Modular Clinical Decision Support Networks (MoDN)—Updatable, Interpretable, and Portable Predictions for Evolving Clinical Environments

Cécile Trottet[1☯] Thijs Vogels[1☯] Kristina Keitel[2], Alexandra V Kulunkina[3,4], Rainer Tan[5,6], Ludovico Cobuccio[5], Martin Jaggi[7], Mary-Anne Hartley[1*],

1 Intelligent Global Health Research Group, Machine Learning and Optimization Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Vaud, Switzerland
2 Pediatric Emergency Centre, University Hospital of Bern (Inselspital), Bern, Switzerland
3 Digital Health Unit, Swiss Center for International Health, Swiss Tropical and Public Health Institute, Basel, Switzerland
4 University of Basel, Basel, Switzerland
5 Clinical Research Unit, Swiss Tropical and Public Health Institute (Swiss TPH), Basel, Switzerland
6 Ifakara Health Institute, Tanzania
7 Machine Learning and Optimization Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Vaud, Switzerland

☯These authors contributed equally to this work.
* mary-anne.hartley@epfl.ch

Abstract

Background

Clinical Decision Support Systems (CDSS) have the potential to improve and standardise care with probabilistic guidance. However, many CDSS deploy static, generic rule-based logic, resulting in inequitably distributed accuracy and inconsistent performance in evolving clinical environments. Data-driven models could resolve this issue by updating predictions according to the data collected. However, the size of data required necessitates collaborative learning from analogous CDSS's, which are often imperfectly interoperable (IIO) or unshareable. We propose Modular Clinical Decision Support Networks (MoDN) which allow flexible, privacy-preserving learning across IIO datasets as well as being robust to the systematic missingness common to CDSS-derived data, while providing interpretable, continuous predictive feedback to the clinician.

Methods & Findings

MoDN is a novel decision tree composed of feature-specific neural network modules. It creates dynamic personalised representations of patients, and can make multiple predictions of diagnoses and features, updatable at each step of a consultation. The model is validated on a real-world CDSS-derived dataset, comprising 3,192 paediatric outpatients in Tanzania.

MoDN significantly outperforms 'monolithic' baseline models (which take all features at once at the end of a consultation) with a mean macro F_1 score across all diagnoses of 0.749 vs 0.651 for logistic regression and 0.620 for multilayer perceptron ($p < 0.001$).

To test collaborative learning between IIO datasets, we create subsets with various percentages of feature overlap and port a MoDN model trained on one subset to another. Even with only 60% common features, fine-tuning a MoDN model on the new dataset or just making a composite model with MoDN modules matched the ideal scenario of sharing data in a perfectly interoperable setting.

Interpretation

MoDN integrates into consultation logic by providing interpretable continuous feedback on the predictive potential of each question in a CDSS questionnaire. The modular design allows it to compartmentalise training updates to specific features and collaboratively learn between IIO datasets without sharing any data.

Funding

Botnar Foundation (grant n°6278)

Author summary

Clinical Decision Support Systems (CDSS) are emerging as a standard-of-care, offering probabilistic guidance at the bedside. Many deploy static, generic rule-based logic, resulting in inconsistent performance in evolving environments. Machine learning (ML) models could resolve this by updating predictions according to the collected data. However, traditional methods are often criticised as uninterpretable "black-boxes" and are also inflexible to fluctuations in resources: requiring retraining (and costly re-validation) each time a question is altered or added.

We propose MoDN: a novel, interpretable-by-design, modular decision tree network comprising a flexible composition of question-specific neural network modules, which can be assembled in real-time to build tailored decision networks at the point-of-care, as well as enabling collaborative model learning between CDSS with differing questionnaire structures without sharing any data.

1 Introduction

Probabilistic decision-making in medicine has the potential to bypass costly and invasive clinical investigations, and holds particular promise to reduce resource consumption in low-income settings. [1] However, it is impossible for any clinician to memorise the increasingly complex, evolving and sometimes conflicting probabalistic clinical guidelines [2], which has driven the need for Clinical Decision Support Systems (CDSS) that summarise guidance into simple rule-based decision trees. [3–5] The digitalization of some commonly used CDSS into mobile apps has shown promise in increasing access and adherence to guidelines, while laying the foundation for more systematic data collection. [6–8]

Despite their promise to bridge the 'know-do gap', a surprisingly low proportion of popular guideline recommendations are backed by high quality evidence ('know'), [9] and an even lower proportion of mobile tools have been rigorously tested in practice ('do'). [10] While likely to generally improve and standardise care, their static and generic logic would result in inconsistent performance in light of changing epidemiology as well as an inequitable distribution of accuracy in underrepresented populations. [1]

To address these limitations, there is a move toward data-driven predictions that incorporates machine learning (ML), [11–14] with the goal to leverage more complex multi-modal data and derive self-evolving algorithms [15]. The WHO SMART guidelines [16] advocate for a faster and more systematic application of digital tools. More specifically, the last layer of these guidelines reflects the use of dynamic, big-data-driven algorithms for optimised outcomes and updatable recommendations. This move is predicted to improve CDSS safety and quality, [17] especially in low-resource settings. [18, 19] Additionally, as such models improve with the addition of good quality data, they inherently incentivise better data collection, a positive feedback termed the *CDSS Loop* [20].

Regardless, the data collected with decision tree logic is fundamentally flawed by biased missingness. [21] Patients are funneled into high-yield question branches, yielding systematic missing values for questions that are not asked. Models trained on such data can easily detect patterns in the missingness of features rather than in their values, thus, not only failing to improve on the rule-based system, but also becoming clinically irrelevant.

Such issues of data quality and utility are secondary to the more general limitation of availability. Patient-level data is rarely shared due to well-considered concerns of privacy and ownership. The inability to share data fragments statistical power, compromises model fairness, [22] and results in poor interoperability, where CDSS users and developers do not align data collection procedures.

The latter can limit collaborative learning across analogous CDSS tools, restricting them to use only features that are available to all participants. [11]

Even if the above issues of data quality, availability and interoperabilty are resolved (for example, with the powerful cross-EHR solution proposed by Google Research [15]), the 'updating' of ML models still poses a major regulatory issue which may make them unimplementable. [23] 'Perpetual updating' is one of the main motivations for the transition to ML [14] to combat 'relevance decay', [24] which can be dramatic in rapidly evolving environments with changing epidemiology and unreliable resources. Indeed, after the laborious processes of data collection, cleaning, harmonisation, model development and clinical validation, the tool could already be outdated. The problem is that while ML models can learn autonomously from updating data streams, each update requires whole-model retraining that invalidates the previous version, which is likely to make the promise of perpetual updates unfeasible.

In this work, we propose the Modular Decision Support Network (MoDN) to provide dynamic probablistic guidance in decision-tree based consultations. The model is

extended during the course of the consultation, adding neural network *modules* specific to each question asked. This results in a dynamic representation of the patient able to predict the probability of various diagnoses at each step of a consultation.

We validate MoDN on a real world CDSS-derived data set of 3000 pediatric outpatient consultations and show how the feature-wise modular design addresses the issues above, such as collaborative learning from imperfectly interoperable (IIO) datasets with systematic missingness, while improving data availability, model fairness and interpretability. The feature-wise modularisation makes MoDN interpretable-by-design, whereby it aligns its learning process with the clinician, in step-by-step "consultation logic". It can thus provide continuous feedback at each step, allowing the clinician to directly assess the contribution of each feature to the prediction at the time of feature collection. A major contribution, is the possibility to compartmentalise updates to affected features, thus retaining validity.

Materials and methods

Below, we describe MoDN, a deep learning CDSS composed of interchangeable, feature-specific neural network modules. We validate it on a real-world CDSS-derived data set and visualise its capacity to represent patient groups and predict multiple diagnoses at each stage of the decision tree. Two additional experiments are designed to test its portability to imperfectly interoperable data sets, and compartmentalise updates to a targeted feature sub-set.

Data set

Cohort description

We train, test and validate MoDN on a CDSS-derived data set comprising 3,192 pediatric (aged 2–59 months) outpatient consultations presenting with acute febrile illness. The data was collected in nine outpatient departments across Dar es Salaam, Tanzania between 2014 and 2016 as part of a randomised control trial on the effect of CDSS on antibiotic use, hereafter referred to as ePOCT. [25] The data had over 200 unique feature sets of asked questions (i.e. unique combinations of decision branches in the questionnaire).

Ethics

Written informed consent was obtained from the caregivers of all participants as described in Keitel et al.ePOCT. [25] The study protocol and related documents were approved by the institutional review boards of the Ifakara Health Institute and the National Institute for Medical Research in Tanzania, by the Ethikkommission Beider Basel in Switzerland, and the Boston Children's Hospital ethical review board. An independent data and safety monitoring board oversaw the study. The trial was registered in ClinicalTrials.gov, identifier NCT02225769.

Features and targets

A subset of eight diagnoses and 33 features were selected in order to ensure interpretable reporting and limit computational cost. Selection of both targets and features were based primarily on prevalence (i.e. retaining the most prevalent). Features were additionally tested for predictive redundancy in bivariate Pearson's correlation, and strongly collinear features were randomly dropped. The features comprise

demographics, medical history, clinical signs and symptoms and laboratory results collected at the time of consultation and are detailed in Table S1.

MoDN aims to simultaneously predict eight retrospectively derived diagnoses, namely anaemia, dehydration, diarrhoea, fever without source, malaria, malnutrition, pneumonia, and upper respiratory tract infection. Patients can have none or several of these diagnoses.

It is also possible to predict (impute) any of the 33 missing features; experimentation on feature decoding is detailed in the supplement (p 18).

Pre-processing

As explained, the decision tree logic of questionnaires in CDSS-derived data creates systematic missingness, where diagnostic endpoints may have unique feature sets. We exploit these patterns to derive the consultation logic (i.e. order of asked questions) and thus align training with clinical protocol. For groups of questions that are either all present or all missing according to the outcome, relative ordering is impossible, and thus randomised.

To ensure that our model performs well for patients outside of the training data set, we randomly partition the data into train ($n = 1914, 60\%$), validation ($n = 639, 20\%$), and test ($n = 639, 20\%$) splits. We optimize the model on the training set only, tune its hyperparameters based on the validation split, and report its final performance on the test split that was not used in creating the model. We then obtain a distribution of estimates using five iterations of two-fold cross validation [26], where data is randomly re-partitioned.

MoDN

Model architecture

MoDN comprises three core elements: *encoders*, *decoders*, and the *state* as listed below and summarised in Fig 1.

- The *state*, $\mathbf{s}$, is the vector-representation of a patient. It evolves as more answers are recorded.
- *Encoders* are feature-specific and update the *state* with the value of a newly collected feature based on the current version of the *state*.
- *(Diagnosis) decoders* are output-specific and extract predictions from the *state* at any stage of the consultation. Predicted outputs can be any data set element, including the features themselves. These *feature decoders* provide a dynamic patient-specific imputation of values not yet recorded. All feature-decoding experimentation is detailed in the supplement (p 18) in figures S1 and S2.

Encoders and decoders are thus respectively feature or output specific multilayer perceptrons (MLP). This modularises both the input space as well as the predictions made in the output space. More details are provided in the appendix (p 17).

We consider the consultation data of a patient as an ordered list of (question, answer) pairs, $(q_1, a_1), (q_2, a_2), \ldots, (q_T, a_T)$. The ordering of questions asked simultaneously is randomized. As new information is being collected, the *state* vector $\mathbf{s} \in \mathbb{R}^s$ evolves as:

$$\mathbf{s}_0 = \mathcal{S}_0 \in \mathbb{R}^s, \text{ a trained constant}, \tag{1}$$

$$\mathbf{s}_t = \mathcal{E}_{q_t}(\mathbf{s}_{t-1}, a_t), \quad \text{for } t = 1 \ldots T, \tag{2}$$

where $\mathcal{E}_{q_t} : (\mathbb{R}, \mathbb{R}^s) \to \mathbb{R}^s$ is an *encoder* specific to question q_t. It is a small MLP with trainable parameters.

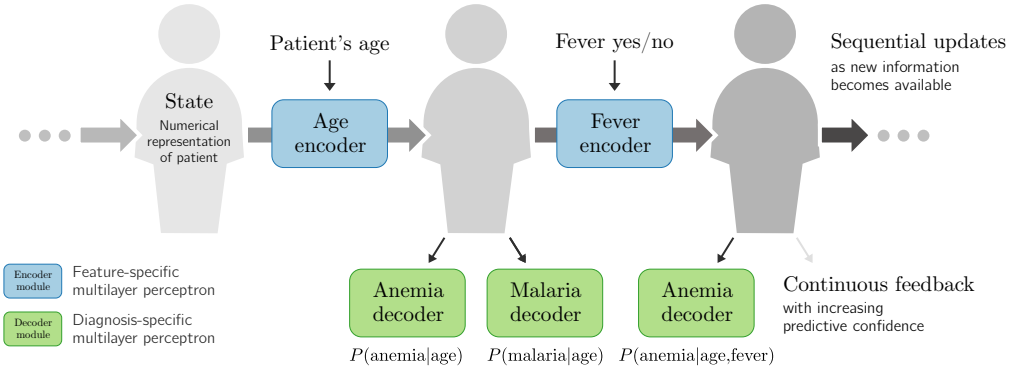


Fig 1. The Modular Clinical Decision Support Network (MoDN). The *state* is a representation of the patient, which is sequentially modified by a series of inputs. Here, we show in blue *age* and *fever* values as modifying inputs. Each input has a dedicated encoder which updates the *state*. At any point in this process, the clinician can either apply new encoders (to update the *state*) or decode information from the *state* (to make predictions, in green).

After gathering the first t answers, the probability of a diagnosis d is then predicted as:

$$p_t(d) = \mathcal{D}_d(\mathbf{s}_t), \tag{3}$$

where each *decoder* $\mathcal{D}_d : \mathbb{R}^s \to [0, 1]$ is also a small MLP with trainable parameters.

Model optimisation

We optimize the parameters of the trainable components of MoDN, written as calligraphic symbols, using a training set of C completed consultations. In addition to a list of questions and answers, each consultation also features binary 'ground-truth' labels y_d^c that indicate whether consultation c was diagnosed with d. Following the principle of empirical risk minimisation, our training objective is a sum over C consultations c, but also over D potential diagnoses d, and T 'time-steps':

$$\min \sum_{c=1}^{C} \sum_{d=1}^{D} \sum_{t=0}^{T} \ell\left(p_t^c(d), y_d^c\right) + R, \tag{4}$$

The parameters to be optimized are implicit in $p_t^c(d)$. For binary diagnoses, ℓ is cross-entropy loss. The inclusion of different time-steps in the objective ensures that MoDN can make predictions at any stage of the consultation.

The regularization term R in our objective ensures that the *states* do not change more than necessary to encode the new information:

$$R = \frac{1}{s} \sum_{c=1}^{C} \sum_{t=1}^{T} \|\mathbf{s}_t^c - \mathbf{s}_{t-1}^c\|^2. \tag{5}$$

We optimize Equation 4 with the Adam optimizer [27]. For each step, we sample a batch of consultations, and sum the decoder losses at the multiple intermediate time-steps T. We randomize the order within blocks of simultaneously asked questions.

Visualizing the MoDN *state*

The *state* **S** may become highly dimensional in complex data sets rendering it uninterpretable. To gain some visual insight into this vectorised representation of a patient, we use the t-distributed stochastic neighbor embedding (t-SNE) [28] dimensionality reduction algorithm, where similar data points are mapped close to each other in a lower dimensional embedding. By overlaying the data points (*states*) with a colour representing the diagnosis, we can visualise how well the *state* represents the predicted label.

Baseline models for MoDN performance

MLP and logistic regression are used as baselines to compare with MoDN for binary diagnostic classification tasks (i.e. one model per diagnosis). Train-test splits and pre-processing is identical to MoDN with the exception of imputation. As traditional ML models cannot handle missing values, mean value imputation was performed. Performance is reported as macro F1 scores (the harmonic mean of precision and sensitivity). Models are compared with a paired t-test on a distribution of performance estimates derived from a 5×2 cross validation as per Dietterich et al. [26] We also report calibration curve.

Experimental set up

A common issue when CDSS are updated in light of newly available resources (e.g. new questions/tests added to the CDSS) is incomplete feature overlap between old and new data sets. To test the capacity of MoDN in such imperfectly interoperable (IIO) settings, we simulate IIO subsets within the $3,192$-patient CDSS-derived data described previously. These IIO subsets are depicted in Table 1, where data sets A and B comprise $2,068$ and 516 patients respectively. Performance is then evaluated on an independent test set of size 320 (D), in which all of the features are available. Internal validation for each model is performed on the remaining 288 patients (C' and C for validation of data sets A and B respectively, which differ in the number of features provided). Three levels of IIO (90%, 80% and 60%) are simulated between data sets A and B by artificially deleting random features in A. These are compared to a baseline of perfectly interoperable feature sets (100% overlap). Within these data sets, all experiments are performed with $5-$fold randomisation of data set splits and available features to obtain a distribution of F1 scores which are averaged to a macro F1 score with 95% confidence interval.

New IIO user experiment: Modularised fine-tune

In this common scenario, a clinical site starts using a CDSS. It has slightly different resources and is thus IIO compared to more established implementation sites. However, it would still benefit to learn from these sites while ensuring that the unique trends in its smaller, local data set are preserved. Due to ethical constraints, data sets cannot be shared.

We hypothesise that MoDN is able to handle this scenario via 'modularised fine-tuning' as depicted in Figure 2. Here, a MoDN is pre-trained on the larger (unseen) data set A and ported to B where all of the modules are fine-tuned. Thus personalising the existing modules as well as adding new modules unique to B, thus creating a new collaborative IIO model without sharing data.

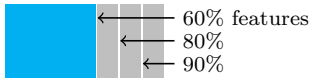
Split	Partition	Patients	Interoperable	Imperfectly interoperable
Train	Source (A)	2 068		
	Target (B)	516		
Validation	Source	288		
	Target	288		
Test	Source	288		

Table 1. Imperfectly interoperable (IIO) data sets. From the 3,192-patient CDSS-derived data set, we create two training sets with three levels of imperfect feature overlap (60, 80 and 90%) compared with perfect interoperability (100%). In our experiments, the owner of a small 'target' data set (fewer patients) wants to benefit from a larger 'source' data set without having access to this data. The 'source' may lack several features that are available in the 'target', yielding several levels of 'imperfect interoperability'. We construct validation sets with and without these missing features, as well as a held-out test set. The F1 scores we report in this paper are averages over five randomized folds of this data-splitting procedure.

New IIO resource scenario: Modularised update

In this common scenario, a site using a CDSS acquires new resources (e.g. new point-of-care diagnostic tools). It would like to update its CDSS model with the data collected from this new feature, but it cannot break the validity of the existing predictions that have been approved by the regulatory authority after a costly validation trial.

We hypothesise that MoDN is able to handle this scenario via feature-wise compartmentalisation of model updates. Keeping all experimental conditions identical as depicted in Figure 2, MoDN is pre-trained on the larger (unseen) data set A and ported to B (hosting new resources). The key difference is that no fine-tuning occurs. Rather, we seek to preserve the predictive validity of the MoDN modules of overlapping features by freezing them. Thus, the modules are first trained on A and then fixed, mimicking a validated model. When including data B, the modules are combined and only the encoders corresponding to the 'new' features in B are trained.

Baseline models for IIO experiments

Three baselines are proposed as depicted in blue, green and purple in Figure 2.

- **The static model** is where modules trained in A are directly tested in B, thus not considering additional IIO features.
- **The local model** is where modules are only trained on the target data set B, thus without insights from the larger source data set.
- **The global model** is the ideal, but unlikely, scenario of when all data can be shared between A and B and the modules are trained on the union of data $(A \cup B)$.

Results

For simplicity, only results for diagnostic decoders are reported. Results for a model including feature decoding and idempotence (i.e. where a specific question-answer pair

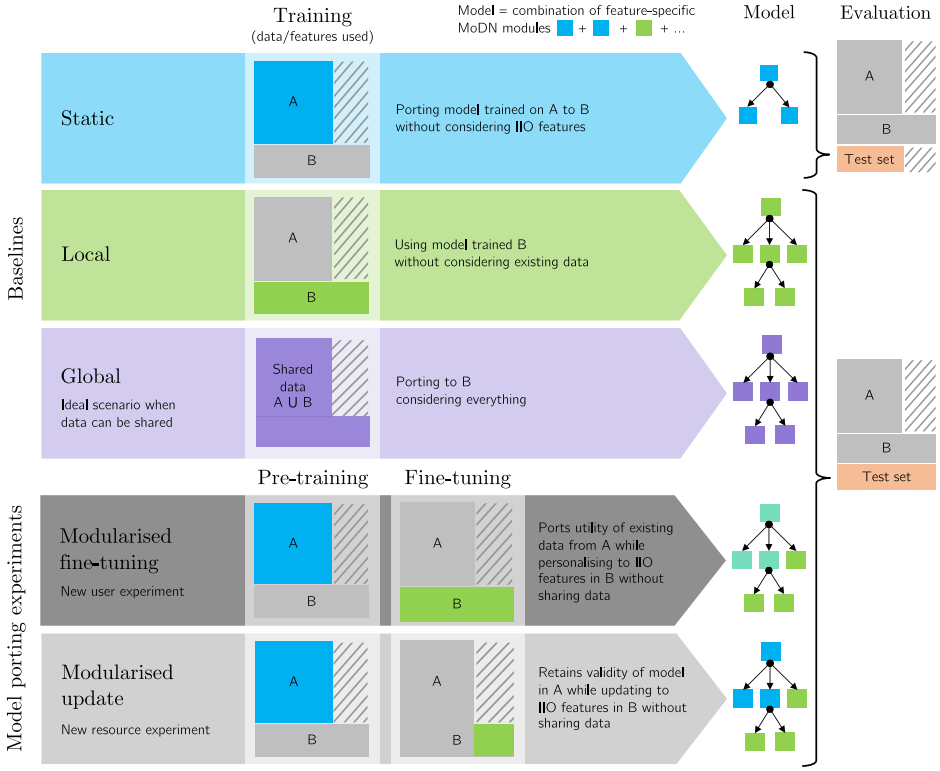


Fig 2. Experimental set up for porting MoDN modules in IIO settings. MoDN is tested in two "model porting experiments" (grey), where modules are ported from a larger *source* data set (**A**) for fine-tuning or updating on a smaller, imperfectly interoperable *target* data set (**B**). The two experiments represent either a scenario where a new user with different resources starts using a CDSS or where an existing user gains new resources and would like to merge training. Three baselines are proposed. **Static** (blue) where modules trained in **A** are directly tested in **B**, thus not considering additional IIO features. **Local** (green) where modules are only trained on the target data set **B**, thus without insights from the larger source data set. **Global** (purple) is the *ideal* but unlikely, scenario of when all data can be shared between **A** and **B** and the modules are trained on the union of data ($\mathbf{A} \cup \mathbf{B}$). The **modularised fine-tuning** experiment, pre-trains on **A** and then fine-tunes all modules (for all features) on **B** (thus personalising the modules trained on **A**). The **modularised update** experiment, pre-trains the blue modules on **A** and then adds modules specific to the new IIO features (in green) which have been independently trained on **B** (thus preserving the validity of the modules trained on **A**). The colors of the MoDN modules illustrate their training on distinct data sets and their potential re-combination in the porting experiments. In particular, the modules trained on **A** (blue) and fine-tuned on **B** (green) are thus depicted in teal.

will not change the prediction regardless of how many times it is asked) can be found in 1 and 1.

Visualizing the MoDN *state*

After encoding all available features and retrieving the resulting *states* from MoDN, we compute the t-SNE mapping of the 'vectorised patients' in order to visualise them as points on a two-dimensional plot. For visual simplicity, we limit this to the patients in

the training set who have one (or a combination) of the top 8 diagnoses, namely, pneumonia +/- anemia, fever without source (FWS) +/- anemia, upper respiratory tract infection (URTI) +/- anemia, diarrhea and 'other' (the latter of which is anticipated to have a more distributed placement on the plot).

Here, the *state* is represented as a point and clustered with similar *states*. Each mapped data point is colored according to their true diagnosis/diagnoses. Figure 3 shows several clearly homogenous clusters indicating that patients with the same diagnoses are 'close' to each other in our internal model representation (and thus that the *state* represents the outcome). Furthermore, we see smooth transitions between the clusters. For example, patients with FWS (in green) are mapped next to patients with a combination of FWS and anemia (in turquoise). Conversely, we see the *other* diagnoses to be more distributed in multiple clusters as would be expected.

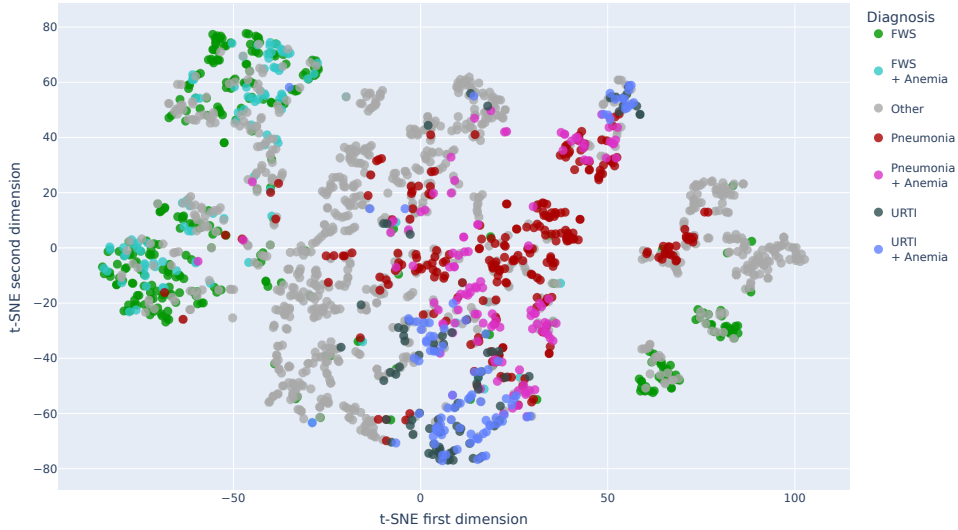


Fig 3. Two dimensional t-SNE decomposition of the *state* vector for the patients of the training set. The projection for each data point is overlaid with a color representing the true diagnosis/diagnoses of the patients.
URTI: Upper Respiratory Tract Infection, FWS: Fever Without Source

MoDN diagnosis decoding

The predictive performance of MoDN was compared to the best logistic regression and MLP algorithms for each target diagnosis. Figure 4 shows the macro F_1 scores (unweighted average of F_1 scores for the presence and absence of the disease). An overall performance is computed as the average over all diagnoses. Paired t-tests show that MoDN significantly outperforms the baselines for all binary classifications as well as for the overall diagnosis prediction. Malaria is an exception, where MoDN and baseline models have equivalent performance.

The confidence calibration plot in Figure 5 shows the predicted diagnosis probabilities by MoDN versus the correctness likelihood in the test set. For example, out of all the test points for which our model predicts a probability 0.9 of having disease d, our model is perfectly calibrated if 90% of these test points are indeed labeled as having d. In Figure 5, the points representing the confidence of our model are close to the line of perfect calibration. This shows that the predicted probabilities of MoDN are a good reflection of its confidence.

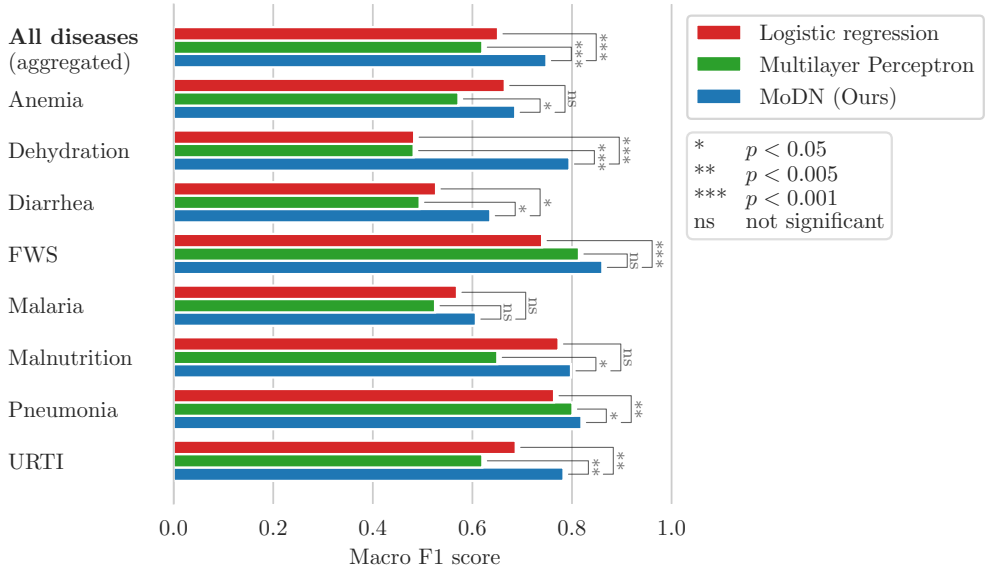


Fig 4. MoDN diagnosis decoding performance Mean of the 5×2 cross-validated macro F1 scores for the diagnosis prediction on the test sets. Furthermore, MoDN significantly beats at least one of the baselines for each of the individual diagnoses except for malaria.

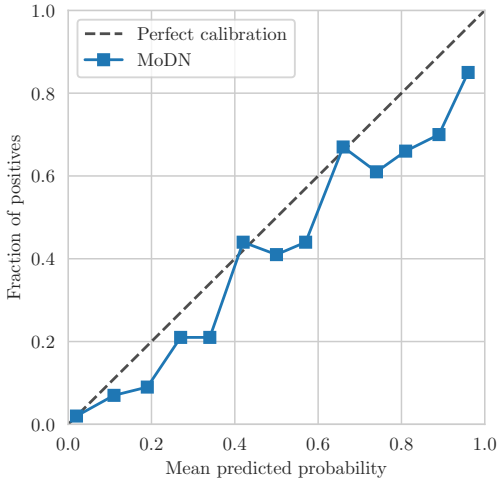


Fig 5. MoDN calibration curve of the predictions on the test set after having encoded all available features.

Visualizing diagnostic trajectories

The metrics in the previous section show the model performance on unseen data once all the available features have been encoded Figure 4. One of the assets of our model is that it provides the clinician with feedback at any point in the consultation. For a given patient, the clinician can thus see how the predictions evolve as the features get encoded. The two heatmaps in Figure 6 show the predictive evolution for two randomly selected patients from the test set. The possible diagnoses are given by the $y-$axis and the $x-$axis shows the sequentially encoded features (from left to right), along with the values for that specific patient. We see how the model shifts towards the colour poles

(blue and red extremes) as it learns more about the patient and gains confidence. The cut-off for binary classification is 50%. Thus, each percentage above 50% is increasing confidence in a positive diagnosis, while each percentage below is increasing confidence in a negative diagnosis. In each case, the model predicts the outcome correctly, but with various levels of confidence, with different patterns of evolution. For example, in Figure 6a predictive confidence accumulates slowly throughout the consultation, while in Figure 6b a confident prediction is achieved early, after a highly determinant question of "fever only". In another case (Figure S3), the model predicts correctly but with less confidence (remaining at about 0.6).

These feature-wise predictions thus give the clinician an assessment of the impact of that feature on the prediction.

The new IIO user scenario: Modularised fine-tuning

Here, we test modularised fine-tuning of MoDN in the 'new IIO user' scenario as described in Figure 2. With MoDN, we can port the relevant modules pre-trained on the larger more established *source* data set to another smaller and IIO *new* data set and fine tune them whilst adding additional modules. Figure 7 shows the performance in macro F_1 score for this proposed solution (dark grey, *Modular fine-tuning*) tested in four levels of feature overlap (from 60—100%), and compared to three baselines described in Figure 2 i.e. global (training on shared data), local (training on full feature set but only on new data), and static (training only on source data and thus with a restricted feature set).

We see that a MoDN model built with modularised fine-tuning (without sharing data) matches the performance of the global model (trained on the union of shared data) and that it maintains its performance at all levels of feature overlap tested. This is in contrast to the static model (teal), which is significantly affected by decreasing feature overlap.

The new IIO resource scenario: Modularised updating

This model isolates training updates to newly added features (in the scenario where new resources become available and are added to the decision tree). We use the trained modules from the source data set as a starting point and keep their parameters fixed. We then apply the frozen modules to the local data set and only train the modules corresponding to new features. The performance of this model is shown in light grey in Figure 7. Similarly to the modularised fine-tuning, we see that modularised update matches the global model where all data is shared. This shows that MoDN decision rules can be adapted to new features without modifying previously validated predictions of existing features, thus preserving the validity of the tool.

Discussion

With the increasing use and complexity of electronic health records, there is enormous potential for deep learning to improve and personalise predictive medicine. However, it has not yet reached wide-spread use due to fundamental limitations such as insufficient performance, poor interpretability, and the difficulty of validating a continuously evolving algorithm in prospective clinical trials. [29, 30] When deployed, there is a tendency to favor sparsely-featured linear models, probably for their inherent interpretability and as a consequence of exploiting fortuitous feature overlaps between imperfectly interoperable (IIO) data sets, which limits feature diversity. However, the

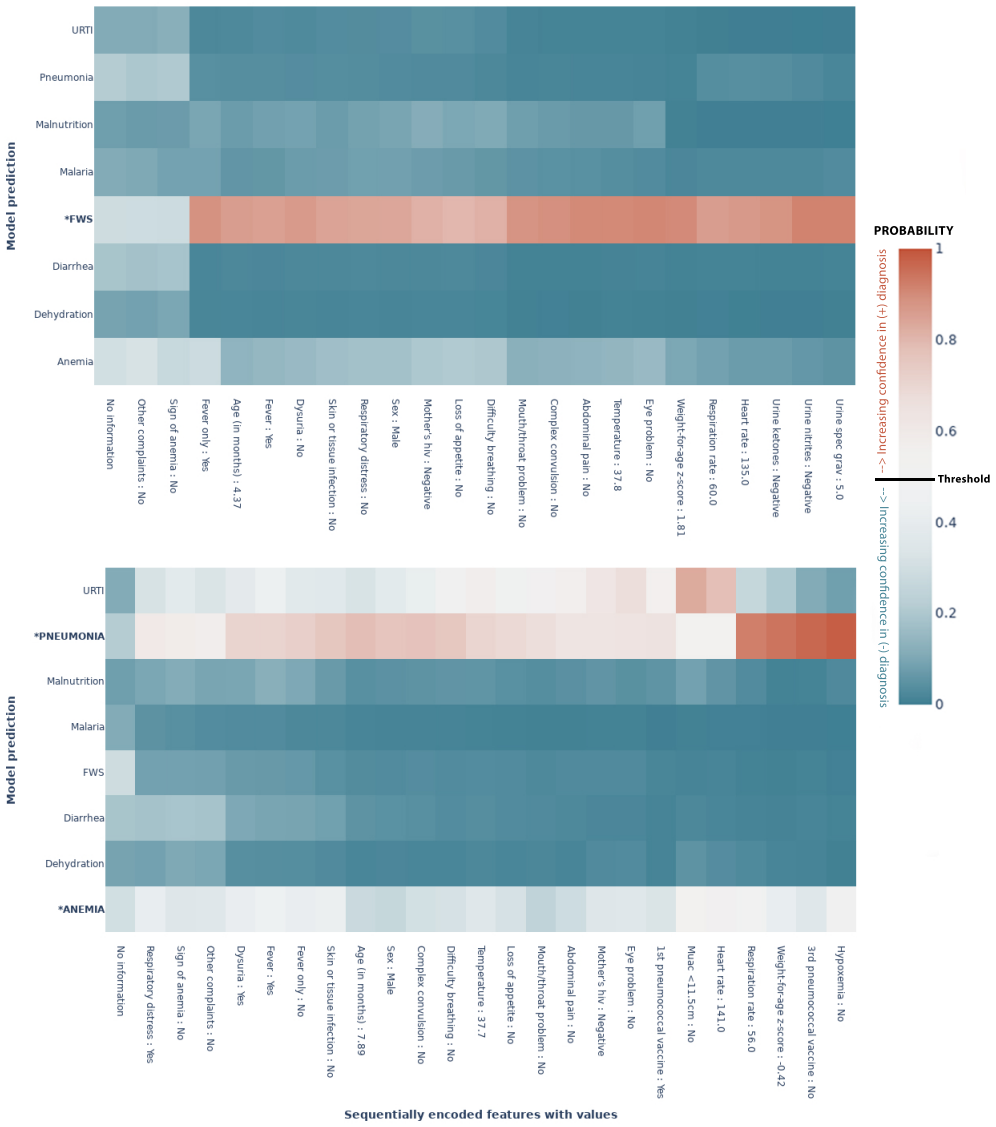


Fig 6. MoDN's feature-wise predictive evolution in two random patients. Each graph represents a single patient randomly selected from the test set. The $y-$axis lists the eight possible diagnoses predicted by our model. The true diagnosis of the patient is in bold and marked by an '*'. The $x-$axis is a sequential list of questions asked during the consultation (the response of that specific patient is also listed). In each case the model predicts the true label correctly. The heatmap represents a scale of predictive certainty from red (positive, has diagnosis) to blue (negative, does not have diagnosis), where white is uncertain. **(a)** Patient with the true diagnosis of pneumonia and anaemia. Here, predictive confidence accumulates slowly throughout the consultation. **(b)** Patient with a true diagnosis of FWS. Here, a confident prediction is achieved early after a highly determinant question of "fever only". ***:** *True diagnosis, URTI: Upper Respiratory Tract Infection, FWS: Fever Without Source, Threshold: probability at which the model categorises the patient with a diagnosis(50%)*

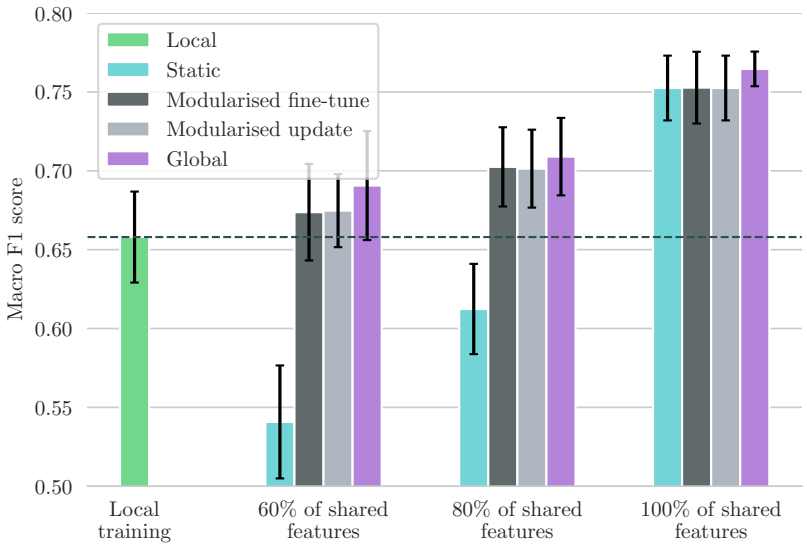


Fig 7. Comparison between the ported models and the baselines. Performance metric is the mean macro F1 scores with 95% CIs. Modularised fine-tuning or updating on additional local features (**gray**) consistently increases the model's performance compared to statically using a source model that only uses shared features (**teal**). The modularised update scenario achieves this without changing the model's behaviour on patients in the source dataset. The fine-tuning approaches perform almost as well as the global baseline (**purple**) that trains on the union of shared data. When the percentage of shared features is 80 or 100%, fine-tuning is significantly better than training only locally on the small 'target' dataset (**green**).

personalising patterns in health data are unlikely to be linear nor explained by a few features. [31]

In this work, we proposed MoDN, a novel approach to constructing decision support systems that seeks to address the issues above, allowing interpretable deep learning on imperfectly interoperable data sets.

In predicting diagnoses Figure 4 and features (supplement p 18), we see that the modularity of MoDN yields a significant performance benefit compared to its 'monolithic' counterparts, the latter of which process all features at once as opposed to an ensemble of feature-wise models. It outperforms logistic regression as well as MLP for all diagnoses excepting malaria. As malaria is strongly predicted by a single feature (i.e the rapid diagnostic test), it is anticipated that model design would have limited predictive value. Particularly useful is that the gain in performance does not come at a higher computational cost, as MoDN uses a similar number of parameters.

The architecture of our model is similar to classic modular neural networks (MNNs) described by Shukla et al. [32] However, there is little literature on the implementation of MNNs in the medical field and in the examples found, the module resolution is limited to pre-defined feature clusters such as in Pulido et al. [33] for the diagnosis of hypertension. The feature-wise modules in our proposed method means that no subgroups of features are constrained to be present together.

Visualising the *states* of MoDN in a 2-dimensional space Figure 3, we see the clustering and granularity of patient representations align with interpretable expectations of patient similarity. For instance, the catchall diagnoses of 'other' and 'fever without source' (FWS) have the most distributed spread, while more specific diagnoses have more homogeneous clusters. Interestingly for FWS, we see the states are distributed into four clear sub-populations, which we hypothesise may relate to the unknown etiology of the patient's fever. A previous effort to cluster patient profiles using unsupervised auto-encoders showed the potential of deriving a general-purpose patient representation from medical data and how it could facilitate clinical predictive modeling. [34]

MoDN sequentially ensembles feature-specific modules into a continuously 'extending' model. A key benefit of this approach is the ability to make granular interrogations of the predictive impact of each feature. Thus, MoDN aligns the learning process with the clinician, where they can visualise the diagnosis evolve during the consultation. This could act as a training tool, helping them understand the impact of their responses, which may in turn guide more careful collection of highly determinant features. To the best of our knowledge, the evolutive feature importance of MoDN is unique to the literature on CDSS. In traditional monolithic models, feature importance is computed retrospectively, using computationally expensive techniques, which may not allow the user to make corrective steps at the time of feature collection.

As stressed in many works, systematic missingness poses a major limitation to traditional deep learning on CDSS data. [21] Typical ML algorithms that operate on a vector of features are particularly affected by missingness because all data points must be encoded in some way in order to use the feature. The result is that the feature is either dropped or imputed, thus either reducing available information or injecting noise/bias. The risk of bias is particularly high when imputing systematically missing data common to CDSS. Thus, traditional models carry the risk of exploiting clinically irrelevant patterns of systematic missingness. The feature-specific modules of MoDN on the other hand, by design, cannot detect cross-feature patterns in missingness, and no imputation or feature limitations are required. Few other options exist for imputation-free neural nets. For instance, *Network Reduction* proposes a single neural net for each possible configuration of complete features [35]; an idea that has since been iterated by Krause et al. [36] and Baron et al. [37]. However, all these approaches suffer

at scale, where the number of possible configurations grows exponentially with the number of features, creating an unfeasible computational overhead in high-dimensionality data sets. By comparison, the number of NN in our approach is linear in the number of features.

As mentioned throughout this work, MoDN seeks to use modularisation to address the issues faced during collaborative algorithm updates, i.e. 1) offering users the ability to update targeted modules in vacuo, thus retaining the validity of previously validated modules, and 2) allowing models to be ported between IIO datasets. A large-scale study on 200,000 patients by Google research, [15] demonstrated an approach to deploying deep learning on IIO data from multiple centers. They developed an automated data harmonisation pipeline, and showed that they could accurately predict multiple medical events. However, such models would still be limited to the fortuitously overlapping feature sets. The results in Figure 7 show that MoDN is able to address the loss of information caused by IIO data in decentralised settings, where it matches performance of a 'global' baseline trained on the union of shared data. This portability, also makes MoDN more amenable to distributed learning.

Limitations

This work sought to validate MoDN on a real-world data set and specifically work within the consultation logic of the CDSS. This also limits its findings to the inherent diversity available in the question structure of this data set (albeit large, with over 200 unique question sets detected). A knock on effect of training MoDN on a single fixed questionnaire logic, is that we cannot guarantee the performance of the algorithm if the question order used in a consultation differs from the order in the training set.

This could be addressed by simply randomising all questions into a global question block, which would also allow MoDN to provide insights on the 'next most predictive questions to ask'.

Conclusion

Modular Clinical Decision Support Networks show the various advantages of modularising neural nets into bite-sized predictions, which not only improves predictive performance on CDSS-derived data but also allows it to interpretably integrate into the sequential logic of a medical consultation. The flexible portability of the modules also provides more granular options to building collaborative models which may address some of the most common issues of model validity as well as data ownership and privacy.

Contributions

C.T. and T.V. contributed equally to this work and are listed as co-first authors. The study comprised 2 interdisciplinary parts:
Clinical study (Data acquisition and CDSS expertise). Data collection and curation: K.K. Methodology: K.K., R.T., L.C. Supervision: A.V.K., R.T., L.C., K.K., M.A.H. Project Administration: A.V.K.
Deriving MoDN. Data curation: C.T. Formal Analysis: C.T. Conceptualization: T.V., M.A.H. Methodology and Investigation: C.T., T.V., M.A.H. Validation: C.T., T.V. Visualization: C.T., T.V., M.A.H. Supervision: M.A.H., T.V., M.J. Project Administration: M.A.H., M.J. Resources: M.J. Writing (original draft): C.T., M.A.H., T.V. Writing (final draft): C.T., M.A.H., T.V.
All authors approved the final version of the manuscript for submission and agreed to be accountable for their own contributions.

Ethics declarations

This work was supported the Botnar Foundation (grant n°6278). The funder played no role in analysis or decision to publish.

All authors declare no Competing Financial or Non-Financial Interests.

Data and code availability

Anonymized data are publicaly available here:
`https://zenodo.org/record/400380#.Yug5kuzP00Q`
The full code are available at the following GitHub repository:
`https://github.com/epfl-iglobalhealth/MoDN-TrottetVogels2022`

Supporting information

Methods

Feature selection

	number of available values	minimum value	maximum value	description
agem	3192	2.069815	59.794659	Age in months
ex	3192	0.000000	1.000000	Sex
hiv_mom	3185	0.000000	9.000000	Mother's HIV status
ttt_before	1725	0.000000	16.000000	Treatment received before D0 visit
vac_pcv1	1569	0.000000	9.000000	History: 1st pneumococcal vaccine
vac_pcv3	1159	0.000000	9.000000	History: 3rd pneumococcal vaccine
bcx_id	622	1.000000	25.000000	Blood cx ID, D0
pglucose	305	3.100000	14.100000	blood sugar (mmol/L), D0
udip_ket	687	0.000000	4.000000	Urine ketones
udip_nit	686	0.000000	2.000000	Urine nitrites, D0
udip_spec	673	1.000000	6.000000	Urine Spec Grav, D0
urine_cx_id	74	1.000000	15.000000	Urine culture: Microbe identified
urine_type	553	1.000000	3.000000	Urine: type of collection, D0, F23
convulscomplex	3191	0.000000	1.000000	Sign: Complex convulsion ¿=2/24h (d0)
pallor	3192	0.000000	1.000000	Sign: Any sign of anemia (d0)
respdistress	3192	0.000000	1.000000	Sign: respiratory distress (d0)
skin_sev	3192	0.000000	1.000000	Sign: Severe skin or soft tissue infection (d0)
hr1	3142	93.000000	214.000000	Heart rate, initial, D0
hypox	1596	0.000000	1.000000	Vital Sign: Hypoxemia, D0
muaclow	2811	0.000000	1.000000	Vital Sign: MUAC ¡11.5cm & age¿6 months
rr1	3180	20.000000	90.000000	Initial RR entered, D0
temp	3186	37.500000	42.000000	Axillary temperature, D0
waz	3188	-6.950000	5.310000	Weight-for-age z-score
complaint	3192	0.000000	8.000000	NaN
eye	3191	0.000000	1.000000	Symptom: Any eye problem (d0)
abdopain	3191	0.000000	1.000000	Chief Complaint Abdominal Pain, D0
dyspnea	3191	0.000000	1.000000	Chief Complaint Difficulty Breathing, D0
dysuria	3191	0.000000	1.000000	Chief Complaint Dusuria, D0
fev	3192	0.000000	1.000000	Chief Complaint Fever, D0
feveronly	3192	0.000000	1.000000	Chief Complaint Fever Only, D0
loa	3191	0.000000	1.000000	Chief Complaint Loss of Appetite, D0
pharyngitis	3191	0.000000	1.000000	Chief Complaint Mouth/Throat problem, D0
uri	3192	0.000000	1.000000	Chief Complaint URI, DO

Tab S 1. Summary statistics of the features of the e-POCT data set.

Model architecture and implementation details

The different modules of MoDN are multilayer perceptrons (MLP). MLPs are fully connected feedforward neural networks. The weights and biases of each neuron are optimized via the backpropagation algorithm.

We used *ReLU* activations and one or two hidden layers for each module. The overall architecture thus remained quite simple. For each training data point, the state $\mathbf{S}$ is initialized as a random `PyTorch` [38] tensor of size s. The initial state is also optimized throughout the training process. The input layers are of size $1 + s$, s and s respectively for the encoders, feature decoders and disease decoders. The output layer is of size s for the encoders and 2 for the continuous feature decoders (predicting the mean and standard deviation). The disease decoders and the categorical feature decoders also both have an output of size 2, with a *softmax* activation applied to the output layer to compute the probabilities of the two classes.

Feature decoding

We present here an extended version of the MoDN, including some additional modules to perform 'feature decoding'. This allows the clinician to retrieve the values for previously encoded features or to get predictions for unavailable features. For each feature, we defined a *feature decoder*. Depending on the nature of the feature, they are either *continuous* or *binary*. The feature decoders are applied to the state $\mathbf{S}$ to predict the value of the feature. If feature j is continuous, $\mathbf{F}_j$, $\mathbf{F}_j : \mathbb{R}^s \longrightarrow (\mathbb{R}, \mathbb{R}^+)$. It predicts the mean and standard deviation of feature j. If feature j is binary, we have $\mathbf{F}_j : \mathbb{R}^s \longrightarrow \{0, 1\}$.

In the optimization process of our model we included an auxiliary loss function to train our model to perform feature decoding. The feature decoding loss is made of two components. A 'known' part, corresponding to the features that have already been encoded, and an 'unknown' part, for the features from a later stage in the tree. The 'known' part ensures that the model retains past information, and the 'unknown' infers correlations between encoded features and later features in the tree.

The continuous features are optimized using the *negative log-likelihood* loss and the binary features with the *cross-entropy* loss.

Let $\mathbf{z}_p^t = [(q_p^0, a_p^0), (q_p^1, a_p^1), \ldots, (q_p^t, a_p^t)]$ be the ordered list of (question, answer) pairs for patient p as defined in Equation 4. $c_\theta(\mathbf{z}_p^t, j)$ is the prediction of the model for patient p and feature j given the patient information up to t and a_p^j is the true value of feature j. For the "known" part of the loss, we sum the predicted feature values for features known to the model.

$$feature\_loss\_known = \sum_{p=1}^{N} \sum_{t=1}^{T} \sum_{j<=t} \ell(c_\theta(\mathbf{z}_p^t, j), a_p^j) \tag{6}$$

where ℓ is the negative log-likelihood loss or the cross-entropy loss, depending on the nature of the feature. We sum over all the N patients in the dataset and their corresponding ordered lists $\mathbf{z}_p^t$. Similarly, the unknown part of the model is given by

$$feature\_loss\_unknown = \sum_{p=1}^{N} \sum_{t=1}^{T} \sum_{j>t} \ell(c_\theta(\mathbf{z}_p^t, j), a_p^j), \tag{7}$$

where we sum over the predicted information that has not yet been provided to the model.

As explained in section 1 during each SGD step, once all the features in a level of the consultation tree have been encoded, the disease decoders are applied. At this stage, we also apply all the feature decoders and compute the *feature_loss_known* and *feature_loss_unknown*.

Idempotence

We trained the MoDN to be **idempotent**. An operator is idempotent if it has the same effect whether it is applied once or several times. In our setting, it means that the information vector for a patient should not change even if the same feature is encoded twice or more. To enforce that constraint, an additional loss, the *idempotence_loss* was added to the global loss minimized during the optimization. For each feature, we computed the mean squared error between the state after having encoded all the features once and the state after re-encoding the given feature. Let F be the number of different features in the model, $\mathbf{S_p}^b$ the state for patient p once all the features have been encoded once, and $\mathbf{S}_p^f$ the state after re-encoding feature f. Then, the loss is given by

$$idempotence\_loss = \sum_{p=1}^{N} \sum_{f=1}^{F} \left(\mathbf{S}_p^b - \mathbf{S}_p^f\right)^2. \tag{8}$$

Results

Calibration curves

The calibration curve in Figure 5 was computed using the observations of the test set. The probability space was split into 13 equally space bins. The $x-$axis shows the mean final predicted probability by the MoDN for the observations in each bin. The $y-$axis shows the proportion of positive diagnoses for the observations in each bin. For a given disease d, let $\hat{P}(d) = p$ be the estimated probability by the MoDN of having d. Then the $y-$axis is an estimate of $P\left(d \mid \hat{P} = p\right)$, the true probability of having d, knowing that the MoDN predicted p.

MoDN with feature decoding

We present here the predictive performance results for MoDN when trained additionally to perform feature decoding. As for the main model, 1 shows that MoDN outperforms the baselines significantly for the overall disease prediction. Furthermore, it outperforms the performance of at least one of the baselines for each of the individual diseases, except for pneumonia. The calibration curve in 2 shows that the model with feature decoding is calibrated close to the perfect calibration line. These results indicate that with accurate tuning, the model maintains a competitive predictive performance even with the increased complexity due to the additional feature decoding loss function.

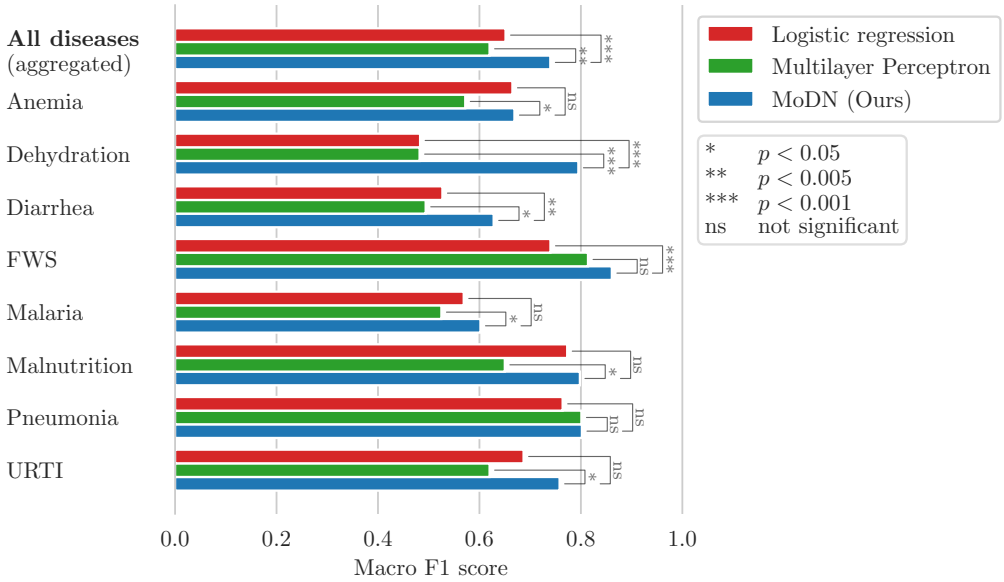


Fig S 1. Macro F1 scores for the disease prediction on test set. The baselines of MLP and logistic regression with L_2 penalty were tuned to achieve maximal performance.

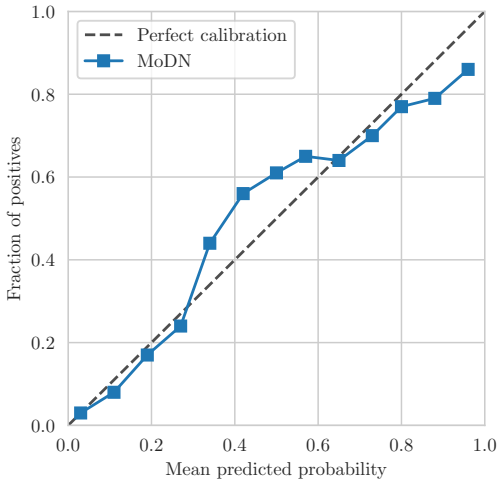


Fig S 2. MoDN calibration curve of the predictions on the test set after having encoded all available features.

Feature-wise predictive evolution in a third patient

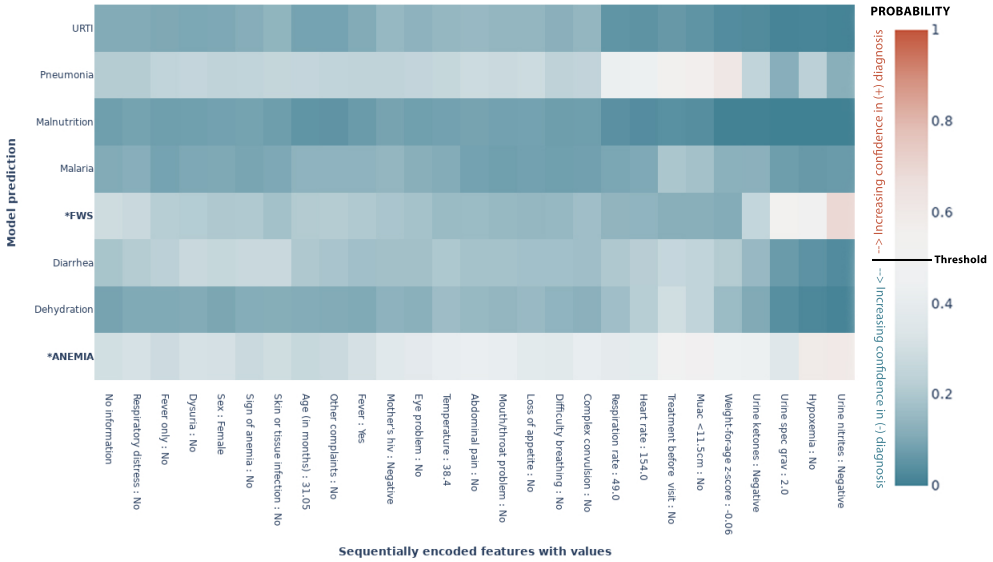


Fig S 3. MoDN's feature-wise predictive evolution in a random patient. This graph represents a single patient randomly selected from the test set. The $y-$axis lists the eight possible diagnoses predicted by our model. The true diagnosis of the patient is in bold and marked by an '*'. The $x-$axis is a sequential list of questions asked during the consultation (the response of that specific patient is also listed). In each case the model predicts the true label correctly. The heatmap represents a scale of predictive certainty from red (positive, has diagnosis) to blue (negative, does not have diagnosis), where white is uncertain. This patient has a true diagnosis of FWS and anemia. The model predicts these correctly but with less confidence, as can be interpreted from lighter colours

**: True diagnosis, URTI: Upper Respiratory Tract Infection, FWS: Fever Without Source*

Acknowledgements

The authors thank the patients and caregivers who made the study possible, as well as the clinicians who collected the data on which MoDN was validated. [25]

References

1. Sutton RT, Pincock D, Baumgart DC, Sadowski DC, Fedorak RN, Kroeker KI. An overview of clinical decision support systems: benefits, risks, and strategies for success. NPJ digital medicine. 2020;3(1). doi:10.1038/S41746-020-0221-Y.

2. Durack DT. The Weight of Medical Knowledge. http://dxdoiorg/101056/NEJM197804062981405. 2010;298(14):773–775. doi:10.1056/NEJM197804062981405.

3. Osheroff J, Teich JM, Levick D, Saldana L, Velasco F, Sittig DF, et al. Improving Outcomes with Clinical Decision Support: An Implementer's Guide. 2nd ed. Chicago: CRC Press; 2012.

4. Middleton B, Sittig DF, Wright A. Clinical Decision Support: a 25 Year Retrospective and a 25 Year Vision. Yearbook of medical informatics. 2016;Suppl 1(Suppl 1):S103–S116. doi:10.15265/IYS-2016-S034.

5. Sim I, Gorman P, Greenes RA, Haynes RB, Kaplan B, Lehmann H, et al. Clinical decision support systems for the practice of evidence-based medicine. Journal of the American Medical Informatics Association : JAMIA. 2001;8(6):527–534. doi:10.1136/JAMIA.2001.0080527.

6. Roukema J, Steyerberg EW, Van Der Lei J, Moll HA. Randomized Trial of a Clinical Decision Support System: Impact on the Management of Children with Fever without Apparent Source. J Am Med Inform Assoc. 2008;15:107–113. doi:10.1197/jamia.M2164.

7. Ant M, Ferreira O, Nogueira R, Santos D, Tygesen H, Eriksson H, et al. Clinical decision support system (CDSS)-effects on care quality. Int J Health Care Qual Assurance. 2014;doi:10.1108/IJHCQA-01-2014-0010.

8. Keitel K, D'Acremont V. Electronic clinical decision algorithms for the integrated primary care management of febrile children in low-resource settings: review of existing tools; 2018. Available from: `https://pubmed.ncbi.nlm.nih.gov/29684634/`.

9. Fanaroff AC, Califf RM, Windecker S, Smith SC, Lopes RD. Levels of Evidence Supporting American College of Cardiology/American Heart Association and European Society of Cardiology Guidelines, 2008-2018. JAMA. 2019;321(11):1069–1080. doi:10.1001/JAMA.2019.1122.

10. Vasey B, Ursprung S, Beddoe B, Taylor EH, Marlow N, Bilbro N, et al. Association of Clinician Diagnostic Performance With Machine Learning–Based Decision Support Systems: A Systematic Review. JAMA Network Open. 2021;4(3):e211276–e211276. doi:10.1001/JAMANETWORKOPEN.2021.1276.

11. Goldstein BA, Navar AM, Pencina MJ, Ioannidis JPA. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. Journal of the American Medical Informatics Association : JAMIA. 2017;24(1):198–208. doi:10.1093/JAMIA/OCW042.

12. Yu KH, Beam AL, Kohane IS. Artificial intelligence in healthcare. Nature biomedical engineering. 2018;2(10):719–731. doi:10.1038/S41551-018-0305-Z.

13. Deo RC. Machine learning in medicine. Circulation. 2015;132(20):1920–1930. doi:10.1161/CIRCULATIONAHA.115.001593.

14. Kohane IS. Ten things we have to do to achieve precision medicine. Science. 2015;349(6243):37–38. doi:10.1126/SCIENCE.AAB1328.

15. Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. npj Digital Medicine 2018 1:1. 2018;1(1):1–10. doi:10.1038/s41746-018-0029-1.

16. Mehl G, Tunçalp Ö, Ratanaprayul N, Tamrat T, Barreix M, Lowrance D, et al. WHO SMART guidelines: optimising country-level use of guideline recommendations in the digital age. The Lancet Digital Health. 2021;3(4):e213–e216. doi:10.1016/S2589-7500(21)00038-8.

17. Coiera E. The fate of medicine in the time of AI. The Lancet. 2018;392(10162):2331–2332. doi:10.1016/S0140-6736(18)31925-1.

18. NORC at the University of Chicago. Understanding the Impact of Health IT in Underserved Communities and Those with Health Disparities — NORC.org; 2010. Available from: `https://www.norc.org/Research/Projects/Pages/understanding-the-impact-of-health-it-in-underserved-communities-and-thos aspx`.

19. Mitchell J, Probst J, Brock-Martin A, Bennett K, Glover S, Hardin J. Association between clinical decision support system use and rural quality disparities in the treatment of pneumonia. The Journal of rural health : official journal of the American Rural Health Association and the National Rural Health Care Association. 2014;30(2):186–195. doi:10.1111/JRH.12043.

20. Ohno-Machado L. Data and the clinical decision support loop. Journal of the American Medical Informatics Association. 2016;23(e1):e1–e1. doi:10.1093/jamia/ocw060.

21. Beaulieu-Jones BK, Lavage DR, Snyder JW, Moore JH, Pendergrass SA, Bauer CR. Characterizing and Managing Missing Structured Data in Electronic Health Records: Data Analysis. JMIR Medical Informatics. 2018;6(1). doi:10.2196/MEDINFORM.8960.

22. Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A. A Survey on Bias and Fairness in Machine Learning. ACM Computing Surveys. 2019;54(6). doi:10.1145/3457607.

23. Aristidou A, Jena R, Topol EJ. Bridging the chasm between AI and clinical implementation. The Lancet. 2022;399(10325):620. doi:10.1016/S0140-6736(22)00235-5.

24. Chen JH, Alagappan M, Goldstein MK, Asch SM, Altman RB. Decaying Relevance of Clinical Data Towards Future Decisions in Data-Driven Inpatient Clinical Order Sets. International journal of medical informatics. 2017;102:71. doi:10.1016/J.IJMEDINF.2017.03.006.

25. Keitel K, Samaka J, Masimba J, Temba H, Said Z, Kagoro F, et al. Safety and Efficacy of C-reactive Protein-guided Antibiotic Use to Treat Acute Respiratory Infections in Tanzanian Children: A Planned Subgroup Analysis of a Randomized Controlled Noninferiority Trial Evaluating a Novel Electronic Clinical Decision Algorithm (ePOCT). Clin Infect Dis. 2019;doi:10.1093/cid/ciz080.

26. Dietterich TG. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation. 1998;10(7):1895–1923. doi:10.1162/089976698300017197.

27. Kingma DP, Lei Ba J. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. ICLR. 2015;.

28. van der Maaten L, Hinton G. Visualizing Data using t-SNE. Journal of Machine Learning Research. 2008;9(86):2579–2605.

29. Fröhlich H, Balling R, Beerenwinkel N, Kohlbacher O, Kumar S, Lengauer T, et al. From hype to reality: Data science enabling personalized medicine. BMC Medicine. 2018;16(1):1–15. doi:10.1186/S12916-018-1122-7/FIGURES/5.

30. Hulsen T, Jamuar SS, Moody AR, Karnes JH, Varga O, Hedensted S, et al. From big data to precision medicine. Frontiers in Medicine. 2019;6(MAR):34. doi:10.3389/FMED.2019.00034/BIBTEX.

31. Prosperi M, Min JS, Bian J, Modave F. Big data hurdles in precision medicine and precision public health. BMC Medical Informatics and Decision Making. 2018;18(1):1–15. doi:10.1186/S12911-018-0719-2/FIGURES/7.

32. Shukla A, Tiwari R, Kala R. Modular Neural Networks. In: Towards Hybrid and Adaptive Computing: A Perspective. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 307–335. Available from: https://doi.org/10.1007/978-3-642-14344-1_14.

33. Pulido M, Melin P, Prado-Arechiga G. Blood Pressure Classification Using the Method of the Modular Neural Networks. International Journal of Hypertension. 2019;doi:10.1155/2019/7320365.

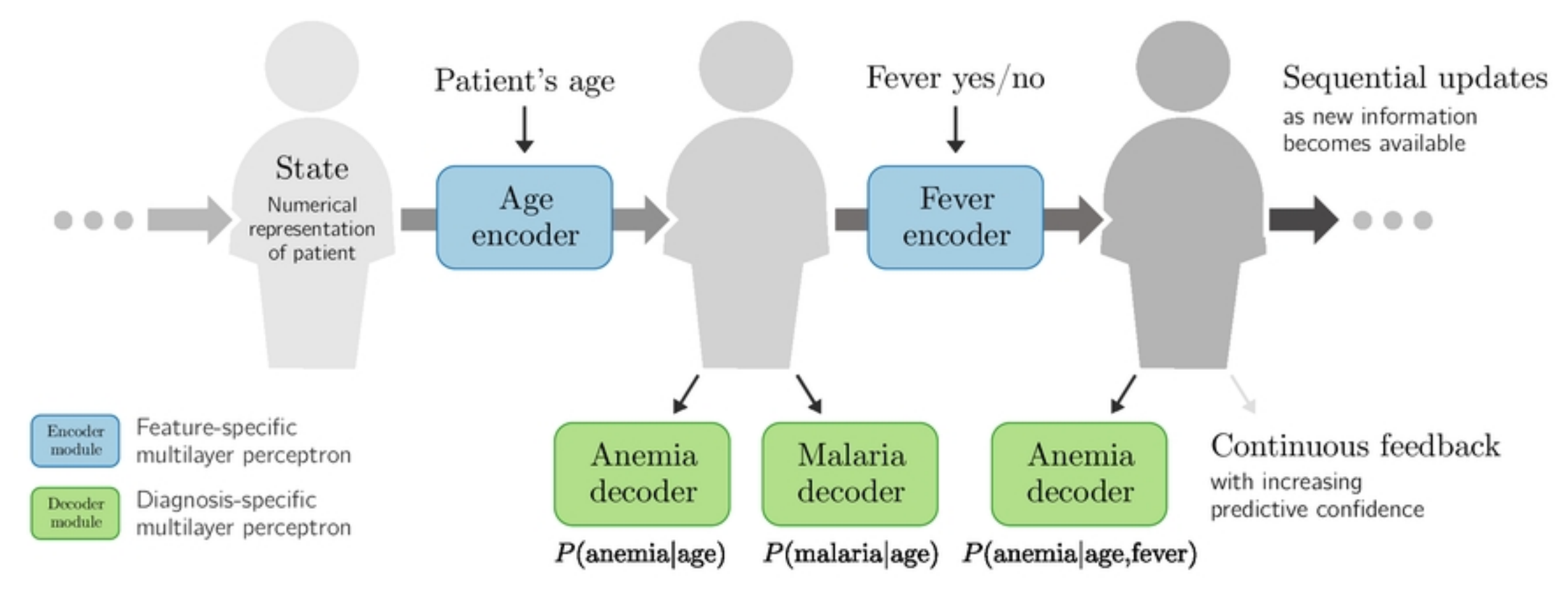
34. Miotto R, Li L, Kidd BA, Dudley JT. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Scientific Reports 2016 6:1. 2016;6(1):1–10. doi:10.1038/srep26094.

35. Sharpe PK, Solly RJ. Neural Computing & Applications Dealing with Missing Values in Neural Network-Based Diagnostic Systems; 1995.

36. Krause S, Polikar R. An ensemble of classifiers approach for the missing feature problem. In: Proceedings of the International Joint Conference on Neural Networks, 2003.. vol. 1; 2003. p. 553–558.

37. Baron JM, Paranjape K, Love T, Sharma V, Heaney D, Prime M. Development of a "meta-model" to address missing data, predict patient-specific cancer survival and provide a foundation for clinical decision support. Journal of the American Medical Informatics Association : JAMIA. 2021;28(3):605–615. doi:10.1093/JAMIA/OCAA254.

38. Paszke A, Gross S, Massa F, Lerer A, Bradbury Google J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems (NeurIPS 2019). 2019;.


Patient's age
Fever yes/no
Sequential updates
as new information becomes available
State
Numerical representation of patient
Age encoder
Fever encoder
Encoder module
Feature-specific multilayer perceptron
Decoder module
Diagnosis-specific multilayer perceptron
Anemia decoder
Malaria decoder
Anemia decoder
Continuous feedback
with increasing predictive confidence
P(anemia|age)
P(malaria|age)
P(anemia|age,fever)

Training
(data/features used)

Model = combination of feature-specific MoDN modules + + + ...

Model

Evaluation

Baselines

Static

A

B

Porting model trained on A to B
without considering IIO features

Local

A

B

Using model trained B
without considering existing data

Global

Ideal scenario when data can be shared

Shared data A U B

Porting to B
considering everything

A

B

Test set

Model porting experiments

Pre-training

Fine-tuning

Modularised fine-tuning

New user experiment

A

B

A

B

Ports utility of existing data from A while personalising to IIO features in B without sharing data

Modularised update

New resource experiment

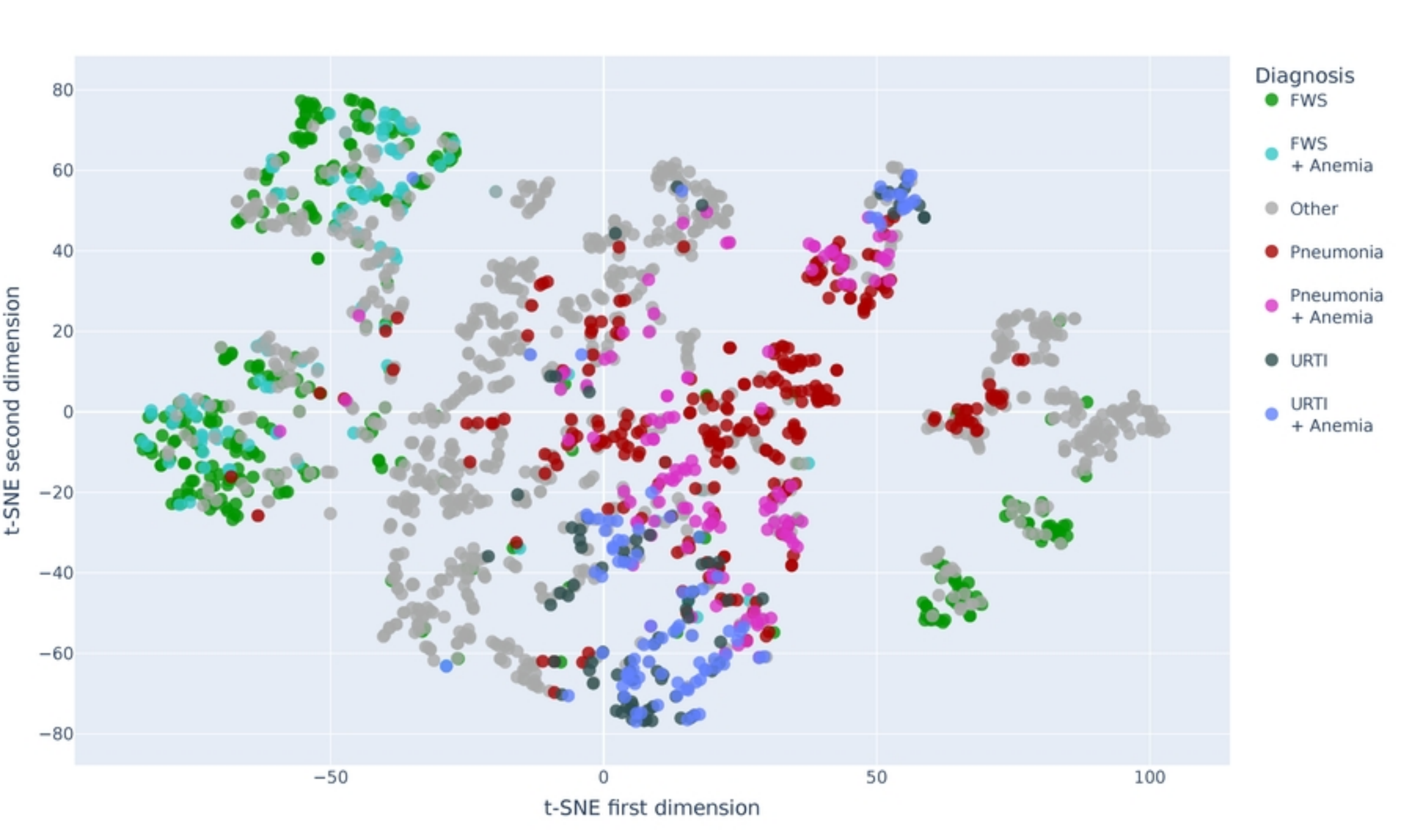
A

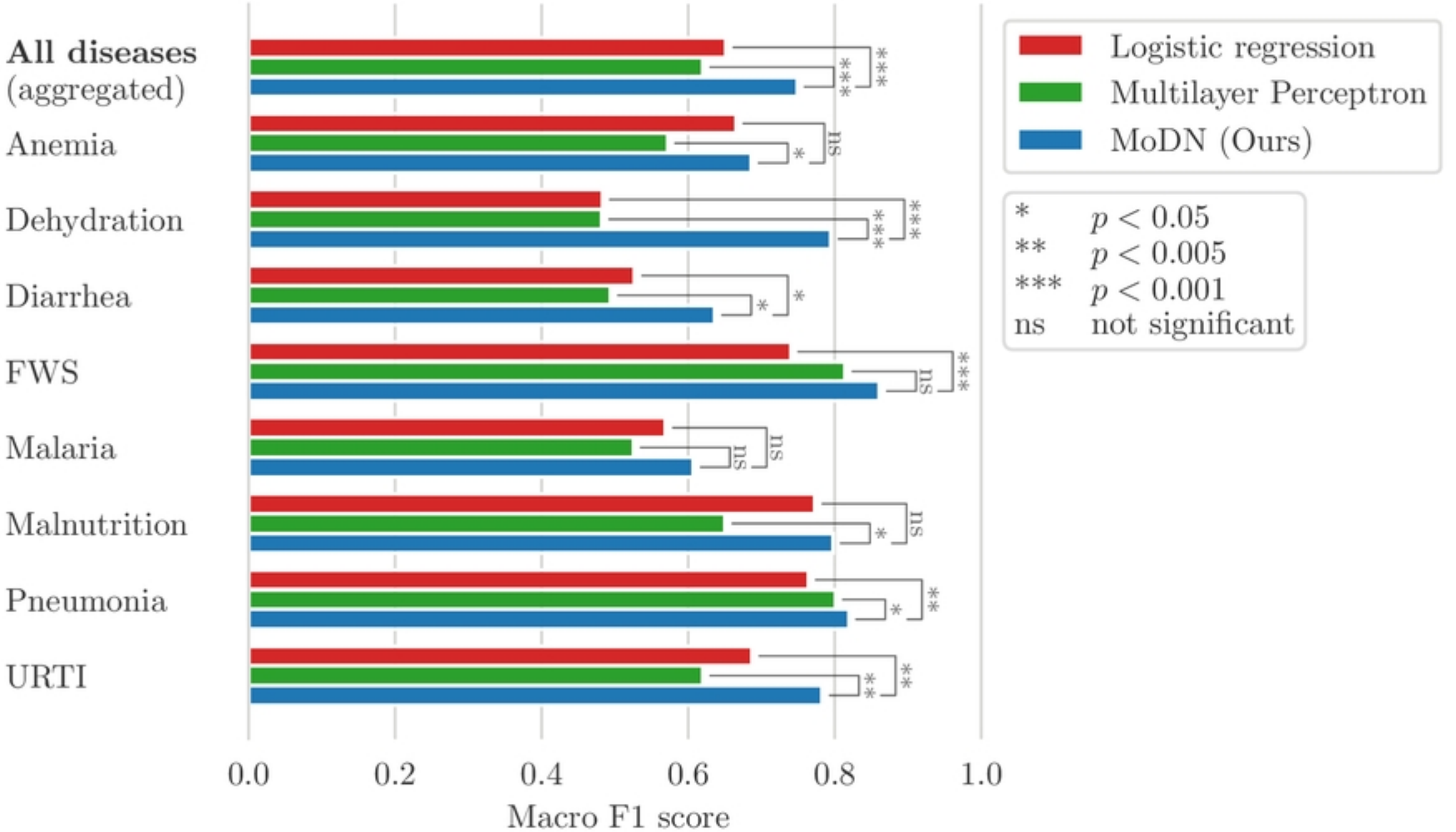
B

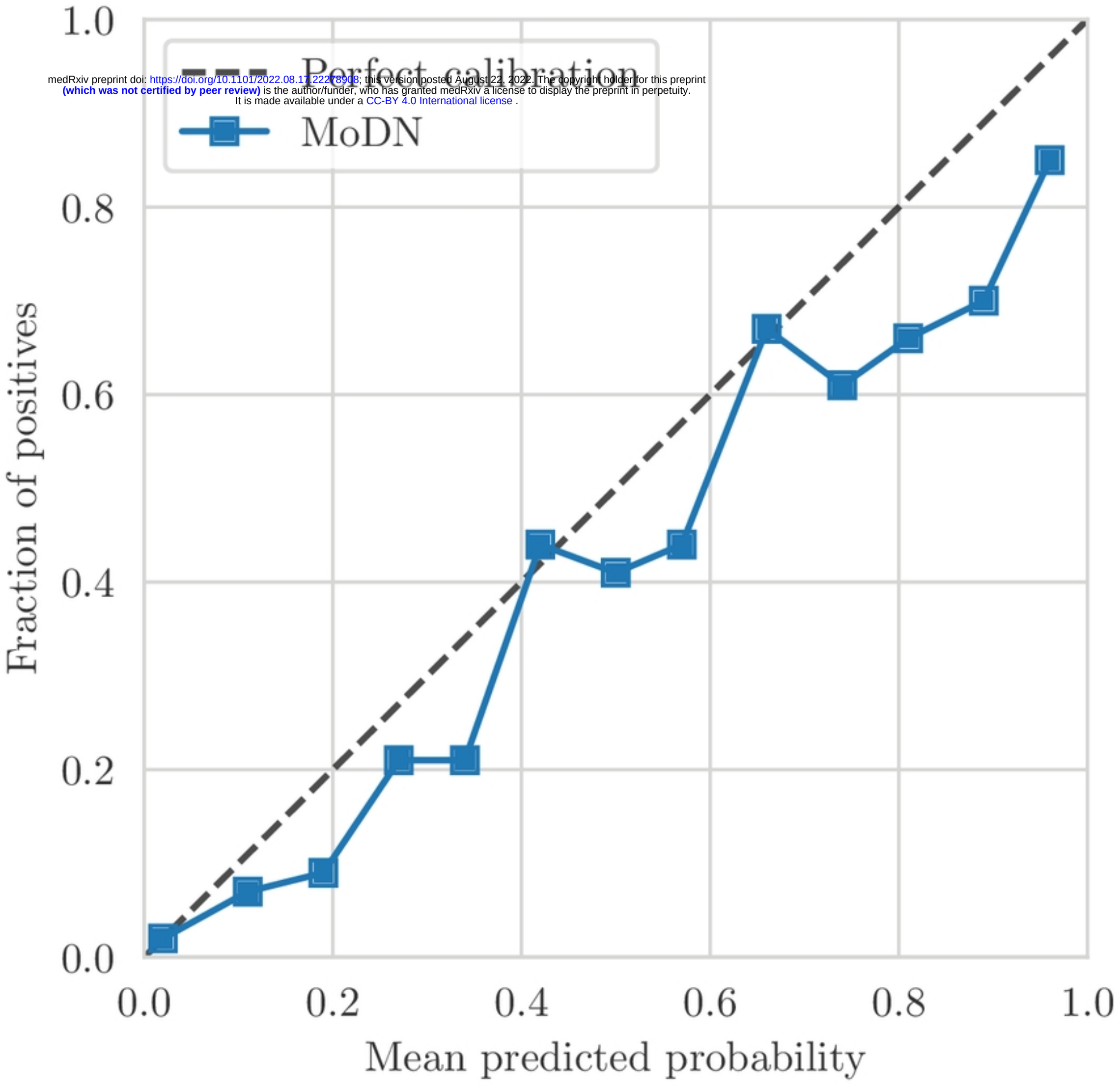
A

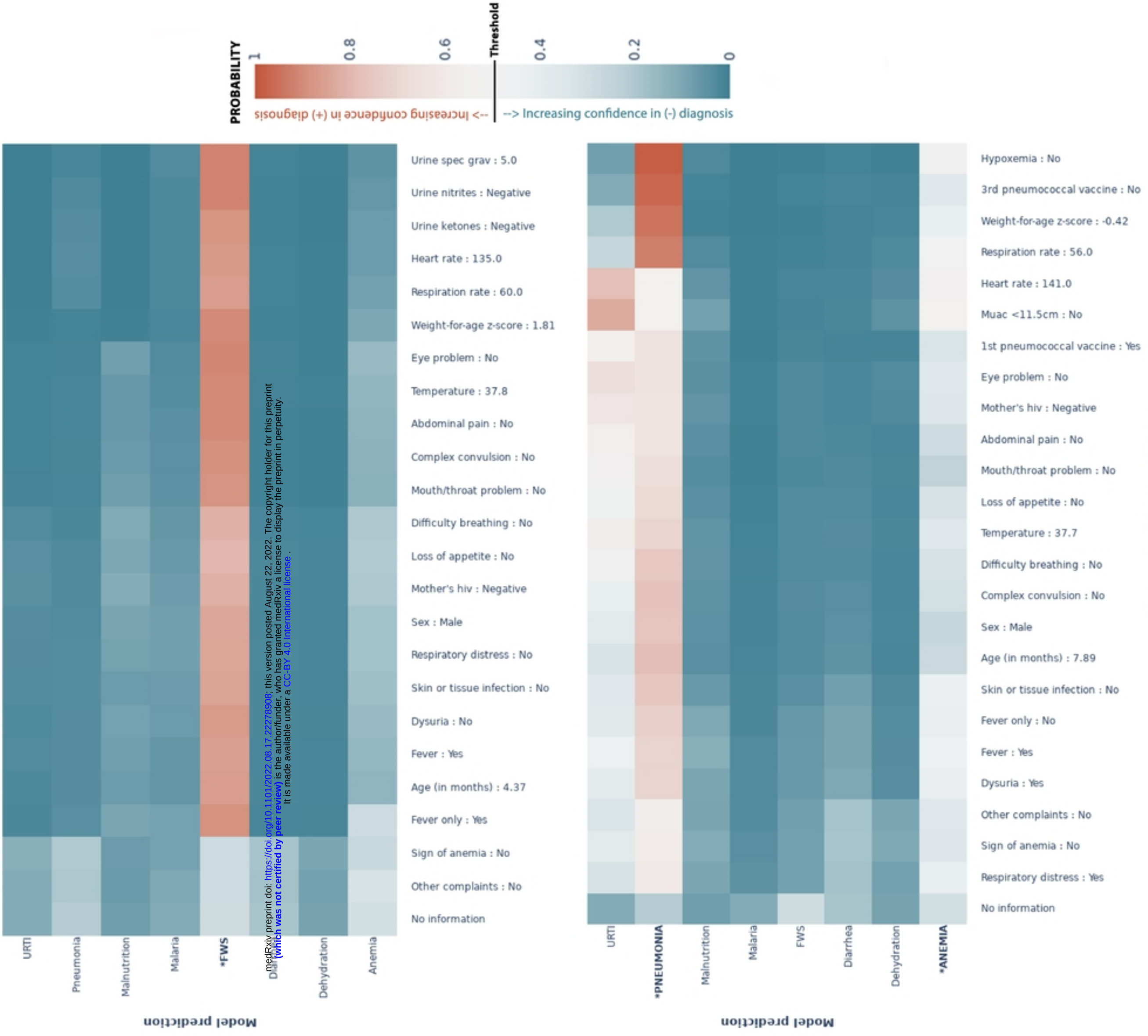
B

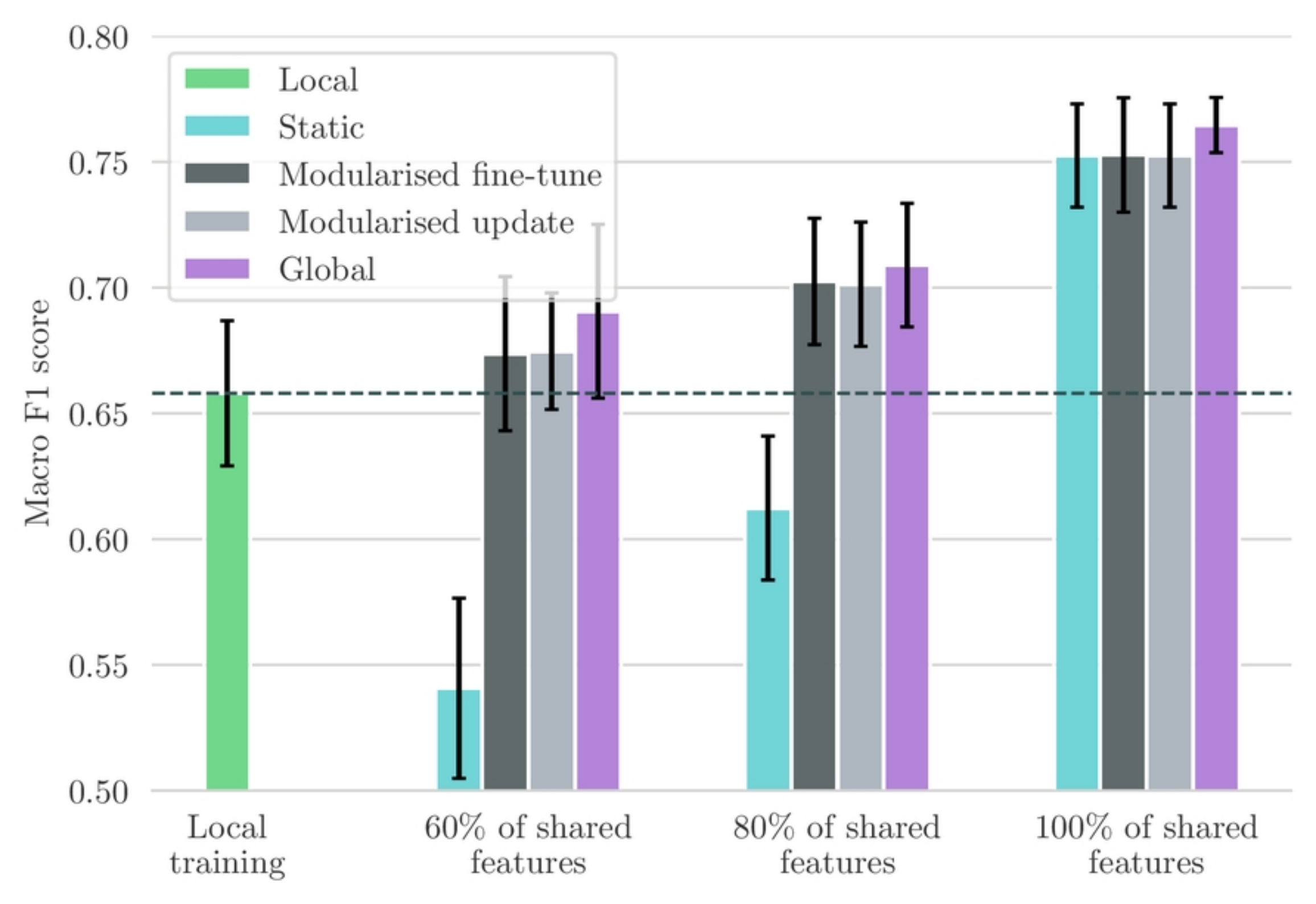
Retains validity of model in A while updating to IIO features in B without sharing data


Diagnosis
FWS
FWS + Anemia
Other
Pneumonia
Pneumonia + Anemia
URTI
URTI + Anemia
t-SNE second dimension
t-SNE first dimension
80
60
40
20
0
−20
−40
−60
−80
−50
0
50
100


All diseases
(aggregated)
Anemia
Dehydration
Diarrhea
FWS
Malaria
Malnutrition
Pneumonia
URTI
Logistic regression
Multilayer Perceptron
MoDN (Ours)
* $p < 0.05$
** $p < 0.005$
*** $p < 0.001$
ns not significant
0.0
0.2
0.4
0.6
0.8
1.0
Macro F1 score






0.80
0.75
0.70
0.65
0.60
0.55
0.50
Macro F1 score
Local
Static
Modularised fine-tune
Modularised update
Global
Local training
60% of shared features
80% of shared features
100% of shared features