Mind the Gap: A Diagnostic Study of Hybrid ICD Coding

Anonymous ACL submission

Abstract

Automated ICD coding maps clinical notes to tens of thousands of codes. Discriminative classifiers perform well on common codes but cannot predict codes outside their training codeset. The label space is long-tailed, and the codebook is revised every year. LLM agent pipelines reach the long tail but cost more and underperform on common codes. We build a two-stage hybrid: a supervised classifier whose mid-confidence band is routed to an LLM verifier and a retrieval-based gap-filler, and use it to diagnose the gap of clinical coding on MIMIC-III and MIMIC-IV. Our contributions are threefold. First, we identify three distinct failure modes through oracle ablation: LLM band verification, gap-fill flooding, and terminology mismatch. Second, the hybrid matches the macro-optimal classifier on seen codes on MIMIC-III and is within 0.4 macro-F1 points on MIMIC-IV, while additionally predicting out-of-distribution codes the classifier cannot reach by construction. Third, because retrieval queries the live codebook, the system is architected to absorb annual edition updates by rebuilding the retrieval index alone, with no classifier retraining; we exercise this mechanism on a within-codebook vocabulary-shift proxy, leaving a paired before/after edition swap to future work.

1 Introduction

Automated assignment of International Classification of Diseases (ICD) codes to clinical notes is critical for cost reimbursement, epidemiological surveillance, and healthcare resource planning (Dong et al., 2022).[1] Yet it remains among one of the most difficult real-world text classification problems. Concretely, given a free-text discharge summary, the task is to predict the set of standardised codes a trained human coder would assign for billing and surveillance.[2] For example, ICD-9-CM (Clinical Modification) `410.71` denotes *subendocardial infarction, initial episode of care*. It is an extreme multi-label task over a label space of tens of thousands of entries. ICD is a family of jurisdictional variants, e.g., US ICD-10-CM (Clinical Modification) and Australian ICD-10-AM (Australian Modification) issued in new editions on fixed cycles. ICD-9-CM and ICD-10-CM contain roughly 14,000 and 68,000 codes respectively (Centers for Medicare & Medicaid Services and National Center for Health Statistics, 2014; National Center for Health Statistics, 2024). The upcoming ICD-11 consists of over 120,000 codable terms mapping to around 17,000 codes (World Health Organization, 2024). Code frequency follows a power law, producing a persistent 20- to 30-point gap between micro-F1 and macro-F1 even for state-of-the-art systems (Edin et al., 2023).

Beyond the long tail, any deployed coder also faces *label-space evolution*. A discriminative classifier is bound to the codes it was trained on; codes introduced or redefined after the training snapshot are structurally invisible. Re-training to track each codebook revision is costly and requires re-annotated data rarely available at release time. This motivates designs in which the deployed system can absorb codebook updates without re-training.

Two paradigms dominate current work, with complementary weaknesses. Discriminative supervised fine-tuning (SFT) classifiers (Huang et al., 2022; Edin et al., 2024; Wu et al., 2025) excel at micro-F1 on frequent codes but collapse on the long tail and cannot reach codes outside their training codeset. Large Language Model (LLM) based agent pipelines, notably Code Like Humans (CLH) (Motzfeldt et al., 2025), achieve higher macro-F1 through open-set retrieval. They fall behind on fre-

[1] Code, prompts, and schemas are released at `https://anonymous.4open.science/r/hybrid-gap-0C56`.

[2] In routine practice, coders consult the full medical record; following the standard MIMIC benchmarks, we treat the discharge summary as the available note (§4.1).

quent codes, however: open-set generation trades head-code precision for rare-code recall and is substantially more expensive at inference.

We propose a two-stage hybrid that couples a discriminative classifier with an LLM verifier and retrieval gap-filler (Figure 1). **Stage 1** is a supervised classifier that emits per-code probabilities. High-confidence predictions are accepted directly; low-confidence predictions are discarded; a mid-confidence *verification band* between two validation-tuned thresholds is routed to Stage 2. **Stage 2** is an LLM pipeline that verifies the routed codes against note text and retrieves additional codes the classifier missed entirely. The final prediction is the union of accepted high-confidence codes, band-verified codes, and gap-retrieved codes (§3). Oracle ablation over the pipeline's saved decisions, which replaces each LLM stage with a ground-truth-perfect version to measure the headroom it leaves, identifies three distinct failure modes (§6).

Contributions. Our contribution is threefold.

First, a failure-mode analysis of hybrid SFT–LLM coding via oracle ablation. By replacing the LLM stage with an oracle, we isolate three distinct failure modes that bound end-to-end performance: verification of mid-confidence classifier predictions, open-set retrieval over the codebook, and terminology mismatch between clinical text and code descriptions.

Second, a competitive end-to-end pipeline. The pipeline matches a strong macro-optimal SFT baseline on in-distribution codes while additionally assigning out-of-distribution codes, yielding a macro-F1 lift on the union codeset at modest micro-F1 cost.

Third, codebook updates without retraining. Because retrieval operates over the live ICD codebook, annual edition updates are reachable by rebuilding the retrieval index alone, with no SFT retraining. Our OOD experiment exercises this mechanism on a within-codebook proxy, establishing the architectural property; a paired before/after edition swap is deferred to future work.

2 Related Work

Discriminative models. Deep learning approaches to automated ICD coding have progressed from convolutional architectures (CAML, MultiResCNN) (Mullenbach et al., 2018; Li and Yu, 2020), through graph- and hierarchy-aware methods (Cao et al., 2020), to transformer encoders with label-wise attention (LAAT, PLM-ICD, MSMN) (Vu et al., 2020; Huang et al., 2022; Yuan et al., 2022) and its hierarchical and extreme multi-label extensions (HiLAT, XR-LAT) (Liu et al., 2022, 2023), and most recently to label–text cross-attention over pre-trained backbones (PLM-CA, CoRelation) (Edin et al., 2024; Luo et al., 2024). We adopt the cleaned MIMIC splits of Edin et al. (2023) throughout (§4.1). On those splits, Wu et al. (2025) (**GoM-ICD**) achieve state-of-the-art discriminative micro-F1 (0.620 on MIMIC-III-CLEAN) via a bidirectional long short-term memory (BiLSTM) mixture-of-experts architecture.

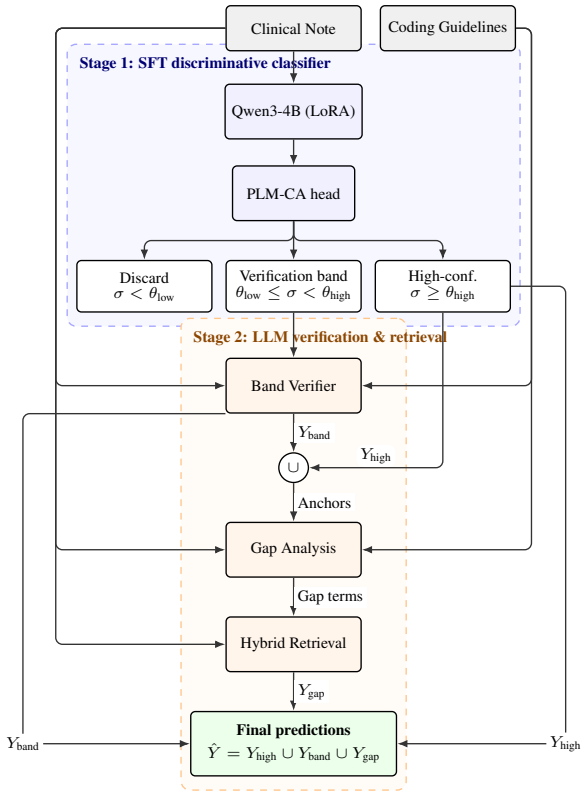

Figure 1: Two-stage hybrid pipeline for automated ICD coding. **Stage 1** partitions per-code probabilities $\sigma(s_c)$ by two thresholds into a discard set, a verification band, and a high-confidence set Y_{high}. **Stage 2** runs an LLM verifier over the band to produce Y_{band}; a prediction-aware gap-analysis LLM with hybrid retrieval populates Y_{gap}. Final predictions $\hat{Y} = Y_{\text{high}} \cup Y_{\text{band}} \cup Y_{\text{gap}}$.

LLM agent pipelines. Direct prompting of LLMs for ICD coding consistently underperforms discriminative baselines (Falis et al., 2024; Soroush et al., 2024), motivating a shift to agentic decompositions that mirror how professional coders work. Motzfeldt et al. (2025) (CLH) implement an Analyze–Locate–Assign–Verify workflow over

the full ∼70K ICD-10-CM codes, reporting higher macro-F1 than PLM-ICD on the MDACE benchmark but lower micro-F1. Li et al. (2025) (MAC) and Li et al. (2024) adopt similar multi-agent decompositions with role-specialised prompts.

Hybrid approaches. Hybrid approaches combine discriminative classifiers with LLM components at inference in two patterns. *LLM-first* systems have an LLM generate candidates that a discriminative verifier scores (Yang et al., 2023a; Li et al., 2025); both inherit the verifier's fixed label space lock-in, leaving codes outside its label set unrecoverable. *Retrieve-then-rerank* systems apply BM25 or dense retrieval over the full label space followed by an LLM or cross-encoder reranker (Wang et al., 2024; Kwan, 2024; Baksi et al., 2025); they open the label space but lack a per-instance prior, bounding reranker precision by first-stage recall, and recent biomedical evidence shows LLM rerankers can degrade retrieval without joint retriever–reranker alignment (Rivera and Menolascina, 2025; Puts et al., 2025).

Generalisation to rare and unseen codes. A parallel line of work targets generalisation beyond the training distribution: zero- and few-shot reformulations (Rios and Kavuluru, 2018; Song et al., 2021; Yang et al., 2023b), hierarchical curriculum learning (Ren et al., 2022), and knowledge-graph-augmented classifiers (Yang et al., 2022; Zhang et al., 2025). Recent LLM-based work attacks generalisation directly: Zhang et al. (2026) fine-tune smaller LLMs with span-level supervision, while Motzfeldt et al. (2025) extend coverage to the full ∼70K ICD-10-CM space via open-set index retrieval and report the strongest rare-code performance to date. Across these approaches, *improvements on the tail typically trade off against micro-F1 on the head.*

3 Method

3.1 Stage 1: SFT Discriminative Model

We predict a code set $\hat{Y} \subseteq \mathcal{L}$ for note x, where $\mathcal{L}$ is the ICD code space; the task is extreme multi-label classification under a long-tailed label distribution.

We pair a Qwen3 backbone with a per-label cross-attention head over the backbone hidden states, following PLM-CA (Edin et al., 2024); per-code probabilities $\sigma(s_c)$ are obtained by a sigmoid over the resulting logits (architecture and training recipe in Appendix A). We train with binary cross-entropy using LoRA adapters (Hu et al., 2022) and an AdamW/OneCycle recipe. We adopt **Qwen3-4B-Base** after a sweep over $\{0.6, 1.7, 4, 8\}$B base and instruction-tuned variants (Appendix B).

Operating points. Stage 1 outputs per-code posteriors $\sigma(s_c)$. Both decision thresholds for Stage 2 are read off the SFT validation curve as the macro-F1- and micro-F1-optimal operating points: $\theta_{\text{low}} = \tau^{\star}_{\text{macro}}$ and $\theta_{\text{high}} = \tau^{\star}_{\text{micro}}$. The interval $[\tau^{\star}_{\text{macro}}, \tau^{\star}_{\text{micro}}]$ is the region of SFT scores on which the macro-F1- and micro-F1-optimal decisions disagree (Lipton et al., 2014); we route exactly this region to the LLM verifier (Appendix A). The verifier's task is per-code confirmation against the note: the band selects which codes are routed, and each is judged independently on quoted note evidence (§3.2).

3.2 Stage 2: LLM Pipeline

All Stage-2 LLM passes share a single frozen open-weights backbone (GPT-OSS-20B in our main experiments; see §4.4 for the full configuration and Appendix F for cross-scale comparisons). Outputs use JSON-schema constrained decoding (schemas in Appendix C) and are post-hoc validated against the source note, so every reported code carries a verbatim evidence quote.

Guideline grounding. Every Stage-2 LLM pass is grounded in the official ICD-9-CM, ICD-10-CM, and ICD-10-PCS (Procedure Coding System) coding manuals: relevant rules are retrieved per call from a pre-built rule index (Appendix C).

Band verification. Stage 1 posteriors $\sigma(s_c)$ are partitioned by the validation-tuned thresholds $(\theta_{\text{low}}, \theta_{\text{high}})$: codes with $\sigma(s_c) \geq \theta_{\text{high}}$ are accepted as high-confidence predictions, codes in $[\theta_{\text{low}}, \theta_{\text{high}})$ form the verification band routed to the LLM, and codes with $\sigma(s_c) < \theta_{\text{low}}$ are discarded. The band verifier receives the note together with chunked $\langle$code, description$\rangle$ pairs and returns a CONFIRMED/RULE OUT decision per code, each carrying a verbatim evidence or contradiction quote (schema and decision rules in Appendix C). The union of the high-confidence tier and the band-verified codes forms the VERIFIED SET consumed by the remainder of the pipeline.

Gap analysis. A single LLM call per note runs a "gap analyst" task: given the note and the VERIFIED SET rendered as code and description pairs, it returns diagnoses and procedures documented in

the note but not covered (Appendix C). Each item must carry a verbatim quote subsequently verified as a substring of the note. Because the verified set has already been confirmed in §3.2, the gap analyst treats it as reliable context rather than as candidates under review.

Hybrid candidate retrieval. Each extracted diagnosis or procedure term is normalised and expanded into ICD candidates by fusing, via Reciprocal Rank Fusion, a dense BioLORD-2023-C bi-encoder and a BM25 lexical index over code descriptions plus their Unified Medical Language System (UMLS) `MRCONSO` synonym pool. We keep the top K=5 fused candidates per term subject to a minimum-score threshold (floor); full retrieval recipe (pooling, stemming, RRF k, source-abbreviation filters) is in Appendix D.

Candidate verification. The top-K candidate codes for each term are verified under a *mutual-exclusivity* instruction: at most one candidate per term may be confirmed, since the retrieved list represents competing codes for the same extracted concept. The LLM receives the note, the extracted term, and the candidate list with retrieval scores and guideline hints, and returns a verdict per candidate under the schema of §3.2. Codes marked `Confirmed` are added to the predicted set; rejected candidates and terms with no passing candidate are dropped.

The predicted set is the union of the high-confidence SFT codes, the band-confirmed codes, and the gap-confirmed codes. Evaluation protocol, including the INDIST vs. OOD decomposition, is presented in §4.

4 Experimental Setup

4.1 Datasets

We use three datasets across the two stages of our pipeline (Table 1). MIMIC-III-CLEAN and MIMIC-IV are used to both train the Stage 1 SFT classifier and evaluate the full system. MDACE is used only to tune the Stage 2 LLM. All splits follow Edin et al. (2023) where applicable.

MIMIC-III-CLEAN Discharge summaries from MIMIC-III, using the cleaning and admission-level split. We keep only codes that appear at least ten times in the training split. We also drop the 21 admissions that overlap with the MDACE prompt-tuning set. This reduces the test set from 8,734 to 8,713 admissions across Tables 3–5.

Dataset	#Codes	Train	Val	Test
MIMIC-III-CLEAN	3,681	38,401	5,577	8,713
MIMIC-IV	7,942	89,102	13,378	19,801
MDACE (dev)	411	89	—	—

Table 1: Datasets and split sizes (admissions). MIMIC-III-CLEAN and MIMIC-IV train/val splits are used for Stage 1 SFT training; their test splits evaluate the full pipeline. MDACE is used only for Stage 2 prompt tuning and operating-point selection, on admissions not seen by Stage 1.

	INDIST		OOD		FULL	
	$\lvert C\rvert$	N_{eval}	$\lvert C\rvert$	N_{eval}	$\lvert C\rvert$	N_{eval}
MIMIC-III	3,681	3,681	5,248	1,896	8,929	5,577
MIMIC-IV	7,942	7,935	18,251	5,972	26,193	13,907

Table 2: Label-space sizes ($|C|$) and macro-F1 denominators (N_{eval}; denominator as in §4.2) per evaluation condition. INDIST is the MIMIC-III-CLEAN classifier label set (Edin et al., 2023); OOD comprises codes appearing in the full dataset but outside that label set; FULL is their union. Test splits: MDACE-filtered MIMIC-III-CLEAN (n=8,713) and full MIMIC-IV (n=19,801); see Table 1.

MIMIC-IV Discharge summaries with the ICD-10 split. The code space is larger and more granular than MIMIC-III-CLEAN: FULL covers 26,193 codes and OOD covers 18,251.

MDACE (Stage 2 development only.) Cheng et al. (2023)'s coder-grade ICD-9 corpus with explicit evidence spans. We use it only for Stage 2 prompt tuning and operating-point selection (Appendix H). The prompt-tuning subset is restricted to admissions that do not appear in the MIMIC-III-CLEAN training set.

InDist versus OOD. We evaluate each test set under three label-space conditions: INDIST, restricted to codes in the SFT classifier's label set; OOD, restricted to gold codes outside that label set; and FULL, their union. Table 2 reports both the prediction space ($|C|$, used as the micro-F1 label space) and the macro-F1 denominator (N_{eval}, the subset of $|C|$ with at least one test positive; see §4.2). A discriminative-only classifier cannot predict any OOD code by construction, so its FULL score is upper-bounded by what it achieves on INDIST, which is a ceiling that the hybrid pipeline is designed to break.

4.2 Evaluation Protocol

We report micro- and macro-F1 using the corrected formulation of Edin et al. (2023) – mean of per-label F1 rather than F1 of averaged precision and recall – which roughly doubles macro scores relative to the older convention and is the protocol used by recent PLM-ICD-family work. Following Edin et al. (2023), macro-F1 is averaged over the codes with at least one positive instance in the corresponding test split (N_{eval}, reported alongside each macro cell), since per-code F1 is undefined for codes with zero test support; micro-F1 is computed over the full prediction label space, so spurious predictions on zero-support codes still penalise precision. We report both micro- and macro-F1 for each of the INDIST, OOD, and FULL scopes (Section 5), with the mean number of predicted codes per note as a calibration check. All comparisons are at the admission level on the same Edin et al. splits; numbers from prior work are taken from cited papers when measured under this protocol and re-evaluated otherwise. Precision/recall decompositions and ranking metrics are in Appendix K.

4.3 Baselines and Ablations

External systems. We anchor against published discriminative classifiers (PLM-ICD (Huang et al., 2022), PLM-CA (Edin et al., 2024), GoM-ICD (Wu et al., 2025)). Non-comparable cells are marked "–".

Internal ablations. The SFT classifier is reported at two operating points: $\tau^{\star}_{\text{macro}}$ (macro-optimal) and $\tau^{\star}_{\text{micro}}$ (micro-optimal, also the band's upper edge θ_{high}). Both are derived from the SFT classifier's validation curve (Lipton et al., 2014). We use the same SFT classifier across every hybrid condition so that gains are attributable to Stage 2.

4.4 Implementation Details

Stage 1. We train one classifier checkpoint per dataset (ICD-9 on MIMIC-III-CLEAN, ICD-10 on MIMIC-IV) and reuse it across every hybrid condition. Inputs are left-truncated to 4096 tokens, the knee of the length–performance curve (Appendix B, Table 8). Full training recipe in Appendix A.

Stage 2. The frozen backbone introduced in §3.2 is GPT-OSS-20B; generation and constrained-decoding hyperparameters are in Appendix E, and cross-scale comparisons on MDACE are in Appendix F. Retrieval uses RRF with k=60 over BioLORD-2023-C (Remy et al., 2024) and `bm25s`; full configuration in Appendix D. For the band-verifier diagnostic of §6 we run an additional 10-variant prompt sweep on MDACE (Appendix G).

Cost and latency. Stage-2 inference adds roughly 30–50× wall-clock over SFT-only (≈1.6–1.7 s/note vs ≈31–56 ms); per-pass cost breakdown and Stage-1 training cost are in Appendix E.

5 Results

We report headline numbers (§5.1), oracle attribution (§5.2), stage-wise decomposition (§5.3), and a breakdown of where gains land and what is missed (§5.4).

5.1 Main results

Table 3 reports all conditions on the three label sets defined in Table 2. The MIMIC-III INDIST and FULL columns correspond to the classifier label set and the full vocabulary of Edin et al. (2023), respectively; the MIMIC-IV columns are the analogous ICD-10 classifier label set and the full MIMIC-IV vocabulary.

On INDIST macro-F1 the hybrid matches the macro-optimal SFT (Table 3; 0.331 vs 0.332, paired bootstrap $|\Delta|<0.001$, p=0.59). It is the only tested condition with non-zero F1 on OOD codes. The combined effect is a +0.038 FULL macro-F1 lift (95% CI [+0.034, +0.042], paired bootstrap $p<0.001$) at a 1.2-point FULL micro-F1 cost. We attribute the macro gain to long-tail recovery (§5.4) and the micro loss to the band verifier rather than the pipeline design (§5.2).

Our SFT alone at $\tau=\tau^{\star}_{\text{macro}}=0.125$ already exceeds the PLM-ICD reproduction of Edin et al. (2023) on INDIST macro-F1, so the hybrid's gains layer on a competitive classifier rather than complement a weak one.

5.2 Oracle ablation: where the bottleneck lives

Before decomposing pipeline stages, we ask where the closable headroom lives. Table 4 reports a sequential oracle ablation that starts from the SFT classifier at the macro-optimal threshold and walks up the band-verify / gap-fill ladder, replacing the LLM components with oracle decisions where indicated. Both oracles use the same rule: the oracle band verifier accepts a code in the verification band iff it is in the gold set, and the oracle gap-fill accepts a retrieved candidate iff it is in the gold

Dataset	Method	INDIST		OOD		FULL	
		micro	macro	micro	macro	micro	macro
MIMIC-III	PLM-ICD (Edin et al., 2023)	0.596[†]	0.266[†]	—	—	0.589[‡]	0.228[‡ §]
	PLM-CA (Edin et al., 2024)	—	—	—	—	0.600[‡]	0.247[‡]
	GoM-ICD (Wu et al., 2025)	**0.620**[†]	—	—	—	**0.617**[‡]	—
	SFT only ($\tau{=}\tau^{\star}_{\text{macro}}{=}0.125$)	0.556	**0.332**	—	—	0.552	0.219
	SFT only ($\tau{=}\tau^{\star}_{\text{micro}}{=}0.311$)	0.613	0.295	—	—	0.607	0.195
	Hybrid (ours)	0.551	0.331	**0.085**	**0.112**	0.540	**0.257**
MIMIC-IV (ICD-10)	PLM-ICD (Edin et al., 2023)	0.585	0.211	—	—	—	—
	GoM-ICD (Wu et al., 2025)	**0.613**	—	—	—	—	—
	SFT only ($\tau{=}\tau^{\star}_{\text{macro}}{=}0.123$)	0.539	**0.306**	—	—	0.534	0.174
	SFT only ($\tau{=}\tau^{\star}_{\text{micro}}{=}0.362$)	0.610	0.268	—	—	**0.602**	0.153
	Hybrid (ours)	0.555	0.302	**0.049**	**0.035**	0.545	**0.187**

Table 3: Main results; all values are F1. Point estimates only; 95% bootstrap CIs (1,000 admission-level resamples, seed 314) and full precision/recall decomposition are in Appendix K, Table 25. Headline contrasts pass paired bootstrap $p{<}0.001$. Macro-F1 denominator as in §4.2. $\tau{=}\tau^{\star}_{\text{macro}}$ and $\tau{=}\tau^{\star}_{\text{micro}}$ are the macro- and micro-F1-optimal global thresholds on the SFT validation split (Lipton et al., 2014); $\tau^{\star}_{\text{micro}}$ also equals θ_{high}. Per-dataset values: $(\tau^{\star}_{\text{macro}}, \tau^{\star}_{\text{micro}}) = (0.125, 0.311)$ for MIMIC-III, $(0.123, 0.362)$ for MIMIC-IV. [†]on MIMIC-III-CLEAN (same vocabulary as ours). [‡]on MIMIC-III full (Mullenbach et al., 2018), which shares vocabulary with our FULL column but uses a different test split, so the comparison is indicative. [§]Macro F1 reported here is the corrected arithmetic-mean formulation introduced by Edin et al. (2023), not the harmonic-mean formulation used in the original PLM-ICD paper (Huang et al., 2022).

Configuration	MIMIC-III	MIMIC-IV
SFT_MACRO	0.552	0.534
SFT_MICRO (SFT_HIGH)	0.607	0.602
+REAL_BV	0.581	0.571
+ORACLE_BV	**0.697**	**0.705**
+REAL_GF	0.540	0.545
+ORACLE_GF	**0.715**	**0.716**

Table 4: Oracle waterfall on FULL (micro-F1). SFT_MACRO / SFT_MICRO are the SFT-only operating-point baselines (SFT_MICRO=SFT_HIGH, the anchor for the pipeline rows). +REAL_BV adds the real band verifier; +ORACLE_BV substitutes the oracle band verifier (accept iff in gold). +REAL_GF adds real gap-fill on top of +REAL_BV (= full pipeline, reproduced from Tables 3–5); +ORACLE_GF substitutes oracle gap-fill on top of +ORACLE_BV, bounded by $K{=}5$ retrieval. Band verification is the binding constraint on pipeline micro-F1. Point estimates without bootstrap CIs.

set. The oracle gap-fill is bounded by the $K{=}5$ retrieved candidates per extracted term; it cannot recover gold codes that retrieval did not surface, so the oracle gap-fill number measures verification headroom given retrieval as fixed rather than the absolute ceiling on the gap branch.

The +REAL_BV → +ORACLE_BV transition (oracle band verifier) contributes +0.116 micro-F1 on MIMIC-III and +0.134 on MIMIC-IV, larger than every other delta; the +ORACLE_BV → +ORACLE_GF oracle gap-fill step adds a further +0.018 / +0.011. *The band verifier is the bottleneck on hybrid micro-F1; retrieval and selection are secondary.* Per-code attribution shows the same asymmetry. For every code the pipeline adds vs. SFT_HIGH, we label it an error if it is not in the gold set and a recovery if it is; the error-to-recovery ratio is the count of errors divided by the count of recoveries. The combined verifier-plus-gap-fill error-to-recovery ratio is 5.22:1 on MIMIC-III and 5.59:1 on MIMIC-IV, which is dataset-invariant within ±0.4 despite the ∼3× code-space gap (Appendix I). A prompt-variant sweep further shows the in-band false-accept rate is corpus-bound rather than prompt-bound: MDACE clusters in $[0.644, 0.710]$ across 10 variants while a 3-variant MIMIC-III replication spans $[0.169, 0.531]$ with non-transferring rank order (Appendix G, Table 13). We characterise the shape of the verifier constraint in §6.

5.3 Stage-wise decomposition

Table 5 reports cumulative pipeline outputs after each stage, anchored at SFT-only at $\tau{=}\tau^{\star}_{\text{micro}}$ (= θ_{high}, SFT_HIGH); Δ columns show each stage's marginal contribution to F1 (i.e., row i minus row $i{-}1$). The three sequential stages added to SFT_HIGH are:

- +BAND_VERIFY (Stage 2.1, LLM verification

Stage	micro	$\Delta\mu$	macro	ΔM
MIMIC-III FULL ($\lvert C\rvert$=8,929, N_{eval}=5,577)				
SFT_HIGH (τ=0.311)	0.607	—	0.195	—
+BAND_VERIFY	0.581	−0.026	0.222	+0.027
+GAP_FILL	0.373	−0.208	0.191	−0.031
+VERIFY_GAP_FILL	0.540	+0.167	0.257	+0.035
MIMIC-IV FULL ($\lvert C\rvert$=26,193, N_{eval}=13,907)				
SFT_HIGH (τ=0.362)	0.602	—	0.153	—
+BAND_VERIFY	0.571	−0.031	0.176	+0.023
+GAP_FILL	0.425	−0.146	0.158	−0.018
+VERIFY_GAP_FILL	0.545	+0.120	0.187	+0.012

Table 5: Stage-wise pipeline decomposition; all values are F1 (μ=micro, M=macro). Point estimates only; 95% bootstrap CIs in Appendix K, Table 25. All stage deltas reach paired-bootstrap $p<0.0001$ except the intermediate +GAP_FILL row, which reports the raw top-5 candidate set before candidate verification (no paired-bootstrap test). Each row is the cumulative output of SFT_HIGH plus every named stage above it (anchored at SFT-only at $\tau=\tau^{\star}_{\text{micro}}=\theta_{\text{high}}$, i.e. the upper edge of the principled band of Lipton et al., 2014); Δ columns report per-stage marginal contribution vs the row immediately above.

of predictions in $[\theta_{\text{low}}, \theta_{\text{high}})$);

- +GAP_FILL (Stage 2.2–2.3, retrieval-based candidate generation for codes not yet predicted);
- +VERIFY_GAP_FILL (Stage 2.4, per-(term, code) LLM verification of retrieved candidates).

The decomposition (Table 5) shows a macro-positive role for BAND_VERIFY at the new principled anchor and a clean failure mode in GAP_FILL. BAND_VERIFY rescues macro-F1 at a small micro cost: the principled band routes a wider mid-confidence pool than a fixed θ_{high}=0.5 cutoff would (cf. Appendix H, Table 14), so the verifier admits a few more borderline false positives but recovers more long-tail true positives. GAP_FILL floods the prediction set with low-precision candidates; VERIFY_GAP_FILL recovers most of that loss, leaving a small net micro cost in exchange for a macro net gain. All other stage deltas reach $p<0.0001$ (paired bootstrap); the intermediate +GAP_FILL row is omitted by design (see caption).

Component-level failure-mode attribution (see §6). An ordering ablation that skips BAND_VERIFY entirely (Appendix H, Table 15) leaves OOD micro/macro statistically unchanged ($|\Delta|\leq 0.005$) while moving in-distribution macro-F1 by -0.025 on removal, attributing the head/tail trade-off to BAND_VERIFY and OOD coverage to candidate-verify. Combining with the oracle ablation (§5.2), this indicates the headroom gap can be closed by improving BAND_VERIFY.

Stage effects are also *chapter-specific* and asymmetric across diagnoses and procedures. Procedure chapters show much larger gap-fill swings than diagnoses, which VERIFY_GAP_FILL only partially recovers: *Op. Urinary System* ends net-negative ($\Delta_{\text{tot}}=-0.380$) and *Op. Nose / Mouth / Pharynx* is the only net-positive procedure chapter (+0.106); on the diagnosis side, *Perinatal Period* shows the deepest gap-fill swing ($-0.686 \rightarrow +0.494$). Per-chapter heatmaps with bootstrap CIs for all 32 kept chapters are in Appendix J (Figures 2, 3).

5.4 Where are the gains, and what is missed

Where are the gains. Binning codes by training-set occurrence (Edin et al., 2023), the SFT classifier at $\tau=\tau^{\star}_{\text{micro}}$ shows a $\sim$3× head-to-tail code-level micro-F1 drop (0.72→0.22, recall 0.74→0.14). The hybrid roughly doubles tail recall (0.14→0.28), lifting Edin tail-bin macro-F1 from 0.13 to 0.20 (Edin's original 1–10 tail bin; the wider rank->1,000 tail of Appendix J gives 0.23→0.27), which is the source of its macro gain. Only gap-fill reaches the MIMIC-III OOD codes (Table 2), at 0.085 micro / 0.112 macro. OOD recovery concentrates in chapters with stable terminology; per-bin and per-chapter breakdowns are in Appendix J.

Search-miss diagnosis. On MDACE's 89-note span subset, $125/177 = 70.6\%$ of false-negative gold codes (Wilson 95% CI $[63.5\%, 76.8\%]$) are *terminology mismatches* where the correct code is present in the note under a different surface form, with hierarchy mismatch (20.3%) and ambiguity (9.0%) accounting for the remainder (Appendix H, Table 18). On the same subset, the LLM's evidence quote falls inside MDACE's annotated spans 6.2% of the time when gap-fill adds a correct code, versus 5.3% when the added code is wrong, a 1.2× separation (Table 19).

6 Discussion

What is the bottleneck in hybrid ICD coding? The dominant bottleneck is band verification. Of the three failure modes the oracle ablation localises, it carries by far the largest closable headroom on hybrid micro-F1 (Table 4: +0.116 MIMIC-

III, $+0.134$ MIMIC-IV), which is larger than retrieval flooding and terminology mismatch combined. More telling is *why* the gap is hard to close from prompting alone: the within-band false-accept rate is not a property of LLM verification in general but of what each corpus treats as the gold standard. A 10-variant MDACE sweep clusters tightly in false-accept rate $[0.644, 0.710]$; a 3-variant MIMIC-III replication spans $[0.169, 0.531]$ with rank order that does not transfer (Appendix G, Table 13). Evidence-span MDACE and billing-code MIMIC-III impose different base rates on an evidence-driven verifier. The productive next intervention is therefore supervision-side, not prompt-side. A learned per-code verification head trained on band-conditioned data would, by construction, inherit each corpus's base rate; we leave that to future work. Its natural complement, per-class thresholds outside the band, is directly testable: a top-50 MIMIC-III pilot recovers $+0.047$ top-50 macro-F1 at no LLM cost (Appendix H, Table 17). It is an instance of the threshold-tuning argument of Lipton et al. (2014) that is complementary to, not in competition with, the band verifier's role on borderline mid-confidence codes.

Does scale solve the problem? No. On MDACE at the principled band, scaling GPT-OSS $6\times$ (20B$\rightarrow$120B) yields $+0.016$ micro-F1 but -0.026 macro-F1, and GPT-OSS-20B remains the best macro-F1 point in the table despite being the smallest backbone tested (Appendix F, Table 11). Capacity is not the binding constraint: the bigger model still inherits the wrong corpus base rate, consistent with the MDACE-vs-MIMIC-III operating-point split (§6). For the verification step here, swapping in a stronger general-purpose LLM is not the productive next intervention.

What does this say about LLM-augmented coding more broadly? Each failure mode sits in a different part of the pipeline and demands a different intervention. Gap-fill flooding is a *retrieval-side precision* problem, as shown by the combined error-to-recovery ratio being close across MIMIC-III and MIMIC-IV (§5.2, Appendix I), consistent with open-set retrieval behaviour pending a third-corpus replication (see Limitations, *Empirical scope*); the skip-band ablation (Appendix H, Table 15) further shows that removing band-verify leaves OOD coverage untouched ($|\Delta| \leq 0.005$), so candidate-verify on retrieved gap-fill candidates is what carries the vocabulary-shift coverage. Terminology mismatch is a separate *retrieval-side recall* problem: 70.6% of MDACE false-negative gold codes are present in the note under a different surface form (§5.4), addressable by dense or hybrid matching against MRCONSO synonym pools, building on biomedical entity-linking work (Liu et al., 2021; Remy et al., 2024). Band verification, as argued above, is *supervision-side*: a dense-retrieval upgrade does not help the verifier's base-rate problem, and a per-code verification head does not help the retriever surface the right candidate strings. The implication is that progress on LLM-augmented coding is more likely to come from *component-specialised* models, such as a retrieval encoder fit to clinical terminology, a verification head fit to a corpus's gold-construction protocol, than from a stronger general-purpose model wired into a fully agentic loop (Motzfeldt et al., 2025).

7 Conclusion

We frame the hybrid pipeline as a diagnostic study of where LLM-augmented ICD coding succeeds and fails. Two headline results: the pipeline matches the macro-optimal SFT on MIMIC-III in-distribution codes ($\Delta_{\text{macro}}=-0.001$, $p=0.53$) and sits within 0.4 macro-F1 points on MIMIC-IV ($\Delta_{\text{macro}}=-0.004$, $p<0.001$), while lifting macro-F1 on FULL by $+0.038$ (95% CI $[+0.034, +0.042]$) on MIMIC-III and $+0.013$ ($[+0.011, +0.015]$) on MIMIC-IV, at a 1.2-point micro-F1 cost on MIMIC-III and no micro cost on MIMIC-IV ($+0.011$ vs the same baseline); and it is the only tested condition with non-zero F1 on the test-supported OOD codes (0.085 micro / 0.112 macro on MIMIC-III; 0.049 micro / 0.035 macro on MIMIC-IV), a within-codebook vocabulary-shift proxy for codebook evolution rather than deployment-grade performance.

Treated as a diagnostic study, the pipeline localises the three failure modes identified in §6. The next intervention is replacing the global LLM band verifier with a learned per-code verification head, paired with dense MRCONSO-synonym retrieval and per-class thresholds outside the band.

Limitations

Empirical scope. Our evaluation rests on MIMIC-derived datasets from a single US academic medical centre; transfer to other clinical corpora with different hospital systems, non-US jurisdictions, or note types other than discharge

summaries remains to be validated, and a third corpus with a distinct gold-construction protocol (e.g., an ICD-10-AM–coded source) is missing. The MDACE dataset is small (89 admissions not included in the MIMIC-III-CLEAN training set, single-annotator gold spans). Per-code test support is single-digit for many of the MIMIC-III-CLEAN OOD tail codes that enter the macro-F1 denominator (§4.2). Compute and time constraints precluded multi-seed runs of the full Stage-1+Stage-2 pipeline, so seed-to-seed variance of the Stage-2 LLM components is not characterised; evaluation throughout treats MIMIC gold codes as ground truth despite known annotation noise, and we did not commission a clinical-coder review of pipeline outputs (in particular gap-fill recoveries) against an independent human gold standard. The paired before/after edition swap that would directly validate the codebook-evolution claim is not run (see *Codebook-evolution claim* below).

Verifier trade-off frontier. The oracle headroom on band verification (Table 4) is robust across both datasets, but the within-band false-accept characterisation rests on a single 3-variant MIMIC-III replication (Table 13) alongside a 10-variant MDACE sweep (Appendix G); a third corpus with a different gold-construction protocol, multi-seed Stage-2 reruns, and a clinical-coder review of band-verified outputs are missing (see §6).

Codebook-evolution claim is architectural and proxy-validated. We do not run the paired before/after edition swap (e.g., ICD-10-CM 2019 $\rightarrow$ 2023) that would constitute end-to-end empirical validation of the codebook-evolution claim (§7); the MIMIC-III OOD evaluation is a within-codebook vocabulary-shift proxy that retains the original codebook's terminological conventions. Absolute F1 on the OOD subset is small: on MIMIC-III the hybrid scores 0.085 micro / 0.041 macro under the full-label denominator, equivalently 0.085 micro / 0.112 macro under the scorable denominator of Table 3 (the micro number is invariant to the macro denominator; cf. §4.2). The result establishes that the architectural property holds in the proxy regime, not that absolute performance on edition-introduced codes will be deployment-grade.

Stage-2 design dependencies. Several Stage-2 design choices encode assumptions we do not fully characterise. Substring verification of the gap analyst's `evidence_quote` eliminates outright fabrication but admits false-positive matches on common phrases (*history of*, *chronic*) and does not verify that the matched span semantically supports the proposed code. The mutual-exclusivity constraint at candidate verification (at most one confirmed code per extracted term) is appropriate for competing near-synonyms returned by retrieval but can over-prune legitimate etiology/manifestation pairs or combination codes sharing a surface term; mutex over-suppresses genitourinary co-occurrence (Appendix H, Table 16). Guideline injection covers only US-issued manuals (ICD-9-CM, ICD-10-CM, ICD-10-PCS Official Guidelines for Coding and Reporting); non-US variants such as the Australian Coding Standards for ICD-10-AM are not represented, and within-codebook advisory updates (e.g., American Hospital Association (AHA) *Coding Clinic*) are not ingested. Threshold values are read from a single SFT validation curve; per-class threshold variation across the full vocabulary is not characterised (cf. batch-composition dependence noted by Lipton et al. (2014)). The per-class pilot of Appendix H (Table 17) confirms head-code macro-F1 has measurable room ($+0.047$ on top-50 at no overall cost), and per-code tuning is left to future work alongside per-code verification.

Ethical Considerations

Data provenance and access. Both MIMIC-III and MIMIC-IV are de-identified clinical corpora released under the PhysioNet credentialed-access data-use agreement. All researchers working with these data on this project hold current PhysioNet credentials and have completed the required CITI Data or Specimens Only Research training. The associated discharge-summary text and ICD code annotations were accessed exclusively via the PhysioNet-managed distribution channel. The MDACE evidence-span overlay (Cheng et al. 2023) is similarly access-controlled. The study was reviewed and approved by the relevant institutional human research ethics committee, with approval covering secondary use of the MIMIC corpora for automated coding research; we omit the protocol identifier here to preserve anonymity for review and will add it at camera-ready.

Potential downstream harms. ICD coding sits directly in the reimbursement pipeline. Erroneous codes, whether under-coded (revenue loss) or over-coded (potential billing fraud), carry material con-

sequences for patients, providers, and payers. Aggregated ICD-coded claims also feed public health surveillance and population-level epidemiological research (disease incidence estimation, health-system resource planning, cohort identification for secondary research); systematic mis-coding patterns introduced by an automated tool would propagate into those downstream analyses and bias them in ways that are hard to detect without ground-truth re-audit. Our pipeline is positioned as a *diagnostic study* of where hybrid SFT–LLM coding succeeds and fails (§6); we explicitly argue against treating its OOD performance as deployment-grade (§7, Limitations). Any deployment in a live billing or surveillance workflow would require, at minimum: (i) clinical-coder review of every machine-suggested code prior to submission, (ii) prospective evaluation on the deploying institution's own note distribution and coding protocol, and (iii) ongoing audit against the post-hoc gold standard. We do not endorse use as an autonomous billing system.

LLM-specific risks in the clinical setting. The Stage-2 LLM components (band verifier, gap analyst, candidate verifier) can hallucinate. Substring verification of the gap analyst's `evidence_quote` eliminates outright fabrication of evidence spans but cannot guarantee that the matched span semantically supports the proposed code (also flagged in Limitations). For any clinical deployment, the human-in-the-loop coder remains the load-bearing safeguard; the LLM components should not be trusted as autonomous decision-makers.

Compute footprint. A single full-pipeline run on MIMIC-III FULL test (8,713 admissions) at the GPT-OSS-20B operating point uses approximately 1.6 wall-clock seconds per note (single H200, 96 in-flight requests), i.e. roughly 4 GPU-hours per evaluation pass. The Stage-1 SFT training run uses $\approx$ 26 H200 GPU-hours per dataset checkpoint. Total project compute, including ablations and prompt sweeps, sits in the low-thousands of H200 GPU-hours; we recognise this is material and have released the full Stage-1 LoRA adapter checkpoints (under the MIMIC data-use agreement, gated to PhysioNet-credentialed users; Appendix E) so that future readers can rerun the Stage-2 pipeline without repeating Stage-1 training.

Disclosure of AI assistance. Per the ACL policy on generative AI in authorship: a large language model assistant (Anthropic Claude) was used in this project for (i) sentence-level copy editing and style-guide conformance checks during manuscript preparation; (ii) code review and optimisation suggestions for the codebase; and (iii) iterative gap-analyst prompt refinement following the autoresearch protocol of Karpathy (2026), with nine accepted iterations selected on development-split macro-F1 (Appendix H). All experimental designs, dataset and metric choices, claims, and writing decisions are the authors' own; every generated suggestion (including each prompt iteration) was reviewed and accepted by a human author against pre-specified acceptance criteria, and final prompts were frozen before evaluation on held-out splits. The Responsible NLP Research Checklist filed with this submission contains the corresponding disclosure; at camera-ready this disclosure will also appear in the Acknowledgements section, per ARR policy.

References

Krishanu Das Baksi, Elijah Soba, John J Higgins, Ravi Saini, Jaden Wood, Jane Cook, Jack I Scott, Nirmala Pudota, Tim Weninger, Edward Bowen, and Sanmitra Bhattacharya. 2025. MedCodER: A generative AI assistant for medical coding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 449–459, Albuquerque, New Mexico. Association for Computational Linguistics.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online. Association for Computational Linguistics.

Centers for Medicare & Medicaid Services and National Center for Health Statistics. 2014. ICD-9-CM: International classification of diseases, ninth revision, clinical modification. `https://www.cdc.gov/nchs/icd/icd9cm.htm`. Final version; superseded by ICD-10-CM on 1 October 2015.

Hua Cheng, Rana Jafari, April Russell, Russell Klopfer, Edmond Lu, Benjamin Striner, and Matthew R. 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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7534–7550. Association for Computational Linguistics.

Hang Dong, Matús Falis, William Whiteley, Beatrice Alex, Shaoxiong Ji, Jiaoyan Chen, and Honghan Wu.

2022. Automated clinical coding: what, why, and where we are? *NPJ Digital Medicine*, 5.

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*, SIGIR '23, page 2572–2582, New York, NY, USA. Association for Computing Machinery.

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, Miami, Florida, USA. Association for Computational Linguistics.

Matúš Falis, Aryo Pradipta Gema, Hang Dong, Luke Daines, Siddharth Basetti, Michael Holder, Rose S Penfold, Alexandra Birch, and Beatrice Alex. 2024. Can gpt-3.5 generate and code discharge summaries? *Journal of the American Medical Informatics Association*, 31(10):2284–2293.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Andrej Karpathy. 2026. autoresearch: Ai agents running research on single-gpu nanochat training automatically. `https://github.com/karpathy/autoresearch`. GitHub repository.

Keith Kwan. 2024. Large language models are good medical coders, if provided with tools. *CoRR*, abs/2407.12849.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8180–8187.

Rumeng Li, Xun Wang, and Hong Yu. 2024. Exploring LLM multi-agents for ICD coding. *CoRR*, abs/2406.15363.

Rumeng Li, Xun Wang, and Hong Yu. 2025. Improving rare and common icd coding via a multi-agent llm-based approach. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 4945–4949, New York, NY, USA. Association for Computing Machinery.

Zachary C. Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. 2014. Thresholding classifiers to maximize F1 score. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. ArXiv:1402.1892.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. 2022. Hierarchical label-wise attention transformer model for explainable icd coding. *Journal of Biomedical Informatics*, 133:104161.

Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. 2023. Automated icd coding using extreme multi-label long text transformer-based models. *Artificial Intelligence in Medicine*, 144:102662.

Junyu Luo, Xiaochen Wang, Jiaqi Wang, Aofei Chang, Yaqing Wang, and Fenglong Ma. 2024. CoRelation: Boosting automatic ICD coding through contextualized code relation learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3997–4007, Torino, Italia. ELRA and ICCL.

Andreas Geert Motzfeldt, Joakim Edin, Casper L. Christensen, Christian Hardmeier, Lars Maaløe, and Anna Rogers. 2025. Code like humans: A multi-agent solution for medical coding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22612–22627, Suzhou, China. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

National Center for Health Statistics. 2024. ICD-10-CM: International classification of diseases, tenth revision, clinical modification. `https://www.cdc.gov/nchs/icd/icd-10-cm.htm`. Maintained by the National Center for Health Statistics (NCHS) and the Centers for Medicare & Medicaid Services (CMS); FY2025 version in effect October 1, 2024.

Sander Puts, Catharina M L Zegers, Andre Dekker, and Iñigo Bermejo. 2025. Developing an icd-10 coding assistant: Pilot study using roberta and gpt-4 for term extraction and description-based code selection. *JMIR Form Res*, 9:e60095.

François Remy, Kris Demuynck, and Thomas Demeester. 2024. Biolord-2023: semantic textual representations fusing large language models and clinical knowledge graph insights. *Journal of the American Medical Informatics Association*, 31(9):1844–1855.

Weiming Ren, Ruijing Zeng, Tongzi Wu, Tianshu Zhu, and Rahul G. Krishnan. 2022. Hicu: Leveraging hierarchy for curriculum learning in automated icd coding. In *Proceedings of the 7th Machine Learning for Healthcare Conference*, volume 182 of *Proceedings of Machine Learning Research*, pages 198–223. PMLR.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3132–3142, Brussels, Belgium. Association for Computational Linguistics.

Eduardo Martínez Rivera and Filippo Menolascina. 2025. Modernbert + colbert: Enhancing biomedical RAG through an advanced re-ranking retriever. *CoRR*, abs/2510.04757.

Congzheng Song, Shanghang Zhang, Najmeh Sadoughi, Pengtao Xie, and Eric Xing. 2021. Generalized zero-shot text classification for icd coding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Ali Soroush, Benjamin S. Glicksberg, Eyal Zimlichman, Yiftach Barash, Robert Freeman, Alexander W. Charney, Girish N. Nadkarni, and Eyal Klang. 2024. Large language models are poor medical coders — benchmarking of medical code querying. *NEJM AI*.

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, IJCAI-20*, pages 3335–3341. International Joint Conferences on Artificial Intelligence Organization. Main track.

Xindi Wang, Robert Mercer, and Frank Rudzicz. 2024. Multi-stage retrieve and re-rank model for automatic medical coding recommendation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4881–4891, Mexico City, Mexico. Association for Computational Linguistics.

World Health Organization. 2024. International statistical classification of diseases and related health problems (ICD). https://www.who.int/standards/classifications/classification-of-diseases. Covers ICD-9 (in effect 1979), ICD-10 (in effect 1993, latest version 2019), and ICD-11 (adopted WHA72.15, in effect 2022).

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.

Zhichao Yang, Sanjit Singh Batra, Joel Stremmel, and Eran Halperin. 2023a. Surpassing GPT-4 medical coding with a two-stage approach. *CoRR*, abs/2311.13735.

Zhichao Yang, Sunjae Kwon, Zonghai Yao, and Hong Yu. 2023b. Multi-label few-shot ICD coding as autoregressive generation with prompt. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 5366–5374. AAAI Press.

Zhichao Yang, Shufan Wang, Bhanu Pratap Singh Rawat, Avijit Mitra, and Hong Yu. 2022. Knowledge injected prompt based fine-tuning for multi-label few-shot ICD coding. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1767–1781, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Xu Zhang, Wenxin Ma, Chenxu Wu, Rongsheng Wang, Kun Zhang, and S. Kevin Zhou. 2026. From documents to spans: Code-centric learning for llm-based ICD coding. *CoRR*, abs/2603.15270.

Xu Zhang, Kun Zhang, Wenxin Ma, Rongsheng Wang, Chenxu Wu, Yingtai Li, and S Kevin Zhou. 2025. A general knowledge injection framework for ICD coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7180–7189, Vienna, Austria. Association for Computational Linguistics.

A SFT Discriminative Model: Hyperparameters and Infrastructure

Head architecture. We follow the PLM-CA head of Edin et al. (2024): given backbone hidden states $H \in \mathbb{R}^{T\times d}$ and learnable per-label queries $Q \in \mathbb{R}^{|\mathcal{L}|\times d}$, we compute $A = \mathrm{softmax}(QK^\top)$, $O = \mathrm{LayerNorm}(AV)$, and per-code logits $s = O\,\mathbf{w}_{\text{out}} + b_{\text{out}}$ with $K = HW_k$, $V = HW_v$, $\mathbf{w}_{\text{out}} \in \mathbb{R}^d$ shared across labels; the softmax scale

is fixed to 1 per Edin et al. The block is single-head at full backbone width d; W_k has no bias, W_v has a bias; LayerNorm is post-attention. Q, W_k, W_v, $\mathbf{w}_{\text{out}}$ are initialised from $\mathcal{N}(0, 0.03^2)$ with W_v and output biases zeroed; LoRA adapters use `peft` defaults.

Preprocessing. Discharge summaries are tokenised with the backbone's native tokenizer and left-truncated to $T = 4096$ tokens, preserving the tail of the note where coding-relevant content concentrates. Formatted laboratory result blocks are stripped prior to tokenisation.

Threshold tuning. Threshold tuning follows the procedure in §3.1. For ICD-9 this yields $(\theta_{\text{low}}, \theta_{\text{high}}) = (0.125, 0.311)$; for ICD-10, $(0.123, 0.362)$. No tuning is performed on the MDACE development split.

Infrastructure. Training runs on a 2×NVIDIA H200 GPU node with DDP data parallelism, PyTorch 2.11.0 (CUDA 12.9) and PyTorch Lightning 2.6.4. We enable bf16-true mixed precision and TF32 matmul kernels; Lightning's `deterministic=True` flag is set and all Python, NumPy, PyTorch, and worker RNGs are seeded for reproducibility. No gradient clipping is applied (Lightning default) and gradient checkpointing is disabled; the effective update batch of 64 is realised as per-device batch 4×2 GPUs $\times$ 8 accumulation steps. A full training run completes in approximately 26 H200 GPU-hours ($\approx$ 13 h wall-clock on the two-GPU configuration above).

Table 6 lists the complete hyperparameter configuration.

Seed policy. `set_global_seed(DEFAULT_SEED)` is invoked at the entry point of every Stage 1 and Stage 2 script, seeding Python `random`, NumPy, PyTorch (CPU + CUDA), the DataLoader worker init, and the vLLM sampler. `DEFAULT_SEED`= 314.

Software versions. Stage 1 (training) and Stage 2 (LLM pipeline and retrieval) share a single client-side environment: Python 3.12.1, PyTorch 2.11.0 (CUDA 12.9), PyTorch Lightning 2.6.4, `transformers` 5.9.0, `peft` 0.19.1, `openai` 2.37.0 (Python client to the vLLM OpenAI-compatible endpoint), `bm25s` 0.3.9. The vLLM server runs separately at version 0.19. The full pinned dependency graph is released as `uv.lock` (main pipeline) and `eval/uv.lock` (reproduction notebook).

Hyperparameter	**Value**
Data	
Sequence length T	4096
Truncation side	left
Optimisation	
Optimiser	AdamW
Peak learning rate	5×10^{-4}
Min learning rate	peak/10
Schedule	OneCycleLR, cosine
Warm-up fraction	0.1
(β_1, β_2)	(0.9, 0.95)
Weight decay	10^{-3} (on 2D params only)
Effective batch size	64
Max epochs	15
Early-stopping patience	3 (val. loss)
Precision	BF16
Loss	per-label BCE
Head dropout	0.1
LoRA	
Rank r	32
α	16
Dropout	0.1
Bias	none
Target modules	$\{q, k, v, o, \text{up}, \text{down}\}$_proj
Reproducibility	
Random seed	314
GPUs	2× NVIDIA H200
Framework	PyTorch Lightning 2.6.4

Table 6: Stage 1 training configuration.

B SFT Backbone Scaling Sweep on MIMIC-III-CLEAN

F1 plateaus around the 1.7B–4B range and regresses at 8B; 4B leads on the micro-optimal micro-F1 (0.613) and matches 1.7B on the macro-optimal macro-F1 (0.332), with 1.7B narrowly leading the other two cells, consistent with the observation of Motzfeldt et al. (2025) that scale alone does not cure long-tail memorisation. We adopt 4B as Stage 1 since the gap to 1.7B at the cells where 1.7B leads is $\leq$0.021 while 4B's micro-optimal micro-F1 is the operating point used by the hybrid pipeline. At identical parameter count `-Base` variants beat Instruct on macro-F1 at every size we tested. Our Qwen3-4B-Base SFT reaches micro-F1 0.613 on the MIMIC-III-CLEAN test set reported in Table 3, within 0.007 of GoM-ICD's reported 2025 discriminative SOTA on the same split.

Micro-F1 climbs steeply from 1024 to 4096 tokens (+12.9 points) but plateaus thereafter, with 6144 yielding only a marginal +1.0 point gain over 4096 at roughly twice the compute, suggesting 4096 sits near the knee of the length–performance curve for MIMIC-III discharge summaries.

Backbone	#LoRA	θ^*_{macro}: mi/ma	θ^*_{micro}: mi/ma
Qwen3-0.6B	22.4M	0.554 / 0.320	0.602 / 0.289
Qwen3-1.7B	43.5M	**0.577** / **0.332**	0.609 / **0.307**
Qwen3-4B	74.4M	0.556 / **0.332**	**0.613** / 0.295
Qwen3-8B	117M	0.508 / 0.312	0.601 / 0.272
GoM-ICD	–	–	0.620 / –

Table 7: Stage-1 SFT results on MIMIC-III-CLEAN (8,734 test notes, 3,681 codes). LoRA $r = 32$, lr 5×10^{-4}, $T = 4096$, left-truncation, lab results stripped. GoM-ICD row is Wu et al. (2025)'s reported micro-F1 on the same split for context.

Seq. Len.	Micro			Macro		
	P	R	F1	P	R	F1
1024	0.511	0.460	0.484	0.245	0.193	0.193
2048	0.584	0.535	0.559	0.294	0.257	0.250
4096	0.635	0.592	0.613	0.345	0.298	0.296
6144	**0.643**	**0.604**	**0.623**	**0.352**	**0.311**	**0.307**

Table 8: Effect of input sequence length on Qwen3-4B-Base PLM-CA on MIMIC-III-CLEAN. All runs use left truncation with lab results removed; thresholds tuned for micro-F1.

C Example Prompts and Schemas

This appendix documents the gap-analyst prompt and the two Pydantic schemas (`GapAnalysisResult`, `VerificationResponse`) used for constrained JSON decoding across the Stage-2 LLM pipeline. The full prompt suite for the three Stage-2 sub-stages: gap analyst, band verifier, and top-N verifier, together with the shared coding-guideline fragments injected into them, is released verbatim in the accompanying codebase. Prompt fragments are parameterised by code system; the gap-analyst skeleton below shows the ICD-9-CM injection points used for the MIMIC-III experiments, with ICD-10-CM variants structurally identical. Unicode right-arrows (→) in the source prompts are rendered as `->` in the listings.

Final prediction. The predicted code set for a note is

$$\hat{Y} \;=\; \{c : \sigma(s_c) \geq \theta_{\text{high}}\} \;\cup\; Y_{\text{band}} \;\cup\; Y_{\text{gap}},$$

where Y_{band} are the band-verified codes confirmed in §3.2 and Y_{gap} are the gap-retrieved codes confirmed by the candidate verifier of §3.2.

Gap analyst

The system prompt is shown with f-string interpolation points retained as `{...}` markers.

System prompt (structural skeleton). The system prompt opens by framing the role (clinical coding gap analyst given a discharge summary and an existing code list) and then walks through the following sections; only the headings are reproduced here, each with a two-sentence summary of intent.

```
CRITICAL GOALS
  Maximise recall of clinically relevant
  diagnoses and procedures documented in the
  note but not already covered by an assigned
  code. Every item must carry auditable
  evidence (exact quoted span + section), use
  the most specific term as written, and pay
  special attention to rare or atypical
  conditions that the upstream classifier
  systematically under-codes.

HOW TO DECIDE "MISSING"
  For each candidate, retain it unless the
  assigned codes already include an exact code
  for the same condition; risk factors and
  upstream conditions are always coded
  separately from their sequelae. Code-system-
  specific combination and etiology/manifestation
  rules are injected by code_type, and when in
  doubt the item is kept for the downstream
  verifier to decide.

WHAT TO SCAN
  Past Medical History / active chronic
  conditions and Social History sections are
  enumerated as required scan targets, including
  named common comorbidities (hypertension,
  diabetes, hyperlipidemia, tobacco use, etc.)
  that must be listed as gaps if not already
  explicitly coded.

INCLUSION RULES
  Code-system-specific diagnoses and procedures
  rule blocks, plus shared DO_NOT_INVENT,
  EVIDENCE_REQUIREMENTS, and DEDUPLICATION
  fragments, are injected here. These constrain
  what counts as an extractable item and forbid
  fabrication of quotes or codes not supported
  by the note.

UNCERTAINTY HANDLING
  Under uncertainty the policy is to favour
  recall and list the item, never to fabricate
  evidence; if an exact substring cannot be
  located, the closest containing phrase is used
  without paraphrasing. The model is instructed
  not to stall or repeat reasoning.

OUTPUT FORMAT (STRICT)
  Return JSON only (no markdown, no commentary)
  with exactly two top-level keys, "diagnoses"
  and "procedures"; empty arrays are returned
  when no gaps are found. Code-system-specific
  normalisation rules and the output schema are
  appended to fix field names and value
  conventions.
```

The full system prompt is released verbatim in the accompanying codebase.

User prompt.

```
Review the following discharge summary and the
list of ICD codes already assigned.
Identify any diagnoses or procedures documented
in the note that are NOT covered by the
assigned codes.

<ASSIGNED_CODES>
{code_list}
</ASSIGNED_CODES>

<DISCHARGE_SUMMARY>
{note}
</DISCHARGE_SUMMARY>
```

where `{code_list}` is the assigned codes formatted one per line as `"i. CODE - DESCRIPTION"` and `{note}` is the raw discharge-summary text.

Output schemas

Both Stage-2 verifier sub-stages share the `VerificationResponse` schema; the gap-analyst sub-stage uses `GapAnalysisResult`.

`GapAnalysisResult`.

```
class Evidence(BaseModel):
    quote: str
    section: str

class Diagnosis(BaseModel):
    name: str
    normalized: str
    evidence: list[Evidence]

class Procedure(BaseModel):
    name: str
    normalized: str
    evidence: list[Evidence]

class GapAnalysisResult(BaseModel):
    diagnoses: list[Diagnosis] = \
        Field(default_factory=list)
    procedures: list[Procedure] = \
        Field(default_factory=list)
```

`VerificationResponse`. Used by both Stage-2 verifier sub-stages (band verifier and top-N verifier); the top-N "at most one verdict marked Confirmed" constraint is enforced at the prompt level rather than in the schema.

```
class CodeVerdict(BaseModel):
    code: str
    evidence_quote: str
    contradiction_quote: str | None = None
    decision: str  # "Confirmed" or "Rule out"
    correct: bool | None = None

class VerificationResponse(BaseModel):
    verdicts: list[CodeVerdict]
```

D Retrieval Recipe

Code corpus. The retrievable code list is the union of (i) the ICD-9-CM and ICD-9-PCS short and long titles for ICD-9 runs, or ICD-10-CM and ICD-10-PCS titles for ICD-10 runs, and (ii) UMLS `MRCONSO.RRF` string variants whose source vocabulary tag matches the active code system and whose language tag is `ENG`. Diagnoses and procedures are indexed separately. For ICD-9, the expanded corpus is 87,990 entries for diagnoses (14,567 titles plus 73,423 synonyms) and 20,963 entries for procedures (3,882 titles plus 17,081 synonyms). For ICD-10, the diagnosis corpus is 231,737 entries (95,109 titles plus 136,628 synonyms); the procedure corpus is 216,179 entries (81,369 titles plus 134,810 synonyms).

Dense encoder. Code text and extracted query terms are embedded with `FremyCompany/BioLORD-2023-C` (Remy et al., 2024) using CLS pooling over the final hidden states followed by L2 normalisation.

Lexical index. BM25 indexing uses the `bm25s` library with default English tokenisation and Porter stemming applied to both index-time and query-time strings. The lexical index is built once over the same expanded corpus.

Fusion. Per query term, we run dense top-20 and BM25 top-20 retrieval independently and combine via Reciprocal Rank Fusion with $k{=}60$:

$$\mathrm{RRF}(d) = \sum_{r \in \{r_{\text{dense}}(d), r_{\text{bm25}}(d)\}} \frac{1}{k+r}.$$

We then keep the top $K{=}5$ fused candidates per term subject to RRF_score ≥ 0.015.

Source-abbreviation filters. `MRCONSO` rows are first filtered to `LAT=ENG`, then matched against a per-edition, per-code-type `SAB` whitelist: {`ICD9CM`, `MTHICD9`} for both ICD-9 diagnoses and procedures, {`ICD10CM`} for ICD-10 diagnoses, and {`ICD10PCS`} for ICD-10 procedures. Each ICD code is joined to its UMLS CUI via this whitelist, after which we additionally pull strings from a fixed cross-ontology set {`SNOMEDCT_US`, `MSH`, `NCI`, `CSP`, `AOD`, `CHV`, `MEDLINEPLUS`, `NDFRT`, `RXNORM`} for any row sharing the CUI; this CUI-bridged step surfaces clinical and consumer-oriented surface forms that the ICD source vocabularies alone do not contain. Per code we retain at most 10 synonyms (deduplicated against the long title), giving the corpus sizes reported above. The narrow primary whitelist preserves coding fidelity (no other vocabularies set the canonical code-string mapping); the broader CUI-bridge whitelist captures lexical variation while excluding noisy or non-clinical sources.

Hyperparameter	Value
Band thresholds	
$\theta_{\text{low}} = \tau^{\star}_{\text{macro}}$	0.125 (MIMIC-III) / 0.123 (MIMIC-IV)
$\theta_{\text{high}} = \tau^{\star}_{\text{micro}}$	0.311 (MIMIC-III) / 0.362 (MIMIC-IV)
LLM serving	
Backbone	GPT-OSS-20B
Quantization	MXFP4 (upstream-native)
Server	vLLM 0.19 (OpenAI-compatible)
`reasoning_effort`	`low`
Concurrency cap	96 in-flight
JSON-schema decoding	enabled
Sampling	
Temperature	0.0
`top_p`, `top_k`	unset (vLLM defaults)
`max_tokens`	unset (no output cap)
`repetition_penalty`	1.1 (gap analyst only)
Sampling seed	314 (bumped on retry)
Band verifier	
`chunk_size`	20 codes per LLM call
Retrieval	
Dense encoder	BioLORD-2023-C
Lexical index	`bm25s`, Porter stem
RRF k	60
Top-K per term	5
min_score floor	0.015
Candidate verifier	
`top_n`	5
Mutual exclusivity	enforced

Table 9: Stage-2 default hyperparameters. Sweep ranges and ablations are in Appendix H.

Caching. Dense embeddings, BM25 index, and per-term retrieval outputs are all cached on disk keyed by code-system + corpus hash to make Stage 2 ablations cheap to re-run.

Corpus and model versions. UMLS Metathesaurus release 2025AB (`MRCONSO.RRF`); ICD-9-CM / ICD-9-PCS FY 2011 (the last edition prior to the US ICD-10 transition, covering the MIMIC-III discharge window); ICD-10-CM / ICD-10-PCS FY 2022 for MIMIC-IV; `FremyCompany/BioLORD-2023-C`.

E Stage-2 Hyperparameters

Stage-1 hyperparameters are listed in Table 6 (Appendix A). Stage-2 defaults are summarised in Table 9.

Cost and latency. Per-pass breakdown is in Table 10. Wall-clock cost is as reported in §4.4.

Released artifacts. Code, prompts, Pydantic schemas, retrieval-index build scripts, and the DVC pipeline (`dvc.yaml`, `params.yaml`) are released at an anonymised GitHub repository: `https://anonymous.4open.science/r/hybrid-gap-0C56`.

Sub-pass	In tok./note	Out tok./note	Calls/note
MIMIC-III FULL *(n=8,734 notes)*			
band verify	6.4K	742	0.99
gap analysis	5.2K	1,012	1.00
candidate verify	40.0K	2,209	7.50
total	**51.7K**	**3,964**	**9.49**
MIMIC-IV FULL *(n=19,801 notes)*			
band verify	7.0K	918	1.00
gap analysis	5.7K	999	1.00
candidate verify	39.1K	2,160	7.07
total	**51.8K**	**4,076**	**9.06**

Table 10: Per-note Stage-2 LLM cost by sub-pass, GPT-OSS-20B + vLLM, summed over admissions with at least one Stage-2 call. Inference runs on the full pre-filter set (8,734 MIMIC-III / 19,801 MIMIC-IV admissions); the MIMIC-III metric tables of §5 use the MDACE-filtered 8,713-admission subset, so n here differs from Table 1 by design. *Calls/note* fractions arise because some admissions have no in-band codes (band verify) or no extracted gap terms (candidate verify). Retrieval (BioLORD-2023-C + `bm25s` RRF) is parquet-only and not an LLM call. End-to-end wall-clock is 1.6 s/note on MIMIC-III, 1.7 s/note on MIMIC-IV (single H200, 96 in-flight requests).

F LLM-Backbone Scale on the MDACE Full Pipeline

This appendix compares three LLM backbones for the entire Stage-2 pipeline (band verifier + gap analyst + top-N candidate verifier) on MDACE (val+test, 89 admissions, in-distribution vocabulary of 411 codes). All three runs reuse the same Stage-1 SFT checkpoint and operate at the principled band (Appendix A) of Lipton et al. (2014), with guideline injection on, `reasoning_effort=low`, and band-verifier `chunk_size=20`.

G Verifier Prompt-Variant Sweep

We ran 10 verifier prompt variants on MDACE in-distribution at the principled band (Appendix A) of Lipton et al. (2014), 308-note evaluation set, single seed. The variants span four classes of intervention: wording strictness, procedural-order inversion, schema reordering / framing, and architectural decoupling (LLM emits structured signals, deterministic Python policy decides).

Across all four intervention classes the in-band

Backbone	Micro-F1	Macro-F1
GPT-OSS-20B	0.659	**0.572**
Qwen3.6-27B-FP8	0.608	0.563
Qwen3.6-35B-A3B	0.645	0.545
GPT-OSS-120B	**0.675**	0.546

Table 11: LLM-backbone scale on the MDACE full pipeline at the principled band (Appendix A). Within the GPT-OSS family, scaling $6\times$ (20B$\rightarrow$120B) yields $+0.016$ micro-F1 but -0.026 macro-F1: the larger model's micro gain comes at the cost of rare-code coverage at this band. Qwen3.6-35B-A3B sits below both GPT-OSS variants on micro and macro, suggesting backbone family rather than parameter count is the dominant factor at this scale. Raising `reasoning_effort` from `low` to `medium` closes the 20B$\leftrightarrow$120B macro-F1 gap to within ± 0.001 (0.575 vs 0.576), reinforcing the family-not-scale reading.

		in-band		top-N	combined
ID	Class	fa_rate	fr_rate	fa_rate	macro-F1
V0	wording (lenient base)	0.681	**0.054**	0.865	**0.572**
V1	wording (strict)	0.667	0.091	0.872	0.567
V2	procedural	0.671	0.096	0.844	0.556
V3	architectural	0.698	0.152	0.827	0.526
V4	architectural	**0.644**	0.144	0.843	0.525
V5	framing	0.699	0.160	0.841	0.514
V6	signal-extractor	0.686	0.166	0.799	0.536
V7	framing	0.669	0.142	**0.793**	0.546
V8	framing	0.710	0.152	0.853	0.529
V9	procedural	0.695	0.150	0.795	0.533

Table 12: Verifier prompt-variant sweep on MDACE in-distribution at the principled band (Appendix A), 308-note evaluation set. Baseline (SFT-only above-band) macro-F1 = 0.339, identical across rows. fa_rate = FA/(FA+TR); fr_rate = FR/(FR+TA). All variants admit $\sim$2 false in-band accepts per true in-band accept; the ten in-band fa_rate values span $[0.644, 0.710]$ (6.6-point range), and no intervention class escapes this floor. All rows are single-seed point estimates under vLLM-backed sampling; differences $\lesssim 0.05$ fa_rate should be treated as within-noise. Macro-F1 denominator as in §4.2 (scorable and full-label denominators coincide on the MDACE in-distribution set).

fa_rate clusters in a 6.6-point band $[0.644, 0.710]$. V0 (lenient baseline) attains the highest combined macro-F1 (0.572) and the lowest in-band fr_rate (0.054): in this low-base-rate verification regime, accepting aggressively recovers more true positives than it admits false positives at the margin. The strictest in-band fa_rate (0.644, V4) is achieved by an architectural-decoupling variant but pays a 14.4% in-band fr_rate and -0.047 combined macro-F1 versus V0. The top-N candidate-set fa_rate is uniformly higher (0.79–0.87) across all variants, consistent with retrieval-side false-positive flooding (§6) being a separate failure mode from in-band over-acceptance.

MIMIC-III in-band replication. We replicate three of the ten MDACE variants (V0, V4, V6, chosen to bracket the MDACE fa_rate range) on a stratified 400-note in-band sample drawn from MIMIC-III FULL test (seed 314, gpt-oss-20b with `reasoning_effort=low`). Table 13 reports per-variant in-band code-level counts on the sample plus the population V0/production-prompt tally over the full MIMIC-III in-band verification run.

H Further Stage-2 Ablations: Verification, Gap-Fill, and Retrieval

Verification

Verification band width (sensitivity check). This sweep was originally used to select θ_{high}; under the principled framing of Lipton et al. (2014) the band endpoints are read off the SFT validation curve as $[\tau^{\star}_{\text{macro}}, \tau^{\star}_{\text{micro}}]$ rather than tuned on MDACE, so we now report the sweep only as a sensitivity check on the band's upper endpoint. Fixing $\theta_{\text{low}} = \tau^{\star}_{\text{macro}} = 0.125$ and sweeping $\theta_{\text{high}} \in \{0.2, \ldots, 0.9\}$ at 0.1 granularity, Table 14 reports SFT+verify micro/macro-F1 on MDACE at the principled lower endpoint. Micro-F1 is flat to within 0.013 across $\theta_{\text{high}} \in [0.3, 0.9]$ and macro-F1 peaks at $\theta_{\text{high}} = 0.3$, confirming that the upper endpoint is not load-bearing once θ_{low} is read off the SFT validation curve.

Verify-then-gap ordering (skip-band ablation). To test whether the verify-then-gap ordering contributes beyond just doing both stages, we re-ran Stage 2 on MIMIC-III FULL test with band-verify disabled: gap-analysis is conditioned directly on SFT_HIGH ($\sigma \geq \tau^{\star}_{\text{micro}}$=0.311), and candidate-verify then prunes the top-N retrieved candidates at the same operating point. Table 15 reports the per-scope deltas.

Mutual-exclusivity ablation. Candidate-verify enforces a mutual-exclusivity constraint at most one confirmed code per extracted term (§3.2). We re-ran candidate-verify on MIMIC-III FULL test with the constraint disabled, holding band-verify output constant; Table 16 reports the global and per-chapter deltas at τ=$\tau^{\star}_{\text{micro}}$=0.311.

Variant	TA	FA	TR	FR	fa_rate	fr_rate
MIMIC-III in-band, 400-note sample ($n_{verdicts}$=4,384)						
V0 lenient base	534	1,878	1,660	312	0.531	0.369
V4 arch. decoupl.	223	756	2,669	572	0.221	0.720
V6 signal-extr.	193	591	2,915	631	0.169	0.766
MIMIC-III in-band, population (V0/production, $n_{verdicts}$=91,756)						
V0 (population)	11,475	39,611	33,947	6,723	0.5385	0.3694
MDACE V0 (reference, Appendix G)						
V0 (MDACE)	—	—	—	—	0.681	0.054

Table 13: Verifier prompt-variant sweep on MIMIC-III in-band notes; V0 is identical to the production band-verify prompt (sha256[:8] = `b1a20cb1`). Sample fa_rate Wilson 95% CI $[0.514, 0.547]$ covers the population 0.5385 $[0.5349, 0.5421]$. Dispersion across V0/V4/V6 spans 0.362 on MIMIC-III ($\approx 5.5\times$ the MDACE 6.6-point range); rank order does not transfer (MDACE V0≈V6≈V4; MIMIC-III V0≫V4>V6). Absolute (fa_rate, fr_rate) values are *not* directly comparable across corpora: MDACE gold is evidence-span-anchored, MIMIC-III gold is billing-code, so an evidence-driven verifier sits at corpus-specific operating points by construction (see §6).

θ_{high}	Micro-F1	Macro-F1	Codes filtered ($\mu \pm \sigma$)
0.2	0.632	0.571	17.03 ± 8.48
0.3	0.660	**0.576**	15.80 ± 7.85
0.4	0.666	0.574	15.30 ± 7.35
0.5	0.666	0.565	14.90 ± 7.17
0.6	0.672	0.572	14.79 ± 7.15
0.7	**0.672**	0.569	14.51 ± 7.08
0.8	0.663	0.548	14.20 ± 6.90
0.9	0.662	0.550	14.19 ± 6.80

Table 14: Verification band-width sweep on MDACE with SFT+verify (GPT-OSS-20B, low reasoning, `chunk_size`= 20), holding $\theta_{\text{low}} = \tau^{\star}_{\text{macro}} = 0.125$ and varying θ_{high}. Average ground-truth codes per note: 10.70; "Codes filtered" is the per-note count of accepted predictions after verification. Reported as a sensitivity check; the principled upper endpoint $\tau^{\star}_{\text{micro}} = 0.311$ is read off the SFT validation curve, not selected from this sweep.

Scope	F1	full pipeline	skip-band	Δ
in-dist	micro	0.551	0.566	+0.016
in-dist	macro	0.331	0.306	−0.025
OOD	micro	0.085	0.079	−0.005
OOD	macro	0.112	0.113	+0.001
Full	micro	0.540	0.552	+0.012
Full	macro	0.257	0.241	−0.016

Table 15: Skip-band ablation on MIMIC-III FULL test (n=8,713 admissions, n_{codes}=8,929) at $\tau=\tau^{\star}_{\text{micro}}=0.311$: full pipeline vs Stage-2 with band-verify disabled (gap-analysis conditioned directly on SFT_HIGH). Band-verify trades a small in-dist macro precision gain (-0.025 macro on removal) for in-dist micro recall ($+0.016$ micro on removal); OOD micro/macro are statistically unchanged ($|\Delta| \leq 0.005$), localising OOD recovery to candidate-verify rather than band-verify.

Per-class threshold pilot. For the top-50 MIMIC-III ICD-9 codes by training-set frequency, we sweep per-class thresholds on the SFT validation split, pick the per-class F1-optimal, and re-evaluate on test. The non-head 3,631 codes keep the global threshold $\tau^{\star}_{\text{macro}}$=0.125. Table 17 reports the top-10 positive movers (test-set ΔF1) and the single regressor.

Gap-fill and prompt search

Prompt search. Following Karpathy (2026), an LLM agent proposes a single gap-analyst prompt modification per iteration, re-runs the full pipeline on the development split, and retains the change only if combined macro-F1 improves; we ran nine such iterations. All MIMIC-III results are reported on splits never seen during prompt search.

Other Stage-2 settings. We use `reasoning_effort=low`: Extract+search at `medium` gains marginally on MDACE at $\approx 5\times$ the token cost. We use `top_n=5` for gap-verify based on a sweep under the earlier gap-then-verify ordering, where `top_n=5` marginally outperformed `top_n=10`; the sweep has not been re-run under the current verify-SFT-first regime, and we treat the choice as fixed.

Retrieval grounding

Search-miss categories. Table 18 reports per-category counts for the 177 false-negative gold codes on the MDACE 89-note span subset. The subset is the 89 of MDACE's 308 admissions for which the Stage-2 pipeline has cached LLM artifacts (in-band verifier, gap-fill term extraction, top-N candidate retrieval, and candidate-verifier verdicts) under the GPT-OSS-20B prompt-tuning configuration. False-negatives are gold codes absent from the hybrid pipeline's final predictions on each note, where final predictions compose SFT-high at $\tau^{\star}_{\text{micro}}$=0.311 (the MIMIC-III SFT classifier applied to the MDACE notes) with the band-verifier-confirmed and candidate-verifier-confirmed code sets. Categories are assigned by a deterministic rule grounded in MDACE evidence spans: a FN gold code is *ambiguity* if present in the union of top-N retrieval candidates; *hierarchy mismatch* if absent from that union but another code sharing its 3-digit

Arm	micro-F1	macro-F1
baseline (with mutex)	0.5399	0.2566
ablation (no mutex)	0.5244	0.2550
Δ	-0.0155	-0.0016

Chapter (top-5 movers)	n_{codes}	baseline macro	Δ
PCS-5 Op. Nose / Mouth / Pharynx	95	0.1734	-0.0200
PCS-2 Op. Endocrine	32	0.1594	-0.0188
PCS-12 Op. Female Genital	78	0.1568	-0.0175
PCS-1 Op. Nervous	88	0.2297	-0.0110
ICD9-7 Circulatory	420	0.2292	-0.0092
Counter-direction (mutex over-suppresses):			
ICD9-10 Genitourinary	—	—	$+0.0152$
PCS-10 Op. Urinary	—	—	$+0.0087$

Table 16: Mutual-exclusivity ablation on MIMIC-III FULL test at $\tau{=}\tau^{\star}_{\text{micro}}{=}0.311$ (full vocab $n_{\text{codes}}{=}8{,}929$). Removing the mutex constraint costs 1.55 micro-F1 points at near-zero macro change, with a $+37\%$ inflation in verified-correct candidate volume; $31/36$ chapters move in the expected direction (worse without mutex). Mutex is a measured precision lever rather than a conjectured one. The two genitourinary/urinary chapters move in the opposite direction (mutex over-suppresses legitimate co-occurrences in that pair), surfaced as a calibration limitation rather than a rule-set refactor.

Code	$n_{\text{val pos}}$	τ_{pc}	test ΔF1
$V58.61$	445	0.397	$+0.1091$
37.23	263	0.473	$+0.1021$
276.1	321	0.340	$+0.0957$
496	458	0.528	$+0.0953$
96.71	893	0.493	$+0.0932$
311	348	0.426	$+0.0881$
272.0	593	0.395	$+0.0850$
410.71	328	0.428	$+0.0830$
272.4	982	0.340	$+0.0776$
37.22	344	0.333	$+0.0769$
Regressor (1/50):			
$V15.82$	300	0.295	-0.0149

Table 17: Per-class threshold pilot on MIMIC-III: top movers among the 50 head codes. Test-set macro-F1 on the top-50 codes rises from 0.6395 (global $\tau^{\star}_{\text{macro}}$) to 0.6868 (per-class), a $+0.0474$ improvement; overall macro and micro F1 also improve marginally ($+0.0006$ macro, $+0.0105$ micro). $49/50$ head codes improve; the single regressor (V15.82 "personal history of tobacco use", -0.015) is a low-frequency history-flag where the validation-tuned threshold over-fits the small positive count. The top-50 macro-F1 is invariant to the macro denominator (all 50 head codes have test support).

Category	Count	%	Description
Terminology mismatch	125	70.6%	Code in note text in different surface form
Hierarchy mismatch	36	20.3%	Parent in gold; specific child not retrieved
Ambiguity	16	9.0%	Code in top-N but oracle picked another

Table 18: Search-miss diagnosis on the MDACE 89-note span subset (177 FN gold codes). Categories assigned by a deterministic MDACE-span-grounded rule. Terminology mismatch dominates the failures.

(diagnoses) or 2-digit (procedures) period-stripped prefix is present; and *terminology mismatch* otherwise, conditional on MDACE coders having annotated at least one in-note evidence span for that code (a fifth branch, *not_attempted*, fires only when no gap-fill terms were extracted on the note). All 177 FN codes had at least one MDACE span, and all 89 events had at least one extracted gap-fill term, so the *un-spannable* and *not_attempted* buckets are both empty and every FN is classified into one of the three reported categories.

Evidence-span grounding rates. Table 19 reports per-decision-class grounding rates on the MDACE 89-note span subset, computed as token-F1 between the LLM's `evidence_quote` and the annotated evidence span at threshold $\tau{=}0.3$. Spannable verifier decisions (those for codes with at least one annotated span on the note) account for $107/568$ verifier outputs; the remainder lack ground-truth evidence locations and are excluded.

I Delta Attribution

Table 20 attributes each code added by the pipeline (relative to SFT_HIGH) to one of four categories: verifier recoveries (TPs admitted from the verification band), verifier false-accepts (FPs admitted from the band), gap-fill recoveries (TPs retrieved by gap-fill and confirmed), and gap-fill errors (FPs retrieved and confirmed). The error-to-recovery ratio cited inline in §5.2 is computed from these counts.

J Per-ICD-Chapter and Frequency-Stratified Breakdowns

Note. All tables and figures in this appendix are reported at the principled band (Appendix A); CIs as in Table 25.

Per-chapter F1 deltas. Figures 2 and 3 report per-chapter Δ_{bv}, Δ_{gf}, Δ_{vgf}, and Δ_{tot} with bootstrap 95% CIs (1,000 resamples, seed 314) for every kept chapter on the MIMIC-III FULL subset, separated by code type, and back the body summary in §5.3. Chapters that do not meet the 30-

Decision class	In-span (%)	n
Verifier (correct decisions)		
correct_accept	13.0	12/92
correct_reject	0.0	0/15
Gap-fill (confirmed adds)		
recovery (TP)	6.2	5/81
errors (FP)	5.3	27/511

Table 19: Evidence-span grounding rates on the MDACE 89-note span subset (token-F1 threshold 0.3). Gap-fill recoveries ground 1.2× more often than making errors; verifier correct-accepts ground at 13%, indicating most accept decisions cite text outside annotated evidence regions.

Category (vs SFT_HIGH)	MIMIC-III	MIMIC-IV
Verifier recovery (TP+1)	11,451	28,342
Verifier false-accept (FP+1)	39,542	103,348
Gap-fill recovery (TP+1)	2,451	3,446
Gap-fill error (FP+1)	32,997	74,425
verifier FP/TP ratio	3.45:1	3.65:1
gap-fill FP/TP ratio	13.5:1	21.6:1
combined FP/TP ratio	5.22:1	5.59:1

Table 20: Delta attribution: for every code the pipeline adds vs SFT_HIGH ($\tau{=}\tau^{\star}_{\text{micro}}$), the combined verifier-plus-gap-fill error-to-recovery ratio is ∼5.2–5.6 across MIMIC-III and MIMIC-IV. The combined ratio is dataset-invariant within ±0.4 across the two datasets (8,713 vs 19,801 admissions; 3,681 vs 7,942 in-vocabulary codes), pointing to a structural property of the verification mechanism rather than dataset-specific noise. The component ratios diverge: gap-fill alone is more error-prone on MIMIC-IV because the larger ICD-10 candidate space inflates retrieval false positives faster than the verifier can prune them.

event minimum-support gate are omitted (4 of 36 on FULL, 13 of 36 on OOD); the omitted-chapter list is in the supplementary CSVs. Diagnosis chapters (Figure 2) are dominated by positive Δ_{bv} with mostly mild Δ_{gf} swings; *Perinatal Period* is the deepest swinger ($-0.686 \rightarrow +0.494$). Procedure chapters (Figure 3) show much larger Δ_{gf} damage, with ten of fourteen kept chapters dropping more than -0.20 before VERIFY_GAP_FILL; the candidate verifier only partially recovers this, leaving thirteen of the fourteen kept procedure chapters net-negative on Δ_{tot}. The deepest are *Operations on the Urinary System* (-0.380), *Female Genital Organs* (-0.117), and *Musculoskeletal* (-0.079); the only net-positive procedure chapter is *Op. Nose, Mouth, and Pharynx* ($+0.106$).

False-positive source by frequency band. Table 21 decomposes the verifier false acceptance and

Band	Verifier FA	Gap-fill hall.	Total FP
Head (>1000)	13,146	4,875	18,021
Torso (51–1000)	23,274	12,111	35,385
Tail (1–50)	3,122	8,124	11,246
OOD (not in train)	0	7,887	7,887
Total	39,542	32,997	72,539

Table 21: False-positive source split by frequency band on MIMIC-III at the principled band (Appendix A). Verifier false-acceptance concentrates in head and torso (where the SFT band $[\theta_{\text{low}}, \theta_{\text{high}})$ predicts densely); gap-fill error spreads to rare-and-unseen bins, reflecting that retrieval populates rare-code slots.

gap-fill error totals from Table 20 across four frequency bands at the principled band (Appendix A). Verifier false-accepts concentrate in head and torso (92% of verifier FP), where the SFT classifier's mid-band predictions are dense; gap-fill errors spread sharply rarer (49% in the tail or OOD bins, vs 8% for the verifier), reflecting that retrieval populates rare-code slots that the verifier then waves through. The OOD column is non-zero only for gap-fill since the band verifier only sees in-vocabulary codes.

Frequency-stratified F1. Table 22 reports code-level micro and macro F1 for each pipeline stage within head/torso/tail bands following Edin et al. (2023): head (top 50 codes by training frequency, all with >1,000 occurrences), torso (rank 51–1,000, n=950), and tail (rank >1,000, n=2,681). We merge Edin et al.'s *few* (11–50) and *tail* (1–10) bins into a single rank-based tail to keep three rows per panel; main-text §5.4 quotes Edin's original 1–10 tail F1 for direct comparability with prior work. The pipeline incurs a roughly uniform ∼0.05 micro-F1 cost across all three bands, while macro-F1 gains concentrate in the tail ($+0.048$) with a small torso lift ($+0.007$) and a modest head loss (-0.023). Band verification alone supplies most of the macro movement (head -0.014, torso $+0.025$, tail $+0.048$); subsequent gap-fill is approximately macro-neutral within each band.

Per-chapter OOD recovery. Tables 23 and 24 report the chapter-level pipeline-vs-SFT_HIGH F1 delta on the OOD subset (codes outside the SFT classifier's training vocabulary); CIs as in Table 25. On diagnoses, recovery concentrates in chapters with stable, well-vocabularised terminology: *Congenital Anomalies*, *Injury and Poisoning*, and *Infectious / Parasitic Diseases* are the top three. On

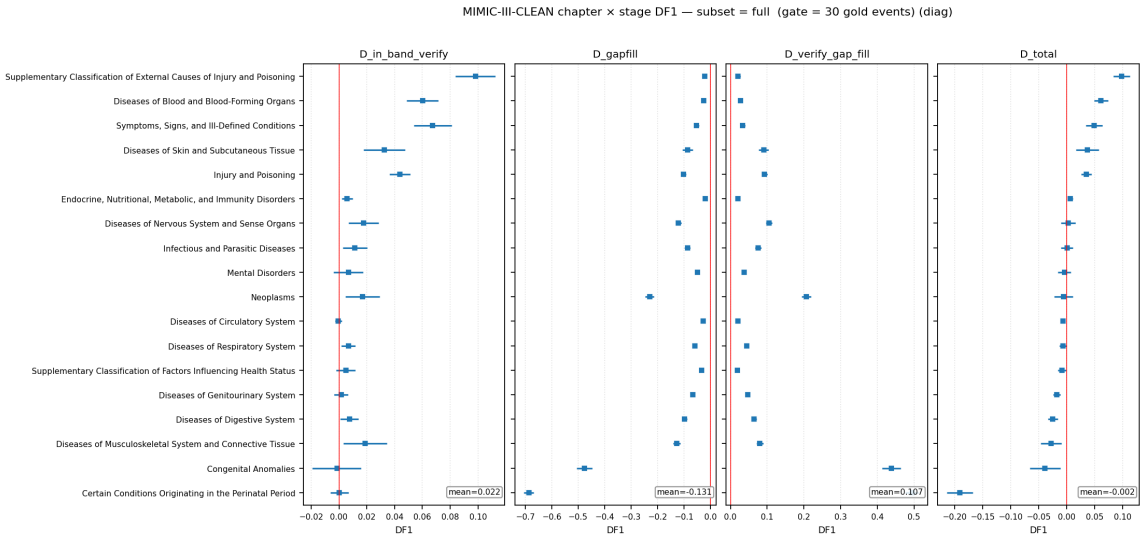

Figure 2: Diagnosis chapters (ICD-9-CM): per-chapter ΔF1 at each pipeline stage on the MIMIC-III FULL subset. Rows ordered by Δ_{tot}. CIs as in Table 25 (1,000 resamples).

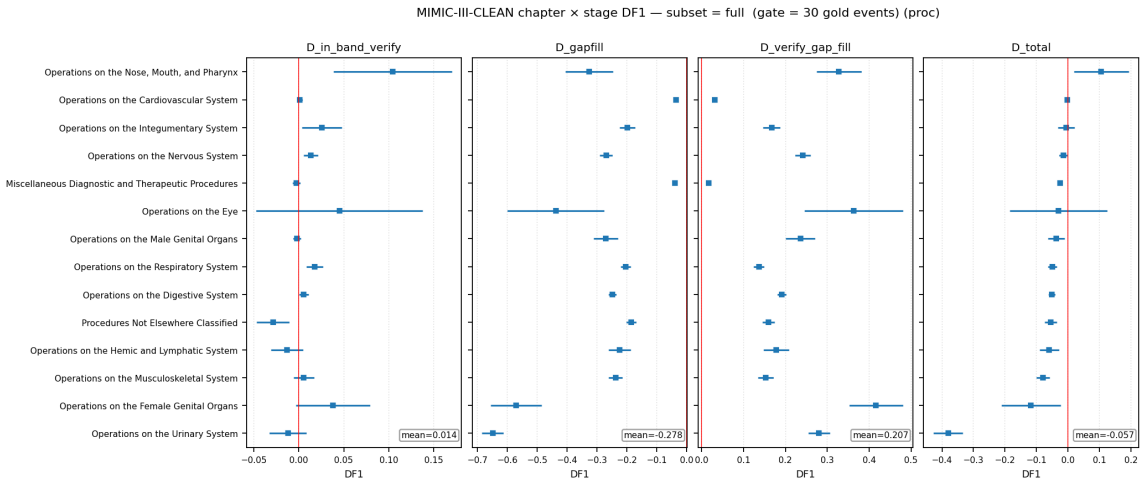

Figure 3: Procedure chapters (ICD-9-PCS): per-chapter ΔF1 at each pipeline stage on the MIMIC-III FULL subset. Rows ordered by Δ_{tot}. CIs as in Table 25 (1,000 resamples). Gap-fill swings are systematically larger than for diagnoses.

procedures, the support gate ($\geq$30 events) leaves only five eligible chapters; *Op. Musculoskeletal* leads. Chapters with high terminological variability (e.g. *Mental Disorders*, *Suppl. Factors Influencing Health*) sit at the bottom of the diagnosis ranking, consistent with the search-miss diagnosis in Appendix H.

Code-set intersection. $|\text{gap-confirmed} \cap \text{gold}|/|\text{gold} \setminus \text{SFT}| = 0.27$ on MDACE, showing that gap retrieval captures 27% of codes the SFT classifier missed entirely.

K Precision/Recall Decompositions

Bootstrap protocol. All 95% confidence intervals reported in this paper come from 1,000 admission-level bootstrap resamples (seed 314). Per-cell intervals are normal-approximation: $\hat{\theta} \pm 1.96 \cdot \widehat{\text{SE}}_{\text{boot}}$, which absorbs the rare-class-dropout bias that displaces the percentile and basic intervals on long-tail macro-F1. Paired contrasts (e.g. hybrid vs SFT@$\tau^{\star}_{\text{macro}}$ on FULL macro-F1) use the same resamples to compute a paired-bootstrap p-value on the per-resample Δ. This is the canonical description for every CI in the paper; Tables 3 and 5 omit CIs to keep the main-body layout single-column-clean, and refer here for full uncertainty

Stage	micro-F1	macro-F1
Head (n=50 codes)		
SFT_HIGH	0.722	0.678
+band_verify	0.693	0.664
+full pipeline	0.680	0.655
Δ_{tot}	−0.042	−0.023
Torso (n=950 codes)		
SFT_HIGH	0.576	0.469
+band_verify	0.552	0.494
+full pipeline	0.525	0.476
Δ_{tot}	−0.051	+0.007
Tail (n=2,681 codes)		
SFT_HIGH	0.361	0.226
+band_verify	0.382	0.274
+full pipeline	0.308	0.274
Δ_{tot}	−0.053	+0.048

Table 22: MIMIC-III FULL code-level F1 by training-frequency band, following Edin et al. (2023), at the principled band (Appendix A). Pipeline micro cost is roughly uniform across bins; macro gain is tail-concentrated. Macro-F1 values are over the in-distribution SFT vocabulary ($|C|$=3,681), all of whose codes have at least one test positive, so the scorable and full-label denominators coincide here.

Chapter	ΔF1	95% CI
Congenital Anomalies	+0.434	[+0.338, +0.530]
Infectious / Parasitic	+0.355	[+0.259, +0.452]
Injury and Poisoning	+0.313	[+0.270, +0.356]
Neoplasms	+0.209	[+0.155, +0.263]
Respiratory System	+0.208	[+0.133, +0.282]
Perinatal Period	+0.191	[+0.100, +0.281]
Nervous System / Sense Organs	+0.177	[+0.126, +0.228]
Skin / Subcutaneous Tissue	+0.174	[+0.069, +0.279]
Blood / Blood-Forming	+0.164	[+0.053, +0.275]
Endocrine / Metab. / Immune	+0.163	[+0.082, +0.244]
Genitourinary System	+0.146	[+0.082, +0.210]
Digestive System	+0.111	[+0.061, +0.162]
Mental Disorders	+0.098	[+0.046, +0.149]
Musculoskel. / Conn. Tissue	+0.094	[+0.057, +0.132]
Suppl. Ext. Causes	+0.086	[+0.026, +0.147]
Circulatory System	+0.067	[+0.040, +0.095]
Symptoms / Ill-Defined	+0.049	[+0.010, +0.087]
Suppl. Factors Infl. Health	+0.014	[−0.004, +0.031]

Table 23: OOD per-chapter pipeline-vs-SFT_HIGH ΔF1 on MIMIC-III, all 18 eligible diagnosis chapters (sorted by ΔF1) at the principled band (Appendix A). Per-chapter ΔF1 is the binary chapter-presence F1 delta, so it is invariant to the macro denominator of §4.2.

quantification.

Precision/recall/F1 decomposition. Table 25 carries the per-precision/recall/F1 micro and macro decomposition for every MIMIC-III and MIMIC-IV row of Tables 3 and 5 at the principled band (Appendix A); CIs as in Table 25.

Ranking metrics. Table 26 reports threshold-free ranking metrics (P@k, R@k for $k \in \{8, 15\}$, and R-precision) for the SFT classifier on MIMIC-III and MIMIC-IV; pipeline stages emit binary code sets without a native code ranking and are excluded by construction.

Chapter	ΔF1	95% CI
Op. Musculoskeletal	+0.265	[+0.188, +0.342]
Op. Respiratory	+0.264	[+0.142, +0.386]
Op. Digestive	+0.193	[+0.127, +0.260]
Op. Cardiovascular	+0.063	[+0.017, +0.109]
Misc. Diag. / Therapeutic	+0.044	[+0.026, +0.062]

Table 24: OOD per-chapter pipeline-vs-SFT_HIGH ΔF1 on MIMIC-III, all 5 eligible procedure chapters at the principled band; the 30-event support gate excludes the remaining 9.

Calibration. Table 27 is the calibration block on MIMIC-III and MIMIC-IV: exact-match ratio (EMR) and mean predicted codes per note. The three sub-tables share the same row order and CI methodology as Table 3.

Method	Scope	Micro			Macro		
		P	R	F1	P	R	F1
MIMIC-III (ICD-9-CM, $\lvert C\rvert = 3{,}681$)							
SFT only ($\tau{=}\tau^{\star}_{\text{macro}}$)	INDIST	$0.451_{[.448,.453]}$	$0.726_{[.723,.730]}$	$0.556_{[.554,.559]}$	$0.305_{[.300,.310]}$	$0.432_{[.427,.438]}$	$0.332_{[.327,.336]}$
	FULL	$0.451_{[.448,.453]}$	$0.713_{[.709,.716]}$	$0.552_{[.550,.555]}$	$0.201_{[.197,.206]}$	$0.285_{[.280,.291]}$	$0.219_{[.215,.223]}$
SFT only ($\tau{=}\tau^{\star}_{\text{micro}}$, SFT_HIGH)	INDIST	$0.636_{[.633,.639]}$	$0.591_{[.587,.595]}$	$0.613_{[.610,.616]}$	$0.345_{[.339,.351]}$	$0.297_{[.292,.302]}$	$0.295_{[.291,.300]}$
	FULL	$0.636_{[.633,.639]}$	$0.580_{[.576,.584]}$	$0.607_{[.604,.610]}$	$0.228_{[.223,.233]}$	$0.196_{[.192,.200]}$	$0.195_{[.191,.199]}$
+BAND_VERIFY	FULL	$0.516_{[.514,.519]}$	$0.664_{[.660,.668]}$	$0.581_{[.578,.584]}$	$0.220_{[.216,.225]}$	$0.258_{[.253,.263]}$	$0.222_{[.218,.226]}$
+GAP_FILL (raw)	FULL	0.255	0.698	0.373	0.152	0.379	0.191
Hybrid (ours)	INDIST	$0.458_{[.455,.460]}$	$0.692_{[.688,.696]}$	$0.551_{[.548,.553]}$	$0.304_{[.299,.310]}$	$0.430_{[.424,.436]}$	$0.331_{[.326,.336]}$
	OOD	$0.061_{[.054,.068]}$	$0.138_{[.124,.152]}$	$0.085_{[.076,.093]}$	$0.112_{[.101,.123]}$	$0.134_{[.123,.146]}$	$0.112_{[.102,.122]}$
	FULL	$0.447_{[.444,.450]}$	$0.682_{[.678,.685]}$	$0.540_{[.537,.542]}$	$0.239_{[.234,.244]}$	$0.330_{[.324,.335]}$	$0.257_{[.252,.261]}$
MIMIC-IV (ICD-10-CM, $\lvert C\rvert = 7{,}942$)							
SFT only ($\tau{=}\tau^{\star}_{\text{macro}}$)	INDIST	$0.427_{[.425,.429]}$	$0.730_{[.727,.732]}$	$0.539_{[.537,.540]}$	$0.274_{[.270,.277]}$	$0.418_{[.414,.422]}$	$0.306_{[.302,.309]}$
	FULL	$0.427_{[.425,.429]}$	$0.712_{[.710,.714]}$	$0.534_{[.532,.535]}$	$0.156_{[.153,.159]}$	$0.239_{[.235,.242]}$	$0.174_{[.172,.177]}$
SFT only ($\tau{=}\tau^{\star}_{\text{micro}}$, SFT_HIGH)	INDIST	$0.646_{[.644,.649]}$	$0.577_{[.574,.579]}$	$0.610_{[.608,.612]}$	$0.313_{[.309,.317]}$	$0.272_{[.268,.275]}$	$0.268_{[.265,.271]}$
	FULL	$0.646_{[.644,.649]}$	$0.563_{[.560,.565]}$	$0.602_{[.600,.604]}$	$0.178_{[.175,.181]}$	$0.155_{[.152,.158]}$	$0.153_{[.151,.156]}$
+BAND_VERIFY	FULL	$0.509_{[.507,.510]}$	$0.651_{[.648,.653]}$	$0.571_{[.569,.573]}$	$0.174_{[.172,.177]}$	$0.207_{[.204,.210]}$	$0.176_{[.173,.178]}$
+GAP_FILL (raw)	FULL	0.311	0.671	0.425	0.138	0.258	0.158
Hybrid (ours)	INDIST	$0.470_{[.469,.472]}$	$0.677_{[.674,.679]}$	$0.555_{[.553,.557]}$	$0.286_{[.282,.289]}$	$0.384_{[.380,.388]}$	$0.302_{[.299,.305]}$
	OOD	$0.049_{[.044,.054]}$	$0.048_{[.043,.053]}$	$0.049_{[.044,.054]}$	$0.035_{[.031,.038]}$	$0.043_{[.039,.046]}$	$0.035_{[.032,.038]}$
	FULL	$0.463_{[.462,.465]}$	$0.662_{[.659,.664]}$	$0.545_{[.543,.547]}$	$0.178_{[.175,.181]}$	$0.237_{[.234,.241]}$	$0.187_{[.184,.190]}$

Table 25: Precision, recall, and F1 (micro and macro) decomposition for every MIMIC-III and MIMIC-IV row of Tables 3 and 5. Bracketed subscripts are 95% bootstrap CIs (1,000 admission-level resamples, seed 314; normal-approximation interval $\hat{\theta} \pm 1.96 \cdot \widehat{\text{SE}}_{\text{boot}}$). OOD rows are omitted for SFT-only configurations because OOD codes are unreachable by construction (SFT vocabulary $\lvert C\rvert$=3,681 on MIMIC-III, $\lvert C\rvert$=7,942 on MIMIC-IV). +GAP_FILL is the raw retrieved-candidate union prior to verification, reported point-only as in Table 5.

Method	Scope	P@8	P@15	R@8	R@15	R-prec
MIMIC-III (ICD-9-CM, $\lvert C\rvert = 3{,}681$)						
SFT (any τ)	INDIST	$0.744_{[.740,.749]}$	$0.580_{[.576,.584]}$	$0.458_{[.454,.462]}$	$0.631_{[.627,.636]}$	$0.624_{[.621,.627]}$
SFT (any τ)	FULL	$0.744_{[.740,.749]}$	$0.580_{[.576,.584]}$	$0.448_{[.444,.452]}$	$0.619_{[.615,.623]}$	$0.617_{[.614,.620]}$
MIMIC-IV (ICD-10-CM, $\lvert C\rvert = 7{,}942$)						
SFT (any τ)	INDIST	$0.728_{[.725,.732]}$	$0.574_{[.571,.577]}$	$0.447_{[.445,.450]}$	$0.616_{[.613,.618]}$	$0.610_{[.607,.612]}$
SFT (any τ)	FULL	$0.728_{[.725,.732]}$	$0.574_{[.571,.577]}$	$0.435_{[.433,.438]}$	$0.600_{[.597,.603]}$	$0.602_{[.599,.604]}$

Table 26: Ranking metrics for the SFT classifier (Qwen3-4B-Base) on MIMIC-III and MIMIC-IV. Top-k and R-precision are properties of the ranked logit output and are independent of the decision threshold τ, so a single SFT row per scope suffices. Pipeline stages (+BAND_VERIFY, +GAP_FILL, Hybrid) emit binary code sets without a native ranking and are omitted. CIs as in Table 25 (1,000 resamples, seed 314).

Method	Scope	EMR	Codes/note
MIMIC-III (ICD-9-CM, $\lvert C\rvert = 3{,}681$)			
SFT only ($\tau{=}\tau^{\star}_{\text{macro}}$)	INDIST	$0.000_{[-.000,.000]}$	$24.83_{[24.58,25.08]}$
	FULL	$0.000_{[-.000,.000]}$	$24.83_{[24.58,25.08]}$
SFT only ($\tau{=}\tau^{\star}_{\text{micro}}$, SFT_HIGH)	INDIST	$0.004_{[.003,.006]}$	$14.32_{[14.18,14.46]}$
	FULL	$0.003_{[.002,.004]}$	$14.32_{[14.18,14.46]}$
+BAND_VERIFY	FULL	$0.001_{[.000,.002]}$	$20.17_{[19.98,20.37]}$
+GAP_FILL (raw)	FULL	0.000	42.93
Hybrid (ours)	INDIST	$0.001_{[.000,.001]}$	$23.30_{[23.08,23.51]}$
	OOD	$0.484_{[.474,.494]}$	$0.65_{[0.63,0.67]}$
	FULL	$0.000_{[-.000,.001]}$	$23.95_{[23.73,24.17]}$
MIMIC-IV (ICD-10-CM, $\lvert C\rvert = 7{,}942$)			
SFT only ($\tau{=}\tau^{\star}_{\text{macro}}$)	INDIST	$0.001_{[.000,.001]}$	$27.10_{[26.91,27.30]}$
	FULL	$0.001_{[.000,.001]}$	$27.10_{[26.91,27.30]}$
SFT only ($\tau{=}\tau^{\star}_{\text{micro}}$, SFT_HIGH)	INDIST	$0.006_{[.005,.007]}$	$14.16_{[14.05,14.26]}$
	FULL	$0.005_{[.004,.006]}$	$14.16_{[14.05,14.26]}$
+BAND_VERIFY	FULL	$0.002_{[.002,.003]}$	$20.81_{[20.66,20.95]}$
+GAP_FILL (raw)	FULL	0.000	35.04
Hybrid (ours)	INDIST	$0.002_{[.001,.002]}$	$22.83_{[22.68,22.99]}$
	OOD	$0.555_{[.548,.562]}$	$0.39_{[0.38,0.40]}$
	FULL	$0.001_{[.001,.002]}$	$23.22_{[23.06,23.37]}$

Table 27: Calibration check on MIMIC-III and MIMIC-IV. EMR is the fraction of admissions whose predicted code set exactly equals the gold set (after intersecting both with the scope vocabulary). Codes/note is the mean predicted-set cardinality. Codes/note is invariant to scope for predictors whose output is a subset of INDIST (SFT-only and +BAND_VERIFY); the OOD row for Hybrid reflects only the candidate-verifier additions to the OOD vocabulary. CIs as in Table 25 (1,000 resamples, seed 314).