swapped column order, which does not affect any metric. All 20 runs' test label matrices equal the baselines' after alignment. (b) *Same protocol.* Every stored threshold equals the one recomputed from the validation scores, and the test metrics recompute exactly from the saved test scores. Choosing the threshold on test instead would add only 0.008 micro-F1 on average. (c) *No overlap.* No document ID or text is shared between train, validation and test. The maximum TF-IDF cosine between a test and any training document is 0.51. (d) *No label strings in the input.* The encoders see only the case text, and no test document contains any of its gold code strings. (e) *Shuffled-label control.* Fine-tuning TinyClinicalBERT on CodiEsp-50 with the training labels permuted across documents (seed 0, otherwise identical) gives test micro-F1 0.141, micro-AUC 0.600 and MAP 0.194. That is no better than the label-frequency prior (0.214 micro-F1), so the gain requires the real text–label pairing. (f) *Stronger linear baseline.* A wider TF-IDF+LR search adds character 3–5-grams and C up to 1000, again selected on validation. It reaches only 0.455 on CodiEsp-50 and 0.471 on Cat50. We therefore attribute the gap to the pretrained encoder, end-to-end fine-tuning, long-document segmenting and label attention, not to a difference in evaluation.

Ablations. Table 4 varies one factor of the draft's design at a time on CodiEsp-50. *Class weighting* is the most important factor. Without inverse-frequency positive weights the BiLSTM collapses to the label prior (micro-F1 0.308→0.207, macro-F1 0.295→0.025), and CAML loses 0.047 micro-F1 and 0.117 macro-F1. *Bidirectionality* helps: a unidirectional LSTM loses 0.039 micro-F1. The earlier

Table 4: Ablations on CodiEsp-50 (single seed). Each row changes one factor relative to its parent model.

Model	Seeds	μF1	MF1	μAUC	P@5	MAP
W2V+BiLSTM (draft repro)	1	0.308	0.295	0.825	0.214	0.381
BiLSTM: no class weighting	1	0.207	0.025	0.643	0.169	0.282
BiLSTM: unidirectional LSTM	1	0.269	0.260	0.806	0.188	0.327
BiLSTM: ICD-title W2V init	1	0.293	0.281	0.793	0.208	0.383
BiLSTM: random init	1	0.353	0.320	0.846	0.258	0.445
CAML (CNN+label attn)	1	**0.377**	**0.346**	**0.854**	**0.278**	**0.495**
CAML: no class weighting	1	0.330	0.229	0.811	0.245	0.416
CAML: random init	1	0.360	0.320	0.833	0.260	0.441
CAML: no label attention (CNN+max)	1	0.348	0.283	0.844	0.242	0.426



(a) Learning curve

(b) Validation micro-F1 per epoch

Figure 5: (a) TF-IDF+LR test micro-F1 versus training-set size on CodiEsp-50. (b) Neural training curves.

draft asserted a +4.1% bidirectional gain without evidence; our single-seed estimate is of similar size but must not be read as confirming it. *Embedding initialisation* gives no reliable gain. Word2Vec trained on ICD titles, as in the original code, covers only 35.6% of the CodiEsp vocabulary and gives 0.293; random initialisation gives 0.353 for the BiLSTM (0.360 for CAML, against 0.377 with in-domain Word2Vec). With one seed these differences are within plausible optimisation noise. *Label attention* helps CAML: replacing it with max-pooling lowers micro-F1 from 0.377 to 0.348 and macro-F1 from 0.346 to 0.283.

Data efficiency. The TF-IDF+LR learning curve (Fig. 5a; 3 random subsets per size) rises from micro-F1 0.269 at 25% of training documents to 0.406 at 100% and has not saturated. Macro-F1 rises from 0.107 to 0.302. More data, rather than more model capacity, is the binding constraint at this scale.

Error analysis. Per-label F1 correlates with training frequency (Spearman ρ=0.53 for TF-IDF+LR on CodiEsp-50; Fig. 6). Of the 50 labels, TF-IDF+LR gets F1$=0$ on 13, the W2V+BiLSTM on 2 and CAML on 3. The weighted neural models leave fewer labels at zero, but they pay for this with many false positives. Twenty-three of the 50 CodiEsp-50 labels are vague symptom codes from ICD-10 chapter R, and these dominate CAML's errors. R69 (illness, unspecified) has 32 false negatives and 47 false positives. R58 (haemorrhage) has 40 false positives against 2 true positives, and R53.1 (weakness) has 38 false positives against 2 true positives. Symptoms are mentioned far more often than they are coded, so lexical presence alone is a poor cue for these labels. On CodiEsp-Cat50, TF-IDF+LR ranges from micro-F1 0.085 (chapter L) to 0.948 (chapter Z) by ICD-10 chapter, and the 18 symptom categories of chapter R get 0.442.

Secondary task: Symptom2Disease. On this short, single-label task all model families do well (Table 5). TF-IDF+SVM reaches 0.953 accuracy, fine-tuned TinyClinicalBERT 0.948 and the W2V+BiLSTM 0.943, so the spread between families is small. The most frequent confusions are clinically plausible, for example dengue→typhoid and peptic ulcer→GERD (Fig. 7). Near-ceiling scores on this dataset should not be read as evidence of clinical coding ability.

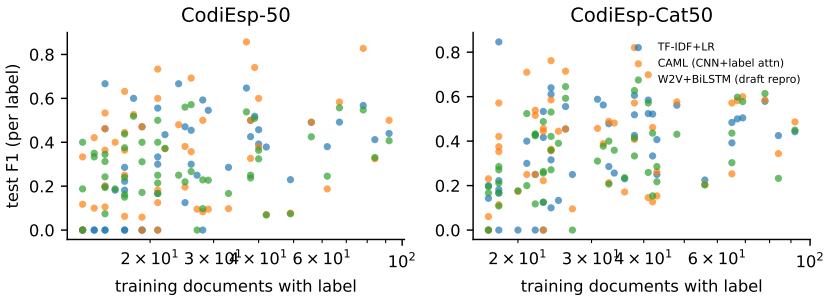


Figure 6: Per-label test F1 versus training frequency (CodiEsp-50).

Table 5: Symptom2Disease test set (212 descriptions, 22 classes, single seed).

Model	Accuracy	Macro-F1
TF-IDF+LR	0.943	0.943
TF-IDF+SVM	0.953	0.952
W2V+BiLSTM	0.943	0.944
W2V+CNN (max-pool)	0.910	0.911
TinyClinicalBERT (FT)	0.948	0.947

6 Claims from the earlier draft that we do not reproduce

For transparency, the following statements from the earlier NLPK draft are *not* supported by any artefact we could inspect, and are not used in this paper: a macro-F1 of 0.837 on 500 ICD-9 codes (the code leaks the target, there is no note text, and the evaluation notebook uses random labels); a "41% error reduction" in an NHS Digital pilot, a 32 req/s load test and a 3.2× inference speed-up (no logs, pilot records or deployment code); UMLS synonym expansion, back-translation augmentation and a +4.1% bidirectional-over-unidirectional gain (not implemented; our own uni- vs bidirectional ablation is in Table 4); and two references we could not locate ("Uvaha & Smith 2016"; "Liu 2021, IEEE Transactions on Medical Informatics").

7 Limitations

(i) **Data.** The English CodiEsp texts are machine translations of Spanish case reports, not hospital discharge summaries. The corpus is small (425 training documents), and we test on the official dev set because the test labels are not in the public mirror. We did not use any credentialed clinical data (e.g. MIMIC-III/IV discharge summaries), so results may not transfer to hospital records such as MIMIC or the NHS. (ii) **Statistics.** The fine-tuned encoders use five seeds on one fixed split. The CPU models use one seed, so their bootstrap intervals capture test-sample variance but not optimisation variance, and no split was re-drawn. (iii) **Tuning budget.** The CPU neural models received a small, fixed budget (about 13 minutes for the whole notebook). The fine-tuned encoders used one hyperparameter setting, with no search. Both families might improve with tuning, and the from-scratch models probably more than the fine-tuned ones. (iv) **Part 1** uses the 100-patient public MIMIC-III demo, so it diagnoses the pipeline rather than estimating any clinical performance.

8 Broader impact

Coding errors affect reimbursement, epidemiological statistics and patient records. Overstated accuracy claims, especially those driven by leakage, can lead to premature deployment. We release the leakage diagnosis and an executable protocol so that future claims can be checked. Our models are research artefacts and are not suitable for unsupervised clinical use. All data used are public and de-identified or synthetic-style; no credentialed patient data were accessed.

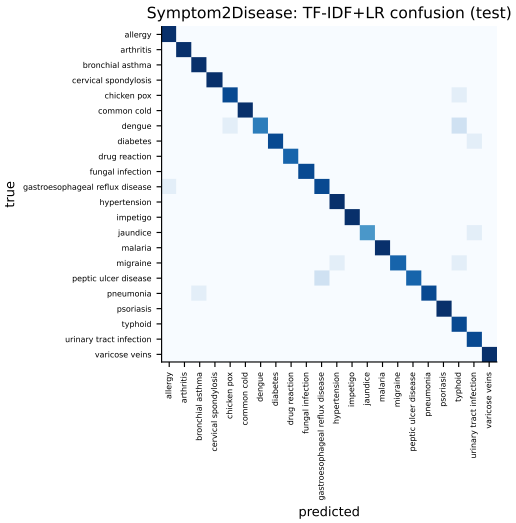


Figure 7: Symptom2Disease confusion matrix (TF-IDF+LR).

Acknowledgments and Disclosure of Funding

The author thanks the maintainers of the CodiEsp, MTSamples, Symptom2Disease and MIMIC-III demo resources. No external funding was received for this work.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019. doi: 10.18653/v1/W19-1909.

Kannan Sekar Annu Radha. NLPK: ICD-9 prediction with Word2Vec and LSTM. GitHub repository, `https://github.com/KannanSA/NLPK`, commit 381727d, 2025.

Barcelona Supercomputing Center, Text Mining Unit. CodiEsp corpus (train/dev/test, with english machine translations). Hugging Face dataset, `https://huggingface.co/datasets/Biomedical-TeMU/CodiEsp_corpus`, CC-BY-4.0, 2026. Accessed 2 November 2026.

Olivier Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270, 2004. doi: 10.1093/nar/gkh061.

Joakim Edin, Alexander Junge, Jakob D. Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo, and Lars Maaløe. Automated medical coding on MIMIC-III and MIMIC-IV: A critical review and replicability study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2572–2582, 2023. doi: 10.1145/3539618.3591918.

Joakim Edin, Andreas Motzfeldt, Simon Flachs, and Lars Maaløe. Symphony for medical coding: A next-generation agentic system for scalable and explainable medical coding. *arXiv preprint arXiv:2603.29709*, 2026.

Yidong Gan, Maciej Rybinski, Ben Hachey, and Jonathan K. Kummerfeld. Aligning AI research with the needs of clinical coding workflows: Eight recommendations based on US data analysis and critical review. *arXiv preprint arXiv:2412.18043*, 2024.

Gretel.ai. symptom_to_diagnosis. Hugging Face dataset, `https://huggingface.co/datasets/gretelai/symptom_to_diagnosis`; adapted from the Kaggle Symptom2Disease dataset, `https://www.kaggle.com/datasets/niyarrbarman/symptom2disease`, 2023. Accessed 2 November 2026.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20, 2022. doi: 10.18653/v1/2022.clinicalnlp-1.2.

Alistair Johnson, Tom Pollard, and Roger Mark. MIMIC-III clinical database demo (version 1.4). PhysioNet, `https://physionet.org/content/mimiciii-demo/1.4/`, 2019.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016. doi: 10.1038/sdata.2016.35.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751, 2014. doi: 10.3115/v1/D14-1181.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020. doi: 10.1093/bioinformatics/btz682.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Antonio Miranda-Escalada, Aitor Gonzalez-Agirre, Jordi Armengol-Estapé, and Martin Krallinger. Overview of automatic clinical coding: Annotations, guidelines, and solutions for non-English clinical cases at CodiEsp track of CLEF eHealth 2020. In *Working Notes of CLEF 2020 – Conference and Labs of the Evaluation Forum*, volume 2696 of *CEUR Workshop Proceedings*, 2020. URL `https://ceur-ws.org/Vol-2696/paper_263.pdf`.

MTSamples.com. MTSamples medical transcriptions (hugging face mirror). `https://huggingface.co/datasets/harishnair04/mtsamples`; original source `https://www.mtsamples.com`, 2026. Accessed 2 November 2026.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of NAACL-HLT*, pages 1101–1111, 2018. doi: 10.18653/v1/N18-1100.

Adam Paszke et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Fabian Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237, 2014. doi: 10.1136/amiajnl-2013-002159.

Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

Omid Rohanian, Mohammadmahdi Nouriborji, Hannah Jauncey, Samaneh Kouchaki, Farhad Nooralahzadeh, Lei Clifton, Laura Merson, David A. Clifton, and ISARIC Clinical Characterisation Group. Lightweight transformers for clinical natural language processing. *Natural Language Engineering*, pages 1–28, 2023. doi: 10.1017/S1351324923000542.

Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. A label attention model for ICD coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3335–3341, 2020. doi: 10.24963/ijcai.2020/461.

Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP 2020: System Demonstrations*, pages 38–45, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

A Reproducibility details

Environment: Python 3.13.5, PyTorch 2.15.0+cpu, scikit-learn 1.9.1, gensim 4.4.0, transformers 5.22.1, on 8 CPU threads. The full notebook ran end-to-end in 13 minutes. Hyperparameter grids: LR $C \in \{1, 10, 100\}$, SVM $C \in \{0.1, 0.3, 1\}$, and threshold candidates at 120 validation-score quantiles in $[0.5, 0.999]$. Fine-tuning ran on one NVIDIA GeForce RTX 3090 (24 GB) under Windows with Python 3.14.8, PyTorch 2.11.0+cu128, transformers 5.22.1 and bf16 autocast, with `PYTHONHASHSEED=0`. The 20 runs (2 encoders $\times$ 2 label spaces $\times$ 5 seeds) took 66 minutes in total. Mean epochs to early stopping were 30.0 / 26.0 (TinyClinicalBERT / Bio_ClinicalBERT, CodiEsp-50). Seed-pooled bootstrap intervals use 2,000 resamples. Code, tables, figures, per-run JSON/score files and `results.json` are included in the supplementary bundle.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim only results reported in Sections 3 and 6, all produced by the released notebook.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section "Limitations discusses machine-translated data, small corpus, single-seed CPU models (five seeds for the fine-tuned encoders), limited tuning budget, no credentialed data and the demo-only Part 1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper contains no theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendix A give data splits, label spaces, architectures, hyperparameters and seeds; the executed notebook and fine-tuning scripts reproduce every number.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets are openly downloadable (URLs in references); the notebook, source and results are in the supplementary bundle.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 4 and Appendix A specify splits, grids, optimiser, epochs, early stopping and threshold selection.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report 95% bootstrap intervals over test documents and paired bootstrap differences; fine-tuned encoders are reported as mean$\pm$SD over five seeds with seed-pooled paired bootstrap tests; single-seed CPU training is stated as a limitation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A reports the CPU hardware (8 threads), the GPU (one RTX 3090) and runtimes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Only public, de-identified or non-patient data are used; no credentialed data were accessed.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the Broader impact section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No high-risk models or scraped datasets are released.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: CodiEsp (CC BY 4.0), MIMIC-III demo (ODbL), Symptom2Disease (Gretel release, Apache-2.0) and MTSamples (Hugging Face mirror, Apache-2.0 per its card; used only for unsupervised embedding pre-training) are cited with their sources.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released notebook and code are documented; no new dataset is created.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing or human subjects research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not used as a component of the methods.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.