ShifaMind: Concept-Mediated Automated ICD Coding

Mohammed Sameer Syed
mohammedsameersyed1@gmail.com

Xuan Lu
luxuan@arizona.edu

Abstract

Automated ICD coders are accurate but not auditable: a coder deciding whether to accept a predicted code sees attention over text, which says where the model looked, not what it used. Concept bottleneck models are the auditable form but have not reached clinical coding because scalar concepts lose capacity, and concept embeddings leak content their names do not describe. We present ShifaMind, an ICD coder with a multiplicative concept bottleneck. Each concept mentioned in a note gets a slot scaled by one scalar gate that is both the displayed score and the forward-pass multiplier, so closing it removes the concept exactly, and a factored label attention head keeps the decomposition exact over 7,940 codes. On MIMIC-IV ICD-10 full coding, with thresholds fixed on validation and ICD-9-coded notes added to training, it reaches 60.6 micro-F1 and 26.8 macro-F1, above every reproduced baseline on both F1 metrics and within 0.7 micro-F1 of the strongest reported system, a three-model ensemble. Causal suppression and slot transplantation, together with seven standard interpretability metrics, quantify both what the displayed concepts carry and what they hide, which turns the trade between accuracy and auditability into a measured quantity.

1 Introduction

Assigning International Classification of Diseases (ICD) codes to a discharge summary is an accountable act: a human coder names the finding that justifies each code and can be overruled on that evidence. Automated coders have become accurate (Edin et al., 2023; Wu et al., 2025) but do not produce evidence of that kind: per-code attention (Mullenbach et al., 2018; Vu et al., 2020; Huang et al., 2022) indicates text a model attended to, whether such weights explain the prediction is contested (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019), and a quantity that is not a term of the computation cannot be audited (Rudin, 2019).

Concept bottleneck models (CBMs) are the auditable form: predict human-named concepts, then predict the label from those concepts alone (Koh et al., 2020). They have not reached clinical coding, where two known weaknesses of the form bite hardest. A label head that reads only concept scalars loses capacity on a long-tailed task with thousands of codes. Widening each concept into an embedding recovers accuracy (Espinosa Zarlenga et al., 2022), but the embedding then carries content the concept name does not describe, the leakage problem of soft concept layers (Mahinpei et al., 2021; Havasi et al., 2022); and once the prediction depends on content the reader cannot see, intervening on the displayed concepts may not control all predictive information.

ShifaMind puts one structural rule in place: a concept reaches the output only as its slot vector multiplied by its own scalar gate, and that gate is the number displayed. At slot width zero the model is a strict CBM; at width $r > 0$ (the rank of an adapter, Section 3.3) it is a concept embedding model whose embedding is multiplied away by the same displayed scalar, so suppression and attribution stay exact at every width. The width is a dial between the two prior designs, and the paper measures what each setting buys in accuracy and what it hides from the displayed score.

Four properties hold by construction (Section 3). **Tied**: the score displayed for a concept is the multiplier on its contribution. **Exact**: the displayed contributions sum to the predicted logit. **Complete**: no pathway reaches a logit except through a named concept. **Grounded**: every active concept carries the span of the note that caused it to be routed and the assertion status of that mention. Appendix D gives a schematic of concept mediation.

Contributions.

- A multiplicative concept bottleneck whose scalar gate is simultaneously the clinician-facing score and the multiplier on an entire concept slot, with a factored label attention head that makes the exact per-code, per-concept decomposition affordable at 7,940 codes and 384 routed concepts.
- To our knowledge the first full-code ICD coder at benchmark accuracy in which every logit is an exact sum of contributions from named clinical concepts: micro-F1, macro-F1 and precision at 8 and 15 above every system reproduced by Edin et al. (2023).
- Training across coding systems: ICD-9-coded notes improve ICD-10 coding, with the encoder and the concept layer shared and only the label rows specific to each system.
- An interpretability evaluation of the coder itself, with standard metrics computed from the gates and contributions that produce the prediction: concept accuracy and intervention accuracy (Koh et al., 2020), comprehensiveness, sufficiency and plausibility (DeYoung et al., 2020), explanation size and seed stability, together with two causal tests, suppression of the supporting concepts against a random-set control and transplantation of slot contents with the gates held fixed.
- A width dial from strict concept bottleneck to concept embedding model, with the accuracy and the share of the decision the displayed gate accounts for measured at each setting.

2 Related Work

Automated ICD coding. Per-code attention over a text encoder is the dominant design: CAML (Mullenbach et al., 2018), MultiResCNN (Li and Yu, 2020), LAAT (Vu et al., 2020), PLM-ICD (Huang et al., 2022) over a pretrained clinical transformer, and MSMN (Yuan et al., 2022) with code synonyms. Edin et al. (2023) reimplemented this family under one protocol, corrected the macro-F1 computation of earlier work, and released patient-disjoint splits of MIMIC-IV; their reproductions are the baselines we quote. GoMICD (Wu et al., 2025) is the strongest reported system on that benchmark and combines three text-segmentation schemes under a model-level mixture of experts, with training-time note augmentation. Liu and Nguyen (2026) showed that training on ICD-9-coded notes helps an ICD-10 coder; Section 5.2 shows the same with a shared encoder and concept layer. None of these systems constrains where predictive signal originates: their explanations are attention weights over contextual token states, and nothing ties a prediction to evidence a reader can name.

Evidence for codes. MDACE (Cheng et al., 2023) provides coder-annotated evidence spans for part of MIMIC-III and scores attention-derived explanations against them; Edin et al. (2024) improve such explanations without span supervision. Both explain a prediction after the fact; ShifaMind makes the explanation and the prediction the same object.

Concept bottlenecks. Koh et al. (2020) formalised the concept-then-label decomposition on two image tasks with 10 and 112 concepts and a single-label target. Mahinpei et al. (2021) showed that soft concept activations encode information beyond the concepts, Havasi et al. (2022) propose remedies on the concept-prediction side, and Espinosa Zarlenga et al. (2022) replace each scalar by two input-dependent embeddings c_k^+ and c_k^-, one for the concept present and one for it absent, mixed by the predicted concept probability p_k as $p_k c_k^+ + (1-p_k)c_k^-$, which restores accuracy at the price of an embedding the concept name does not describe. ShifaMind takes a different route to capacity: a single scalar gate multiplies an entire concept slot, so an absent concept contributes nothing rather than a learned vector, closing the gate annihilates the slot, and the slot's width is a parameter whose effect on what the displayed scalar accounts for is measured causally (Section 6.1). The design also changes scale, from at most 112 concepts and a single-label target to a vocabulary of 16,227 concepts, 384 routed per note, and 7,940 codes with an exact per-code attribution.

3 Method

3.1 Problem formulation and notation

A note is a token sequence x of length $T \leq 6{,}144$. The target is a code set $y \in \{0,1\}^L$ over $L = 7{,}940$ ICD-10 codes. An encoder produces token states $H \in \mathbb{R}^{T \times d}$ with $d = 768$. A concept vocabulary $\mathcal{C}$ holds $|\mathcal{C}| = 16{,}227$ clinical concepts; each concept $k \in \mathcal{C}$ has a name, a set of surface forms, a learned query $q_k \in \mathbb{R}^d$ and a learned identity embedding $E_k \in \mathbb{R}^d$, both initialised from the en-

coder's mean-pooled embedding of the name. We write σ for the logistic function and $\langle \cdot, \cdot \rangle$ for the inner product. Figure 1 shows the forward pass.

3.2 Concept vocabulary and routing

The vocabulary holds 16,227 concepts. Their names were written by a language model from ICD-10 code titles in two passes, a shared vocabulary pass and a code-assignment pass; each name was then grounded by string match to a UMLS concept (Bodenreider, 2004), which supplies its surface forms (on average 11.3) and, with the generator's assignments, its links to ICD-10 codes (on average 4.00) (Appendix C). A dictionary matcher scans each note once with an Aho-Corasick automaton (Aho and Corasick, 1975) over all surface forms, and a ConText-style classifier (Harkema et al., 2009) assigns each mention one of five assertion classes: affirmed, negated, hypothetical, historical, or family. The routed set $\mathcal{R}(x) \subseteq \mathcal{C}$ holds the $K = 384$ concepts with the most affirmed or negated mentions in x; on the 2,000 validation notes of Section 6, 252 concepts per note are routed on average. Every routed concept is a literal string present in the note with a character span.

Routing admits negated mentions deliberately: if only affirmed mentions were routed, the gate supervision target of Section 3.5 would be all ones by construction. The affirmed share of routed slots is 84.2%, so the gate faces a genuine assertion task.

3.3 The multiplicative concept bottleneck

For each routed concept k, multi-head attention with eight heads reads the token states with q_k as the query and H as keys and values, $Z_k = \mathrm{MHA}(q_k, H, H) \in \mathbb{R}^d$. The readout attends over the whole note, not only the span that routed the concept; Section 6.1 measures what it carries. From Z_k the model forms a scalar score $\phi(Z_k) = \langle w_g, Z_k \rangle + b_g$ with learned $w_g \in \mathbb{R}^d$ and $b_g \in \mathbb{R}$, and

$$g_k = \sigma\big(\phi(Z_k)\big) \in (0, 1), \tag{1}$$

$$s_k = E_k + UDZ_k, \tag{2}$$

$$u_k = g_k \, s_k, \tag{3}$$

where $D \in \mathbb{R}^{r \times d}$ and $U \in \mathbb{R}^{d \times r}$ form a rank-r adapter and U is initialised to zero, so that $s_k = E_k$ at initialisation and any note-specific content in a slot must be learned. Slots of concepts outside $\mathcal{R}(x)$ are forced to $g_k = 0$ (a gate logit of -30).

Three properties follow from Eq. 3. *The gate is tied*: g_k is the number displayed for concept k and the multiplier on concept k's contribution, with no second head and no calibration layer between them. *Suppression is exact*: because the gate multiplies the whole slot, $g_k \to 0$ sends u_k to the zero vector, so concept k contributes exactly zero to every logit; its only remaining trace is one share of the attention normaliser in Eq. 5, which the suppression protocol of Appendix E removes as well. A design that adds the concept score to a shared representation cannot promise this, since zeroing one coordinate there leaves the rest free to carry the same information. Closing every gate leaves the model emitting its label prior. *Width is a dial*: at $r = 0$ the slot is the fixed vector E_k, the gate is the only note-dependent quantity, and the displayed number is the entire account of concept k's effect, which is the strict CBM. At $r > 0$ the slot also carries note-specific content, as in a concept embedding model, but every unit of it is still multiplied by the displayed gate. Section 6.1 measures how much of the decision the displayed number accounts for at each width.

3.4 Factored label attention and exact attribution

Each code ℓ has a learned query $a_\ell \in \mathbb{R}^d$, an output vector $w_\ell \in \mathbb{R}^d$ and a bias b_ℓ. The logit is

$$z_\ell = b_\ell + \sum_{k \in \mathcal{R}(x)} \underbrace{\alpha_{\ell k} \, \langle w_\ell, u_k \rangle}_{c_{\ell k}}, \tag{4}$$

$$\alpha_{\ell k} = \frac{\exp(\langle a_\ell, u_k \rangle / \sqrt{d})}{\sum_{j \in \mathcal{R}(x)} \exp(\langle a_\ell, u_j \rangle / \sqrt{d})}. \tag{5}$$

Both the attention scores and the projections are contractions of the slot tensor against the query and output matrices, so the $L \times d$ per-code representation that a direct implementation implies is never materialised; this is what makes an exact decomposition affordable at $L = 7{,}940$ and $K = 384$. With h attention heads the slot is split into h blocks and $c_{\ell k}$ sums over heads; the headline model uses $h = 1$; Appendix B reports a four-head arm.

The decomposition is the explanation. The term $c_{\ell k}$ in Eq. 4 *is* concept k's contribution to code ℓ; summing over k and adding b_ℓ reproduces z_ℓ, checked at every forward pass.

What to display. The natural quantity to display is the product $g_k \alpha_{\ell k}$. Under a sigmoid gate and

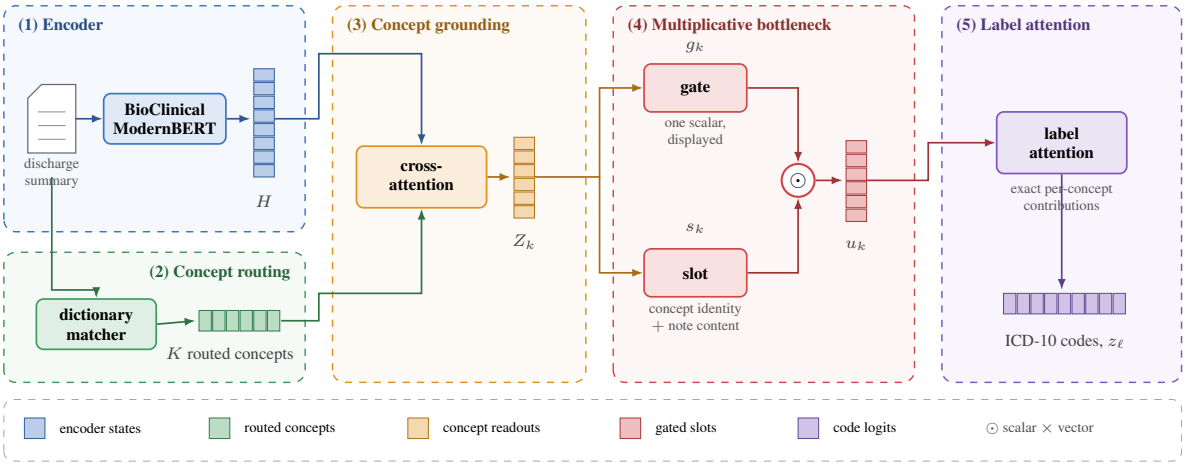


Figure 1: The forward pass. Token states reach a code only through routed concepts: each concept's readout becomes a gate, which is the displayed score, and a slot; their product is the concept's only representation at the output, and the label attention head sums per-concept contributions into every logit.

softmax attention it is non-negative everywhere, so a concept whose evidence lowers a code's logit cannot be identified as negative evidence by that score, and the displayed values do not sum to the logit. The signed term $c_{\ell k}$ does take negative values, and ranking evidence by $|c_{\ell k}|$ surfaces the strongest support and the strongest objection alike (Section 6.2).

Completeness. Two variants that are not Shifa-Mind serve as references in Section 5.3: an ungated residual from pooled token states, and a residual scaled by the concept support $\sum_k \alpha_{\ell k} g_k$.

3.5 Training

The objective is

$$\mathcal{L} = \tfrac{1}{2}\,\mathcal{L}_{\text{code}} + \lambda_c \mathcal{L}_{\text{gate}} + \lambda_s \mathcal{L}_{\text{sparse}} + \lambda_f \mathcal{L}_{\text{fid}}, \quad (6)$$

where $\mathcal{L}_{\text{code}}$ is binary cross-entropy summed over all L codes of all notes in the batch and divided by the number of gold codes in the batch (a cell mean at 0.2% positive rate sits near zero and stalls learning); $\mathcal{L}_{\text{gate}}$ is binary cross-entropy with logits between $\phi(Z_k)$ and a silver target that is 1 when concept k has an affirmed mention in the note and 0 otherwise, averaged over routed slots; $\mathcal{L}_{\text{sparse}}$ is the mean g_k over routed slots; and $\mathcal{L}_{\text{fid}}$ is an InfoNCE term (van den Oord et al., 2018) that recovers concept identity from the readout, with Z_k as the anchor and the identity embeddings E_j of the concepts in the batch as candidates. We use $\lambda_c = 1$, $\lambda_s = 0.05$, $\lambda_f = 0.3$, chosen on validation. The bias b_ℓ is initialised to the log-odds of code ℓ in the training notes.

Training on a second coding system. The concept layer does not depend on the coding system, so ICD-9-coded notes supervise the same concepts and only the label rows differ. We append the ICD-9 codes of the training notes of Edin et al. (2023)'s MIMIC-IV ICD-9 split as $L_9 = 6{,}145$ label rows with their own a_ℓ, w_ℓ, b_ℓ, dropped after training. Every batch holds one coding system and only its rows are scored, so an ICD-9 note never produces a gradient on an ICD-10 row and the exported ICD-10 logits are identical to the combined model's. To keep the benchmark's patient-level separation we remove every ICD-9 note whose patient or admission appears in the ICD-10 validation or test split (10,749 notes), leaving 143,780.

4 Experimental Setup

Data. We use MIMIC-IV discharge summaries (Johnson et al., 2023b,a) with ICD-10 diagnosis and procedure codes under the patient-disjoint splits of Edin et al. (2023), full-code setting: 89,089 training, 13,378 validation and 19,799 test notes over 7,940 codes; the test split holds 314,100 gold (note, code) pairs. We keep digits in the text, since laboratory values define several codes, and truncate at 6,144 tokens, which affects 1.1% of training notes.

Metrics and thresholds. We report micro and macro AUC-ROC, micro and macro F1, precision at 8 and 15, R-precision and MAP, following Edin et al. (2023). Macro-F1 is the mean of per-code F1 at the micro-optimal threshold (Edin et al. average over codes present in test, we over all 7,940;

7,933 are present, so nothing changes at one decimal). Both thresholds (one maximising micro-F1, one macro-F1) are selected on validation and fixed before test is scored. Baseline rows are quoted from Edin et al. (2023) and Wu et al. (2025); none was re-run. Concept share, used in Table 2, is the fraction of each predicted logit's mass carried by named concepts.[1]

Implementation. The encoder is BioClinical ModernBERT base (Sounack et al., 2025), 22 layers, $d = 768$. AdamW (Loshchilov and Hutter, 2019), peak learning rate 5×10^{-5} for the encoder and ten times that elsewhere, linear decay with 6% warmup, weight decay 0.01, 16 notes per step, bfloat16, $r = d = 768$ unless stated. The ICD-10-only model trains for six epochs (33,408 steps, 11 hours on one A100 40GB); the headline model for six epochs over the combined notes (87,324 steps, 12.5 hours on four A100s). Six epochs were chosen on validation (Appendix A).

5 Results

5.1 Main comparison

Table 1 reports the comparison. ShifaMind exceeds every system reproduced by Edin et al. (2023) on both F1 metrics, precision at 8 and 15, R-precision and MAP; against PLM-ICD, the strongest of them, the margin is +2.1 micro-F1 and +5.7 macro-F1. The ICD-10-only row is the like-for-like comparison on training data: +1.2 micro-F1 and +4.6 macro-F1 over PLM-ICD. Part of that margin is the encoder and its 6,144-token context rather than the bottleneck: at matched eight-epoch training, the same encoder with a gated residual pathway scores 0.5 micro-F1 and 3.3 macro-F1 above the concept-only model (Section 5.3). Against GoM-ICD, ShifaMind trails on micro-F1, precision at 8 and 15 and both AUC-ROC metrics by between 0.2 and 1.0 points (0.7 micro-F1, 0.8 precision at 8, 0.5 precision at 15, 0.2 micro and 1.0 macro AUC-ROC), and under GoM-ICD's own macro-F1 definition reaches 28.4 against their 28.5 at one seed. The headline model trains on ICD-9-coded notes that GoM-ICD does not use; with ICD-10 notes only the gap to the mixture is 1.6 micro-F1, while GoM-ICD's best single model scores 60.6 on one seed with a test-selected threshold (Wu et al., 2025, Table 5), which the headline model matches under a validation-fixed one. And every other row may route signal through any internal pathway, whereas in the ShifaMind rows every predicted logit is a sum of contributions from named concepts, each matched in the note.

5.2 Concepts transfer across coding systems

Training on ICD-9-coded notes through the shared concept layer improves ICD-10 coding by +0.9 micro-F1 and +1.1 macro-F1 over the ICD-10-only model, both as three-seed means, with no ICD-9 parameter retained at test time. For seed 10 of the headline model against each ICD-10-only seed, a paired bootstrap over test notes (1,000 resamples) gives three 95% intervals, all within 0.8 to 1.1 micro-F1 and 1.0 to 1.8 macro-F1, while seed 10 of the ICD-10-only model differs from seeds 11 and 12 by at most 0.2 with intervals that include zero. Figure 2 shows where the gain lands: +2.1 and +2.2 micro-F1 on codes with 10 to 99 training occurrences, against +0.6 on codes with 1,000 or more. At equal optimizer steps rather than equal epochs, where ICD-9 notes take 62% of the batches and ICD-10 codes receive correspondingly fewer updates, micro-F1 still rises (59.9) while macro-F1 falls (22.8): the extra notes help common codes immediately and rare ICD-10 codes only once they are seen as often as before.

5.3 The slot-width dial

Table 2 reads the dial from the strict CBM to the two designs that are not concept-mediated. Width is monotone at matched budget (49.0, 51.4, 52.7 micro-F1) and width 768 keeps its lead at full length. The two residual designs are within 0.5 micro-F1 of each other, so gating the residual costs almost nothing; removing it is what costs, and that cost shrinks with training length, from 8.4 macro-F1 at two epochs (against width 768) to 3.3 at eight (27.5 against 24.2 for the eight-epoch width-768 model), with micro-F1 within 0.5. Every point the residual buys is on rare codes, bought by carrying up to about 60% of the predicted logit mass out-

[1] GoM-ICD defines macro-F1 as the F1 of label-averaged precision and recall (their Eqs. 7, 10, 11), the formula Edin et al. (2023) identify as inflating earlier results; by Jensen's inequality it is never below the mean of per-code F1. Under that definition the headline model reaches 28.4 at the micro threshold and 31.4 at the macro threshold. GoM-ICD's Table 5 states that its thresholds are selected on the test split and its released code does the same; the protocol of its Table 4, the row we quote, is not stated. Re-tuning both thresholds on test, the Table 5 protocol, moves seed 10 of the headline model by less than 0.1 on micro-F1 (60.7) and on macro-F1 under their definition (31.6 against their Table 5 value of 32.6 for one seed).

Table 1: Test split, percent. The Edin et al. (2023) rows are ten-seed means (standard deviation at most 0.9; 2.3 for PLM-ICD macro-F1); the GoM-ICD row is quoted from Wu et al. (2025); both ShifaMind rows are three-seed means (standard deviation at most 0.1 and 0.3). n/r: not reported. †: GoM-ICD's macro-F1 definition (footnote 1); ShifaMind scores 28.4 under it.

	AUC-ROC		F1		Precision at				Concept
Model	Micro	Macro	Micro	Macro	8	15	R-prec.	MAP	share
CNN	97.5	87.9	47.2	8.0	60.3	45.7	47.3	48.2	none
Bi-GRU	98.3	92.4	50.1	10.6	62.6	47.7	49.6	51.1	none
CAML	98.5	91.1	55.4	16.0	66.8	52.2	54.5	57.4	none
MultiResCNN	99.0	94.5	56.9	21.1	67.8	53.5	56.1	59.3	none
LAAT	99.0	95.4	57.9	20.3	68.9	54.3	57.2	60.6	none
PLM-ICD	99.2	96.6	58.5	21.1	69.9	55.0	57.9	61.9	none
GoM-ICD	**99.4**	**97.7**	**61.3**	28.5†	**72.6**	**57.3**	n/r	n/r	none
ShifaMind, ICD-10 notes only	99.2	96.6	59.7	25.7	71.2	56.0	59.1	63.0	**100**
ShifaMind	99.2	96.7	60.6	26.8	71.8	56.8	**60.1**	**64.2**	**100**

Table 2: Matched: two epochs on the six-epoch schedule, validation, macro-F1 at the macro threshold. Full length: test, macro-F1 at the micro threshold; six epochs and three seeds for width 768, eight epochs and one seed otherwise. Probe columns: matched-budget concept-only models, 2,000 validation notes; gates set to the assertion flags, slots swapped for another patient's readout with gates fixed, comprehensiveness of the exact ranking. Residual rows: width 16 at matched budget, 768 at full length; the ungated row uses the legacy slot $s_k = Z_k$. The gated-residual share is a lower bound, since its support factor inflates the denominator of the share.

	Matched		Full length			Gate	Gates to	Slots	Compre-
Design	Micro	Macro	Micro	Macro	Share	AUROC	flags Δ	swapped Δ	hensiveness
Width 0 (strict CBM)	49.0	9.5	not run		100	0.989	−15.0	0.0	0.65
Width 16	51.4	11.9	58.6	22.8	100	0.997	−0.2	−17.6	0.73
Width 768 (ours)	52.7	13.7	59.7	25.7	100	0.996	−0.3	−20.3	0.64
+ gated residual	56.8	22.1	59.6	27.5	≥39.9	not concept-mediated			
+ ungated residual	57.3	22.4	not run		39.2	not concept-mediated			

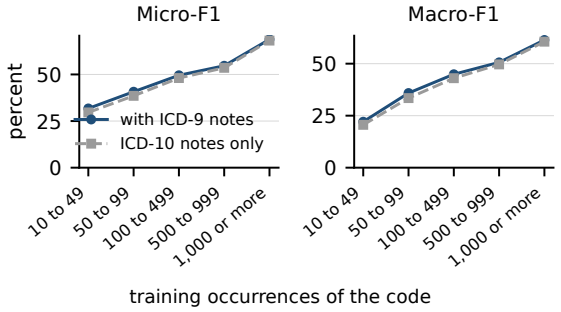


Figure 2: Test F1 by training frequency of the code, micro threshold; the bins hold 4,242 / 979 / 1,076 / 227 / 251 codes, and the 1,165 codes with fewer than 10 training occurrences are not shown. Headline model: seed 10; ICD-10-only: mean of three.

side the concepts.

The probe columns locate where the decision lives along the dial. At width 0 the gate is graded (mean 0.97 on affirmed mentions) and replacing it by the binary assertion flags costs 15.0 micro-F1, while swapping slot content changes nothing by construction. At widths 16 and 768 the gate saturates (0.99), binarising it costs at most 0.3, and swapping slot content costs 17.6 and 20.3: as the slot widens, the note-specific information moves out of the gate's magnitude and into the slot. Comprehensiveness and plausibility change little along the dial (0.64 to 0.73; linked top-3 concepts 33 to 39%).

6 Interpretability Evaluation

Every quantity in this section is computed from the gates g_k and contributions $c_{\ell k}$ of the headline model (seed 10) unless stated, so each metric scores the computation that produced the prediction and not a separate explanation head. The concept pathway carries 100% of every predicted logit; Eq. 4 reconciles to the logit at every call.

6.1 Causal tests

Suppression. Suppressing all 16,227 concepts returns the label prior by construction (micro-F1 8.0 from 60.7, macro AUC-ROC 50.0), a check of completeness rather than evidence. The con-

Table 3: Headline model, validation split: suppression over all 13,378 notes, transplantation over 2,000.

Intervention	Micro-F1	Δ
Concept suppression		
none	60.7	0.0
all 16,227 concepts	8.0	−52.7
least-used 90%	55.2	−5.5
random 90%	23.3	−37.4
Slot transplantation, gates held fixed		
own readout (identity check)	60.8	0.0
concept mean over other notes	36.9	−23.9
same concept, another patient	31.1	−29.7
a different concept, another patient	7.2	−53.6

trols carry the evidence: suppressing the 90% of concepts the model uses least by total gate mass, which largely means the concepts routed least often, costs 5.5 micro-F1; suppressing a random 90% costs 37.4, 6.8 times more. Suppressing the least-used 50% costs under 0.1 micro-F1: the accuracy sits in the few thousand concepts the model routes and opens most.

Transplantation. The bottleneck guarantees that all signal passes through some concept's gate, not that the displayed gate *value* is the whole story: at $r > 0$ the slot of Eq. 2 also carries note-specific content. We test this causally: readouts Z_k are collected from other notes, then each note is rerun with every slot's content replaced while every displayed gate keeps its true value. Replacing slot content removes 23.9 to 29.7 micro-F1 and the predicted code sets overlap the originals by only 25 to 32%; another patient's genuine readout for the same concept is worse than the concept mean. These results cover the 48.2% of concepts observed in other notes. At width 16 the drop is 26.2 of 58.8 after eight epochs: narrowing the adapter does not return the decision to the gate.

Stated plainly: at $r = 768$, with every displayed gate held at its true value, replacing what the slots carry removes 23.9 to 29.7 of 60.8 micro-F1, so the displayed gate alone does not determine the prediction. The contract holds, since that content still travels inside concept k's slot, is multiplied by its gate, and vanishes when the concept is suppressed. But the stronger claim that the displayed number is a complete account of concept k's effect holds only at $r = 0$, and width 16 does not recover it, so that guarantee costs the full width-0 price, 3.7 micro-F1 at matched budget (Table 2) and unmeasured at full length.

Table 4: Headline model, 2,000 validation notes (recall row: test split; stability row: the three ICD-10-only seeds). Comprehensiveness and sufficiency average over $n \in \{1, 3, 5, 10\}$ removed or kept concepts; concept accuracy uses the silver labels that supervised the gate.

Metric	Value
Concept accuracy: AUROC / F1 at 0.5	0.999 / 0.998
mean gate, affirmed / negated	0.998 / 0.013
Comprehensiveness: exact $\lvert c_{\ell k}\rvert$ ranking	0.64
gate × attention / random	0.63 / 0.01
Sufficiency: exact / random ranking	0.98 / 0.26
Intervention accuracy (own gates 60.8)	
gates set to silver assertion flags	60.8
all routed gates open	60.0
gates shuffled within note	57.8
Plausibility: top-3 linked / random (%)	37.3 / 1.0
Recall with / without a linked concept	62.4 / 56.5
Explanation size: concepts for 80% / 90%	3 / 4
Stability: Jaccard@5 / top-1 in top-5 (%)	0.54 / 95

6.2 Interpretability metrics

Table 4 reports the metrics. Concept accuracy and intervention accuracy are the two tests Koh et al. (2020) define for a concept bottleneck: does the model predict each concept correctly, and does editing a concept change the prediction as the concept's weight says it should. Comprehensiveness, sufficiency and plausibility follow the ERASER protocol (DeYoung et al., 2020), with concepts in place of tokens as the rationale units and the vocabulary's concept-to-code links as the reference for plausibility; sufficiency is reported as the probability retained, so higher is better. Explanation size is the number of concepts that carry a fixed share of a code's positive support, and stability is the overlap of the top concepts across training seeds. Concept-conditioned recall, the recall of the coder on gold pairs for which a linked concept is asserted in the note, is added on the test split.

Concept accuracy. Against the silver assertion labels of its own slot, the gate reaches AUROC 0.999 over 503,786 routed slots, with mean value 0.998 on affirmed mentions, 0.013 on negated ones, and 0.07 to 0.14 on family, hypothetical and historical mentions. The gate therefore decides *whether* a concept counts, and does so almost binarily; *how much* it counts is carried by the signed contribution $c_{\ell k}$, which is why a code's explanation lists $c_{\ell k}$ beside each concept, with g_k as the concept's assertion score. Intervention accuracy, a defining test of a concept bottleneck, is intact: re-

placing the gates by the silver assertion flags, the setting a coder who confirmed every assertion label would make, leaves micro-F1 unchanged (60.8 against 60.8), showing agreement with the silver assertions without loss of coding accuracy; opening every gate costs 0.8, shuffling gates within a note 3.0, and setting every gate to 0.5 collapses micro-F1 to 11.0.

Comprehensiveness and sufficiency. Deleting the concept with the largest $|c_{\ell k}|$ removes 0.26 of the target code's probability on average, the top three 0.65, the top ten 0.86 (comprehensiveness); keeping only the top concept retains 0.99 (sufficiency). Deleting concepts chosen by gate value alone or at random removes at most 0.01, so the ranking, not the number of concepts removed, is what matters. The exact ranking and the attention ranking tie on these two measures (0.64 against 0.63), as the saturated gate predicts; the exact term differs from the attention product in sign, and that difference is what makes objections visible (Section 6.3). Two cautions: after a deletion the attention of Eq. 5 renormalises over the remaining concepts, so deletion effects are not additive in $c_{\ell k}$ and a kept concept's attention rises towards 1, which flatters sufficiency; and the target codes are predicted positives with probabilities near 1, so one deletion moves probability less than its logit share suggests.

Plausibility, size and stability. For a predicted code, 37.3% of its three strongest supporting concepts are concepts the vocabulary links to that code, against 1.0% for three random routed concepts of the same note. The links are far from complete: only 30.6% of gold test pairs have a linked concept asserted in their note, and the model recovers 62.4% of those codes against 56.5% of the codes without a recorded link (concept-conditioned recall). Most evidence therefore travels through concepts the concept-to-code graph does not link to the code; that graph combines generated assignments with UMLS mappings and is not independently annotated evidence. A prediction rests on few concepts: a median of 3 concepts carries 80% of a code's positive support and 4 carry 90%. Explanations are stable across training seeds: for the same (note, code) pairs, the ICD-10-only seeds share 0.54 of their top-5 concepts (Jaccard, mean over two pairs), and one seed's strongest concept lies in the other's top 5 for 95% of pairs.

Table 5: Two codes on one test note (3,345 tokens, 11 gold codes, 10 of 12 predictions correct). Rows are the largest terms $c_{\ell k}$ of Eq. 4 by magnitude, every other N17.9 term being below 0.04; with the remaining concepts and the biases (-6.08, -1.66) they sum to the logits ($+3.47$, $+3.07$). All gates were 1.00.

Concept	$\alpha_{\ell k}$	$c_{\ell k}$
K75.4 autoimmune hepatitis, $p = 0.97$		
autoimmune hepatitis	0.660	$+11.13$
therapeutic substance	0.061	-0.65
release	0.017	-0.33
subsequent encounter	0.013	-0.27
N17.9 acute kidney failure, $p = 0.96$		
elevated creatinine	0.456	$+2.11$
specified disease	0.244	$+1.43$
calcium-channel blocker	0.244	$+1.20$

6.3 A worked explanation

Table 5 shows the accounts a coder would read. For K75.4 one concept carries the code and the objections are small; they would be invisible under the non-negative product $g_k\alpha_{\ell k}$. For N17.9 no concept names the code: the evidence is a laboratory finding and a drug class, plus a generic entry ("specified disease") the model has learned to weight. Its one missed gold code (a urinary tract infection) had probability 0.009 because no routed concept supported it.

7 Conclusion

ShifaMind makes a coder's computation its own explanation: a gate that is at once the displayed score and the multiplier, an exact decomposition of every logit into named concepts, and no pathway that bypasses them. Under that constraint it exceeds every system reproduced by Edin et al. (2023) on both F1 metrics and comes within 0.7 micro-F1 of the strongest reported mixture, and two gaps are now measured: concept mediation against an unconstrained coder, and what the gate shows against what the slot carries. The concept layer is independent of the coding system, and ICD-9-coded notes raise ICD-10 micro-F1 by 0.9 and macro-F1 by 1.1 through the same encoder and concepts. Seven standard interpretability metrics, from concept accuracy to seed stability, and two causal tests accompany the accuracy results.

8 Limitations

The displayed gate is not the whole account at $r = 768$. Section 6.1 measures this directly. We report the width we trained rather than the width at

which the claim is cleanest, and we give the cost of the clean width.

Gate supervision is automatic. Concept targets come from a dictionary matcher with assertion classification, not from human annotation. Matcher errors propagate into the gates a coder reads, which is why an interface should show the matched span beside the gate, and why concept accuracy is reported against the same silver labels rather than against expert annotation. Codes justified by historical or family mentions, such as personal and family history and status codes, are supported only through concepts whose gates close on such mentions; their accuracy was not analysed separately.

The link graph is incomplete. Only 30.6% of gold test pairs have a concept the vocabulary links to the code asserted in their note, and recall without a recorded link is 56.5% against 62.4% with one. The model therefore relies on concepts the concept-to-code graph does not link to the code, which the plausibility metric counts as implausible. Of the 7,933 codes present in test, 44.3% are never predicted correctly. The vocabulary and its links, not the architecture, are the first place to look.

Seeds and scope. Both ShifaMind models are reported over three seeds. All results are on MIMIC-IV ICD-10 full coding. No clinician study was run; the metrics here measure faithfulness and consistency, not usefulness to a coder.

Controls not run. A conventional label-attention head on the same encoder with the same preprocessing and schedule, which would separate the cost of the constraint from the encoder; a multitask head without concepts trained on both coding systems, which would show whether the transfer of Section 5.2 needs the concept layer; and the width-0 model at full training length.

Comparability. GoM-ICD combines three models, augments training notes, and states test-selected thresholds for its Table 5; ShifaMind is a single model with thresholds fixed on validation but trains on ICD-9-coded notes that no baseline uses. Edin et al. (2023) report 122,279 notes and 7,942 codes; our join of their split file to the note release yields 122,266 notes (89,089 / 13,378 / 19,799) and 7,940 codes, a difference of 13 notes and 2 codes. Unlike the baselines, which remove all digits, we keep them, since Edin et al. (2023) trace errors on body-mass-index and gestation codes to digit removal; this departs from the baselines' preprocessing and may favour our model on those codes. We truncate at 6,144 subword tokens rather than 4,000 words; Edin et al. (2023) found on MIMIC-III clean that truncating at 2,500 rather than 4,000 words changed micro-F1 by at most 0.6. We report the comparison as published rather than re-running the baselines.

Ethics Statement

MIMIC-IV and MIMIC-IV-Note are credentialed resources under a PhysioNet data use agreement. No note text left the training machine, no note text appears in this paper, and the concept vocabulary contains only terminology strings. UMLS content is used under its licence. Automated coding affects billing and care records; ShifaMind is designed to support, not replace, a human coder, and its explanations should be read with the matched span in view. Upon acceptance we will release the code, the concept vocabulary (concept names and code links) and the scripts that rebuild its UMLS surface forms under a UMLS licence, and share trained weights under the PhysioNet terms for models trained on MIMIC.

References

Alfred V. Aho and Margaret J. Corasick. 1975. Efficient string matching: An aid to bibliographic search. *Communications of the ACM*, 18(6):333-340.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl 1):D267-D270.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew Gormley. 2023. MDACE: MIMIC documents annotated with code evidence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7534-7550. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443-4458. Association for Computational Linguistics.

Joakim Edin, Alexander Junge, Jakob D. Havtorn, Lasse Borgholt, Maria Maistro, Tuukka Ruotsalo,

and Lars Maaløe. 2023. 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. ACM.

Joakim Edin, Maria Maistro, Lars Maaløe, Lasse Borgholt, Jakob Drachmann Havtorn, and Tuukka Ruotsalo. 2024. An unsupervised approach to achieve supervised-level explainability in healthcare records. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4869-4890. Association for Computational Linguistics.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, Pietro Lio, and Mateja Jamnik. 2022. Concept embedding models: Beyond the accuracy-explainability trade-off. In *Advances in Neural Information Processing Systems*, volume 35.

Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. 2009. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839-851.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. 2022. Addressing leakage in concept bottleneck models. In *Advances in Neural Information Processing Systems*, volume 35.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10-20. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543-3556. Association for Computational Linguistics.

Alistair Johnson, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2023a. MIMIC-IV-Note: Deidentified free-text clinical notes (version 2.2).

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. 2023b. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10:1.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338-5348. PMLR.

Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8180-8187.

Jinghui Liu and Anthony Nguyen. 2026. Bridging the version gap: Multi-version training improves ICD code prediction, especially for rare codes. In *BioNLP 2026*, pages 372-381, San Diego, California. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. 2021. Promises and pitfalls of black-box concept learning models. *arXiv preprint arXiv:2106.13314*.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101-1111. Association for Computational Linguistics.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206-215.

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

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for ICD coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3335-3341.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11-20. Association for Computational Linguistics.

Yifan Wu, Weiyan Qiu, Min Zeng, Xi Chen, Min Li, and Hongtao Zhu. 2025. GoM-ICD: Automatic ICD coding with gap schemes and mixture of experts. *Big Data Mining and Analytics*, 8(6):1211-1224.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. 2022. Code synonyms do matter: Multiple synonyms matching network for automatic ICD coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 808-814. Association for Computational Linguistics.

A Hyperparameters and selection

Learning rate was selected by matched-step comparison on validation (5×10^{-5}: 56.9 micro-F1 at step 19,488; 10^{-4}: 54.7; 10^{-5}: 53.5; 3×10^{-6}: 43.6) and confirmed by a learning-rate range test whose usable band spanned 3×10^{-6} to 3×10^{-4}. Weight decay 0.1 scored below weight decay 0.01 at every matched step, by 0.6 micro-F1 on average. Dropout 0.1 on the concept attention and label head; gradient clipping at 1.0. A fifteen-epoch grid with the same schedule found both AUC metrics declining after roughly epoch five while F1 plateaued; six epochs were chosen from those validation curves, and an eight-epoch schedule trained afterwards scored below six on every test metric (59.1 against 59.7 micro-F1). Validation every 2,784 steps; the checkpoint with the best validation micro-F1 is scored once on test at the two thresholds it pinned. Validation and test micro-F1 agree to 0.1 for both ShifaMind rows in Table 1.

B Matched-budget sweep

Table 6: Eight arms, two epochs each on the six-epoch learning-rate schedule, validation. Each arm differs from the reference (width 16) in the stated setting, except the ungated residual arm, which uses the legacy slot $s_k = Z_k$. Macro-F1 at the macro-optimal threshold.

Arm	Micro-F1	Macro-F1	Share
Ungated residual	57.3	22.4	39.2
Gated residual	56.8	22.1	39.3
Width 768	52.7	13.7	100
Width 16 + λ_s, λ_f	51.5	12.3	100
Width 16, 2 attention layers	51.5	12.5	100
Width 16 (reference)	51.4	11.9	100
Width 16, 4 label heads	50.4	10.3	100
Width 0	49.0	9.5	100

C Concept vocabulary and matcher

Concept names come from a language-model generator prompted with ICD-10 code titles: a first pass builds a shared vocabulary in ICD-chapter order and is told to reuse names before inventing them, a second pass assigns each code to names in that vocabulary, and ICD-10-PCS axis titles are added as concepts for procedure codes. Each name, or failing that one of its generated surface forms, is matched by normalised string to a UMLS concept; strings that map to several concepts are skipped. The matched concept supplies the surface forms, taken from curated clinical source vocabularies (SNOMED CT, MeSH, RxNorm, LOINC, ICD-10-CM, ICD-10-PCS, ICD-9-CM, MedDRA and others; organism and gene-symbol sources are excluded) and restricted to the semantic groups disorders, chemicals and drugs, procedures, anatomy, physiology, devices and phenomena. A surface form is kept only if it is 3 to 80 characters long, contains a letter and is not on a stop list of ordinary English words (“history”, “normal”, “left”); when a concept exceeds the cap on forms the shortest are kept. Concept-to-code links are the generator’s assignments that exist in the label vocabulary plus the UMLS ICD-10-CM mappings of the matched concept. Concepts left without a surface form are removed. The matcher runs one Aho-Corasick pass per note with word-boundary checks and longest-match resolution. Assertion follows ConText with four non-affirmed classes; section headers override only affirmed mentions. Silver targets store mention counts per assertion class, which are needed to rank concepts when more than K fire. Initial queries for left and right concept pairs (809 pairs) sit at cosine similarity 0.988 after mean pooling of the name; a single global laterality axis added to the queries lowers this to 0.947. In the headline model on validation, the mean over concepts of each concept’s mean gate when routed is 0.91, and the median per-concept open rate is 1.0.

D Concept mediation at a glance

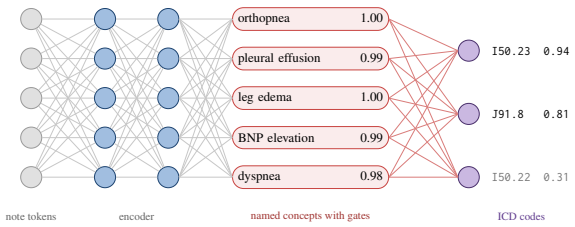


Figure 3: What concept mediation means. Token states pass through the encoder, but a code can be reached only through named concepts, each shown with the gate that scales it; the number beside a concept is the number used in the prediction. Concept names and values are illustrative.

E Probe protocols

Suppression. A suppressed concept has its routing mask zeroed, which the model turns into a gate logit of -30, a gate of effectively zero, and an attention score of $-\infty$; nothing in the weights changes. Thresholds are pinned from the unsuppressed reference so a uniform shift in scores cannot be absorbed by re-tuning.

Transplantation. Up to eight readouts per concept are banked from a first pass over 2,000 validation notes. In the second pass the slot of every routed concept is rebuilt from a substitute readout through Eq. 2 while the gate of Eq. 1 is computed from the note's own readout, so displayed gates are unchanged.

Faithfulness, intervention, plausibility, stability. Token states are cached once per note and only the head is re-run. Faithfulness uses one target code per note, the highest-probability predicted code that is also gold when one exists. Intervention replaces every routed gate before the head is re-run; the silver flags are the assertion labels of Section 3.2. Plausibility counts a concept as linked to a code when the vocabulary records that link. Stability loads the other seeds' weights and recomputes the top-5 concepts for the same (note, code) pairs.